Guest Editors' Introduction to the Special Issue on Computational Intelligence

Computational intelligence is branch of artificial intelligence that studies a wide range of tasks and problems especially difficult for traditional algorithms to deal with: those problems that include uncertainty, highly stochastic behavior, or fuzziness. Probably the majority of problems currently dealt with within the artificial intelligence framework, and a large share of problems addressed by modern computer science in general, fall in this category. Computational intelligence seeks to achieve solutions of such complex problems with the techniques that presumably human brain uses, and with the degree of quality equal, or superior, to the quality with which humans solve such problems.

Among the problems particularly difficult for computers and particularly easy for human brain we can mention everyday human activities such as intuition, learning, noticing patterns, dealing with language, vision, and spatial orientation. Their computer counterparts are the areas of machine learning, natural language processing, and image processing, among others. Of these areas, machine learning is a set of techniques deeply penetrating all other areas of artificial intelligence, while natural language processing, image processing and computer vision, and other research areas have more applied character and are oriented to model different specific abilities of the human brain.

For this special issue, we have selected a representative collection of fourteen papers presenting the latest advances in all these areas of research and practical applications.

The first two papers in this issue represent an important application of computational intelligence: recommender systems and evaluation scales. Recommender systems improve the quality of life of the consumers by helping them to make informed decision on buying products and services, basing on the experience of other users. Evaluation scales play a key role in correct functioning of recommender systems, as well as help the businesses to improve their products and services to better match the opinions of the users.

C. Ríos et al. from Argentina in their paper "Selecting and weighting users in collaborative filtering based POI recommendation" analyze a wide range of techniques that improve location awareness of the recommender systems. In many scenarios it is important for the user to obtain recommendations of products and services available in the geographic vicinity of that user, as well as rated highly by other users from the same geographical region. In addition, geographic awareness of the system helps in disambiguating the names of products and services, such as local restaurants or shops with common names. The authors show how to extract information on geographic location from the data available in various social networks.

I. Batyrshin et al. from Mexico and Russia in their paper “Bipolar Rating Scales: A Survey and Novel Correlation Measures Based on Nonlinear Bipolar Scoring Functions” give a comprehensive state of the art review of the theory and practices of the use of bipolar rating scales: evaluation scales in which people can express various degrees of positive or negative opinion on some product or subject. Such scales are ubiquitous in all kinds of evaluation and their applications, from opinion mining and recommender systems to healthcare, administration and politics. Basing on the analysis of the current state of the art, the authors propose novel, improved techniques for the analysis of opinions expressed in terms of such scales. The techniques proposed in the paper are based on the so-called non-linear bipolar scoring functions, which describes in the objective terms the degree of utility, or satisfaction, expressed in a given scale.

The next group of five papers are devoted to natural language processing, one of the key areas of research and application in computational intelligence and probably the most “human” one. In a wide sense, natural language processing is a research area devoted to the ways to enable the computers to deal with text, or speech, in ordinary human language, such as English or Hungarian, the way people do, or even better—given the ability of the computer to quickly process huge quantities of data. In more specific applications, natural language processing research results in a wide range of important technologies, from information retrieval and machine translation to opinion mining and sentiment analysis. The latter techniques, for example, are the basis for the development of business intelligent tools and recommender systems.

I. Markov et al. from Mexico and Portugal in their paper “Authorship Attribution in Portuguese Using Character N-grams” present a method for detecting the author of a given text out of a number of possible alternatives. They show that character n-grams are very good features for this task, and analyze the performance of a wide range of types of character n-grams. Authorship attribution has numerous applications in culture, education, forensics, and business intelligence. For example, in culture and education it helps fighting plagiarism, a dangerous phenomenon that has become threateningly common with the proliferation of Internet. In forensics, it allows for the determination of the author of texts related with, for example, a crime. In business intelligence, authorship attribution and author profiling methods improve the performance of opinion mining techniques.

J.-P. Posadas-Durán et al. from Mexico in their paper “Algorithm for Extraction of Subtrees of a Sentence Dependency Parse Tree” describe the procedure for enumerating the so-called syntactic n-grams present in a syntactic dependency tree of a sentence. Syntactic information present in a sentence is important for its interpretation. However, it is difficult to represent this information in a way useful for modern machine-learning methods, which are mostly suitable for data represented in the form of vectors and not trees or other graphs. Typically, for the use with such methods, the text is represented in the form of word n-grams, which are linear sequences of words located in the text next to each other, with the syntactic information completely lost. The technique of syntactic n-grams allows to keep the syntactic information while still representing the text as a vector of features. Thus, extracting these features from the text becomes the basic task for any application of syntactic n-grams. The paper presents a detailed algorithm for extracting these features from the texts.

S. Miranda-Jiménez and E. Stamatatos from Mexico and Greece in their paper "Automatic Generation of Summary Obfuscation Corpus for Plagiarism Detection" continues the discussion of the topic of authorship attribution and plagiarism detection by describing a method they used for automatic generation of a corpus of plagiarized documents of a specific type: plagiarism obfuscated via summarization. The corpus was used as a dataset for the most prestigious international competition of plagiarism detection systems. Automatic plagiarism detection is nowadays extremely important for the normal functioning of our education system as well as academia, given, on the one hand, the ease of committing this kind of severe academic misconduct using huge information resources of the Internet and, on the other hand, the fact that our education system and academia completely depend on the evaluation of texts authored by the student or researcher for correct scoring and promotion of productive and honest researchers.

O. Pichardo-Lagunas et al. from Mexico in their paper "Automatic detection of semantic primitives with multi-objective bioinspired algorithms and weighting algorithms" address the topic of automatic semantic analysis of explicatory dictionaries and evaluation of their quality via detection of primitive concepts on which the description of all other words can be based—much as the description of all notions in school geometry is based on a few non-definable concepts such as point and line. For this purpose, they represent the dictionary as a directed graph, with words being the nodes, and an arc from one word to the other if the latter is a part of the description of the former. Using computational intelligence algorithms, the authors determine the optimal way of making this graph cycle-free; then the nodes only used in the definitions of other words represent the optimal defining vocabulary of the given dictionary.

J. Alvarado-Uribe et al. from Mexico in their paper "Semantic Approach for Discovery and Visualization of Academic Information Structured with OAI-PMH" discuss the ways for analysis of the complex network of interrelated information on published scientific papers available from open-source public repositories via indexing metadata. Open source publication model is considered by many researchers to be the best way of dissemination of scientific results. However, the great amount of published papers available via open source publishers and repositories requires efficient aggregation and search tools for its analysis and extraction of information relevant for a specific user and specific research topic. The authors present the tools they have developed for visualization of the structure of information available in open-source repositories and for finding relevant materials by semantic queries using ontologies and other advanced analysis techniques.

The next group of three papers is devoted to another important research and application area of computational intelligence: image processing and computer vision, which is responsible for the ability of intelligent computers to “see” and interpret images as persons do, or even better. Among numerous applications of this research area are image retrieval, medical diagnostics, public security and forensics, agricultural monitoring, robotics, and autonomous vehicles, to name only a few. Image processing was the area where modern deep learning revolution began and from which it has spread to other areas of computational intelligence, such as natural language processing and general machine learning.

E. Moya-Albor et al. from Mexico in their paper “An Edge Detection Method using a Fuzzy Ensemble Approach” explain the use of fuzzy logic techniques for edge detection in image processing. Edge detection is one of the basic techniques in image processing, which is a preprocessing step for more advanced classification and image recognition methods. The task of edge detection is to determine the boundaries of the objects seen on an image, or the boundaries between different parts or faces of the object, such as different walls of a building (the one facing the camera and the ones facing sideways), the limits of a road in front of an autonomous vehicle, the contours of human figures, etc. The task is particularly interesting because it involves global analysis of the image and not only relations between neighboring pixels. The authors show that their method outperforms the known methods on a ground truth dataset for which the task has been previously solved manually.

H. Castillejos-Fernández et al. from Mexico in their paper “An intelligent system for the diagnosis of skin cancer on digital images taken with dermoscopy” apply edge detection techniques and fuzzy logic classification to the task of automatic skin cancer diagnosis. Skin cancer is a major health threat in all countries across the world, difficult to diagnose with traditional methods, which require highly qualified and experienced medical personnel. Such personnel is not available in many places, especially in highly populated regions of Asia, Africa, and Latin America with developing economy. On the other hand, early diagnosis of skin cancer is crucial to reduce mortality rate from this disease. The combination of these two factors makes computational methods of automatic diagnosis especially relevant. The authors present a detailed diagram of the system they have developed, and show that their method outperforms existing state-of-the-art classifiers on an available dataset of pre-classified medical images.

T. Katanyukul and J. Ponsawat from Thailand in their paper “Customer Analysis via Video Analytics: Customer Detection with Multiple Cues” present a computer vision application for video surveillance in closed circuit video system in a shop or supermarket. Identification and monitoring of specific customers provides important security and business intelligence information through the analysis of customer behavior in a commercial establishment. However, such identification is a complex process that involves a number of cues of different nature, such as the semantic and spatial context, common-sense and domain knowledge, previous experience, etc. The authors present an integrated pipeline of feature extraction, classification, and integration of various sources of knowledge for detection of persons in the surveillance data. They report very significant improvement of more than 42% over the existing methods in terms of precision.

Finally, the last four papers sample practical useful applications of traditional computational intelligence methods such as forecasting, clustering, optimization, and intelligent control and show how these techniques can be used in diverse practical tasks from disaster prediction, bioinformatics, and economy to robotics and mechatronics.

Justin Parra et al. from the US in their paper "Use of Machine Learning to Analyze and – Hopefully – Predict Volcano Activity" analyze a wide specter of natural phenomena that precede known volcanic eruptions, with the ultimate goal of predicting new volcanic eruptions in time for the evacuation of people from the affected zone. The task is very important because, as the authors mention, hundreds of volcanoes in the state of unrest, ready to erupt at any moment, are located near large urban areas. As a case study, the authors consider a well-documented 1999 eruption of Redoubt volcano in Alaska, with rich geological and geophysical data relevant for this eruption publicly available through Smithsonian Institution Global Volcanology Project, as well as the Aerocom database. While not reporting a ready forecasting technique, the authors show that a suitable analysis of the precursors of eruption yields geophysically meaningful results, which makes such analysis promising for eventual development of algorithms for predicting dangerous volcanic activity.

I. Bonet et al. from Colombia in their paper "Clustering of Metagenomic Data by Combining Different Distance Functions" develop a novel clustering method based on the consensus of clustering results with different similarity measures and use this method for identification of species by genome sequences extracted from a natural mix of genetic material. In pure laboratory experiments one can isolate a species of microorganisms and obtain a clean sample of genomic material. However, in real-life environment often only a mix of genetic fragments belonging to many different organisms is available, with many of them not being identifiable using existing genomic databases. In such circumstances it is important to define automatically which genetic fragments belong to the same species and cluster the genetic sequences by this criterion. The authors solve this problem using a variant of k-means classifier with an ensemble of different distance functions.

J. G. Flores Muñiz et al. from Mexico, Russia, the US, and Ukraine in their paper "Gaussian and Cauchy Functions in the Filled Function Method – Why and What Next: On the Example of Optimizing Road Tolls" argue for computational complexity to be an important criterion in selection the best smoothing function in filled function method of solving optimization problems. In optimization of functions with complex behavior, the key issue is to avoid local optima, since this prevents the algorithm from finding the globally optimal solution. The filled function method consists in approximating the original function by a smoother function with much simpler behavior the global optimum of which is easier to find; the global optimum of the original function is likely to be located near the global optimum of the smoothed function. The authors explain why two particular functions often used for smoothing are so efficient, and illustrate this with a case study of the economic problem of optimization of road tolls.

V. V. Chikovani et al. from Ukraine in their paper “External Disturbances Rejection by Differential Single-Mass Vibratory Gyroscope” introduce a novel operational mode of a vibratory gyroscope: a differential operation mode, which in their experiments shows high improvement in robustness as compared with the usual rate mode. Gyroscopes are essential for spatial orientation and stabilization of intelligent physical systems such as drones, robotic arms, and virtual reality devices, to name a few. Many of these intelligent devices operate in real-life environments where they are prone to mechanical stress such as vibrations and shocks, which prevent traditional gyroscopes from normal function. With their detailed analysis the authors show that in the new operational mode the sensitivity of the gyroscope to vibration and shocks is greatly reduced.

This special issue of Acta Polytechnica Hungarica devoted to diverse topics of computational intelligence theory and applications will be useful to researchers and students working in such areas of artificial intelligence as recommender systems, natural language processing, image processing and computer vision, forecasting, clustering, optimization, and intelligent control.

Ildar Batyrshin **Alexander Gelbukh** **Grigori Sidorov**

CIC, Instituto Politécnico Nacional, 07738 Mexico City, Mexico

nlp.cic.ipn.mx/~batyrshin www.cic.ipn.mx/~gelbukh www.cic.ipn.mx/~sidorov

Selecting and Weighting Users in Collaborative Filtering-based POI Recommendation

Carlos Ríos, Silvia Schiaffino and Daniela Godoy

ISISTAN Research Institute (CONICET-UNCPBA)
Campus Universitario, Paraje Arroyo Seco
CP 7000, Tandil, Bs. As., Argentina
{carlos.rios,silvia.schiaffino,daniela.godoy}@isistan.unicen.edu.ar

Abstract: Location-based recommender systems (LBRSs) provide a technological solution for helping users to cope with the vast amount of information coming from geo-localization services. Most online social networks capture the geographic location of users and their points-of-interests (POIs). Location-based social networks (LBSNs), like Foursquare, leverage technologies such as GPS, Web 2.0 and smartphones allow users to share their locations (check-ins), search for POIs, look for discounts, comment about specific places, connect with friends and find the ones who are near a specific location. LBRSs play an important role in social networks nowadays as they generate suggestions based on techniques such as collaborative filtering (CF). In this traditional recommendation approach, prediction about a user preferences are based on the opinions of like-minded people. Users that can provide valuable information for prediction need to be first selected from the complete network and, then, their opinions weighted according to their expected contribution. In this paper, we propose and analyze a number of strategies for selecting neighbors within the CF framework leveraging on information contained in the users' social network, common visits, visiting area and POIs categories as influential factors. Experimental evaluation with data from Foursquare social network shed some light on the impact of different mechanisms on user weighting for prediction.

Keywords: Location-based social networks; recommender systems; user-based collaborative filtering.

1 Introduction

Recent technological advances in the development of wireless communications, the great explosion of cell phone use, and the easiness to acquire the geographical location of people, have allowed the creation of social services whose main feature is the geographical location of users. In this new era, users can benefit from obtaining a pervasive and ubiquitous access to location-based services from anywhere through mobile devices. Foursquare[1] is the most popular location-based social media service [11], allowing users to easily share their geographical location as well as contents related to that location in an online way. The user location is a new dimension in social networks that created new opportunities and challenges for traditional recommendation systems. Recommender systems are an alternative to deal with the problem of information overload that users face while seeking information about items of interest in vast amounts of knowledge. Traditional methods such as collaborative filtering (CF), content-based recommendation (CB) and hybrid methods [29] process information obtained from the ratings provided by users and the characteristics of the items involved to generate a list of recommendations. However, in social networks there is additional information that recommendation methods should take into account, such as users' behavior and relations of friendship between them [38].

Regarding location-based social networks (LBSNs), geo-localized data is a physical dimension that traditional social networks do not possess. In this context, location-based recommender systems (LBRSs) have emerged [5] as a means to exploit geographical properties as an auxiliary source for recommending friends [25, 31], places [23], activities [39, 7] and events [27, 15]. The heterogeneity of the data produced by location-based social networks creates the need for new approaches in recommendation systems, using different data sources and methodologies for enhancing recommendations. The Collaborative Filtering (CF) approach, for example, relates users to items through ratings or opinions, so that it can be straightforwardly applied to the construction of LBRSs. However, the traditional CF approach lacks the geo-localization dimension.

In this work, we propose different strategies for including the additional dimensions available in LBSNs in the context of user-based collaborative filtering for recommending locations. In a LBSN, there are relationships of various types, such as the User-User relationship, showing the friendship between two users or the coincidence in places visited by these users; the User-Place relationship showing that a user visited a given place; the Place-Place relationship, which shows distance relationships between places or categorical membership. Also, in addition to these relationships, users generate content-based relations by providing comments or tips after visiting a place.

User-based approaches recommend items (e.g. places) based on an aggregation of the preferences of similar users or neighbors, i.e. users with similar tastes. As user-based CF trusts neighbors as information sources, the quality of recommendations is a direct consequence of the selected neighborhood and the importance given to each neighbor for prediction. In a previous work [30], we have studied different strategies for selecting neighbors considering the information contained in the LBSNs. We have concluded that selecting as neighbors those users that have visited the same places as the target user we can obtain lower errors in the preference estimation process. To weight neighbors, some works have considered factors such as trust or geographical or social influence [3]. In this work, our main hypothesis is that location-based social networks provide rich information that can enable us to

[1] http://es.foursquare.com/

give neighbors representative weights and, hence, improve the estimation of preferences during the recommendation process. We propose and evaluate nine strategies using real data from a Foursquare dataset. From the empirical evaluation, we found that some of the proposed alternatives outperformed the traditional approach, giving developers some hints about which aspects to consider when making recommendations.

The rest of the article is organized as follows. Section 2 presents an overview of the CF approach for POI recommendation, describing the framework in which the proposed strategies for selecting and weighting neighbors fits in. Section 3 describes the experimental results we carried out to evaluate the different strategies and our findings. Then, Section 4 analyzes some related works. Finally, in Section 5 we present our conclusions and outline some future works.

2 CF-based POI recommendation

In a traditional CF scenario, there are m users $U=u_1, u_2, \ldots, u_m$, and a list of n items $I=i_1, i, \ldots, i_n$, that can be recommended to users. Each user has expressed her opinion about a set of items $I_{u_i} \subseteq I$, generally in an explicit way with a rating or value in a given numerical scale. This information is stored in a user-item matrix M of size $m \times n$, such that the value of each cell in M represents the preference score (rating) given by user i to item j. Memory-based CF approaches make predictions based on the user-item matrix in two ways, based on users or based on items [1]. Given an active user who requires a prediction for an item without rating, CF algorithms measure the similarities between the active user and other users (user-based approach), or between the item and the remaining items (item-based approach). Therefore, a rating is predicted by an aggregation of the ratings that the item received from similar users in the first case, or ratings given by the active user to similar items in the second case.

The classic user-based CF model is then defined as in Equation 1 [28]:

$$\tilde{r}(u,i) = \bar{r}(u) + C_o \sum_{v \in N_k(u,i)} sim(u,v)(r(v,i) - \bar{r}(v)) \tag{1}$$

where $r(v,i)$ is the rating given by user v to item i, $\tilde{r}$ is the rating prediction (different from the observed rating r), $N_k(u,i)$ is the set of k most similar users to u and $sim(u,v)$ is the function that determines the similarity between users u and v. C_o is a normalizing factor. The preference of user u for an item i is predicted according to the average rating $\bar{r}(u)$, the sum of deviations of the ratings given by the neighbors v to item i and the average ratings $\bar{r}(v)$, weighting by the similarity with neighbors.

User-based approaches assume that not all users are equally useful in the prediction for a given user, thus two main problems emerge: (1) selecting neighbors for a user to generate recommendations; (2) how to use properly the information provided by those neighbors in the generation of recommendations. Usually, the selection of neighbors is based on their similarity to the active user, while a common practice is to define a maximum number of users to narrow the neighborhood. Once the neighborhood is defined, the contribution of each neighbor to the prediction is weighted based on their distance from the active user. For example, a widely used alternative is a linear combination of the ratings weighted by the similarity of the neighbors. However, there are other factors that may be valuable for selecting neighbors. For example, in the case of this work the users' history of visits can be considered relevant beyond the ratings similarity.

To properly separate the two problems, the selection of neighbors on the one hand and the weighting of their opinions on the other hand, [6] proposes a modification of the classic formula. This new formula considered an allocation score function (scoring) depending on the active user *u*, a neighbor *v* and an item *i*, or some combination thereof. This function gives a higher value when the triplet of user-neighbor-item is more valuable or expected to work better in predicting a rating according to the available information. Eq. 1 is then generalized as Equation 2:

$$\tilde{r}(u,i) = \bar{r}(u) + C_o \sum_{v \in g(u,i;k;s)} f(s(u,i,v), sim(u,v)) * (r(v,i) - \bar{r}(v)) \tag{2}$$

where g is the function that selects neighbors and f is an aggregation function that combines the outcomes of the scoring function s and $sim(u,v)$ the similarity between users.

The selection of neighbors involves the determination of the similarity of users to the target user, by making a comparison with all the users in the database. So any user that is similar to the target user may contribute to the preference estimation. The function g (selection of neighbors) may be influenced by relations present in a LBSN. Thus, restricting with some criteria the potential neighborhood of a target user by exploiting the information generated in LBSN, we can reduce the number of comparisons and, at the same time, improve the preference estimation.

Regarding the scoring function s, we can use different information available in the LBSN to determine which neighbors are better predictors and hence, improve the preference estimation. For example, it can be assumed that users visiting the same places are more useful for prediction than those visiting a different set of places. Likewise, we can suspect than users establishing a friendship relationship are more valuable as a source for estimating preferences. In this context, we present different approaches for the scoring function once neighbors are selected and evaluate them empirically.

2.1 Selection of Neighbors

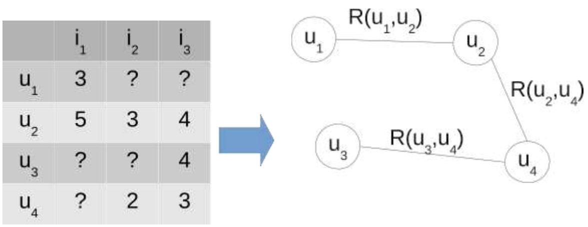
In a location-based social network users can be related by common visited places. In this paper, we used a strategy for selecting neighbors based on graph of common visited places. In a previous work [30], this strategy outperformed other strategies considered for selecting neighbors.

Particularly, it was compared with a strategy that selects neighbors from the user social network (friendship relationships), and two strategies based on the geographical location of users, one choosing users from the same state and other based on the intersection of visiting areas.

The strategy based on common visited places assumes that users that have visited the same places as the target user are the most valuable for asking opinions. Thus, a graph is generated starting from the preference matrix where nodes represent users and an edge between two nodes implies that both users have visited at least one place in common. The relations in the graph are weighted by the number of coincides in the same places normalized considering the total number of places visited by the user. Equation 3 shows how to obtain the value of the relationship $R(u_1,u_2)$ between two users u_1 and u_2. p_{u1} and p_{u2} are the places visited by users u_1 and u_2 respectively. In order to keep only strong relationships between users, we discarded those edges with a weight below the average.

$$R(u_1,u_2) = \frac{|p_{u1} \overset{\text{n}}{\;} p_{u2}|}{|p_{u1} \quad p_{u2}|} \qquad (3)$$

Figure 1 shows an example of the graph formation. Once the graph has been created, it is traversed for selecting the best possible neighbors. The graph traversal can be performed up to several depth levels. In other words, the first level will be formed only by users that have visited the same places as the target user. After the first level, users can be selected if they are related to users in the first level. As the graph is explored further, the number of selected neighbors grows. Consequently, not only the computational cost of computing prediction is higher, but also noisy users can be incorporated. In the experimental evaluation performed, the extent to which the graph should be explored is analyzed.



	i_1	i_2	i_3
u_1	3	?	?
u_2	5	3	4
u_3	?	?	4
u_4	?	2	3

Figure 1
Example of a graph of common visiting places

In the graph exemplified by Figure 1, there is not edge between users u_1 and u_3 as they have no common places. The stronger relationship is between users u_2 and u_4 as $R(u_2,u_4) = 0.40$, whereas the remaining ones are $R(u_1,u_2) = 0.25$ and $R(u_3,u_4) = 0.33$.

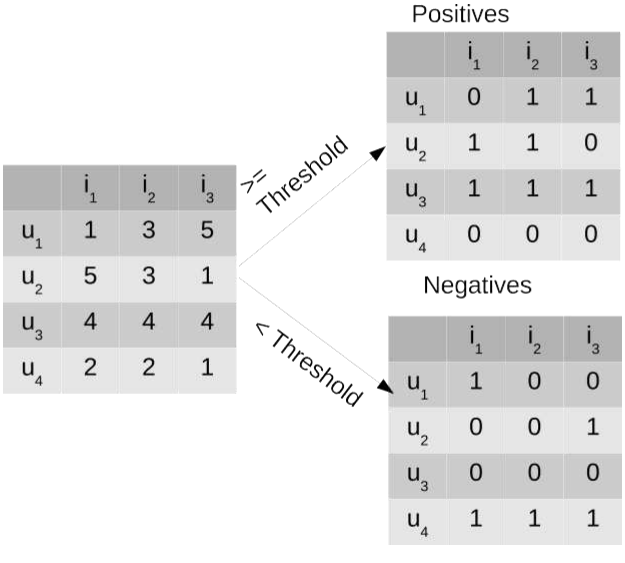
2.2 Strategies for Weighting Neighbors

In this article, we propose nine scoring or neighbor weighting strategies considering different pieces of information contained in a LBSN. These strategies are described below in different groups, those based on a discretization of the preference matrix (Subsection 2.2.1), those based on the proximity of the area visited by users (Subsection 2.2.2), those leveraging the categories of POIs (Subsection 2.2.3) and those using the friendship graph and the graph of common visited places (Subsection 2.2.4).

2.2.1 Scoring based on preference matrix discretization

This strategy proposes to derive two new matrices from the original preference matrix by discretizing user preferences. These matrices are called "positive" and "negative" and they represent a coincidence both in the places visited and in the positive or negative user preference towards places. The rationale behind these strategies is that the commonalities on positive and negative preferences makes users more valuable for estimating the preferences of new places.

A first strategy is based on the positive matrix exclusively and a second one is based on both positive and negative matrices. In the last case, we use Jaccard similarity to combine both matrices and obtain a metric that represents the correspondence in negative and positive preference. Jaccard similarity is shown in Equation 4. Figure 2 shows an example of user preference discretization. In this example, a threshold value of 3 in a five-point scale is considered to classify preferences into positive or negative ones. Equations 5 and 6 represent these scoring strategies.



	i_1	i_2	i_3
u_1	1	3	5
u_2	5	3	1
u_3	4	4	4
u_4	2	2	1

Positives

	i_1	i_2	i_3
u_1	0	1	1
u_2	1	1	0
u_3	1	1	1
u_4	0	0	0

Negatives

	i_1	i_2	i_3
u_1	1	0	0
u_2	0	0	1
u_3	0	0	0
u_4	1	1	1

Figure 2
Matrix binarization process

$$Jaccard(u,v) = \frac{|u^{\cap} v|}{|u \quad v|} \tag{4}$$

$$s1(u,v,i) = s(u,v) = Jaccard(u,v)_{PositiveMatrix} \tag{5}$$

$$s2(u,v,i) = s(u,v) = mean(Jaccard_{PositiveMatrix}, Jaccard_{NegativeMatrix}) \tag{6}$$

2.2.2 Scoring based on users' area proximity

In this strategy, the area visited by users is used for weighting higher those users that are moving in the same region. For each user, we first obtain the geographical area that covers the places most visited by the user. Then, the scoring is computed by assessing the distance between the areas for the different users. Equation 7 shows the scoring function for this strategy, where $dist\ (areaU, areaV)$ is the geodesic distance between the centers of each area.

$$s3(u,v,i) = s(u,v) = 1/dist(areaU, areaV) \tag{7}$$

According to this equation, the closer the areas of both users the higher the scoring. Thus, the users visiting areas less distant to each other are considered more relevant for obtaining opinions.

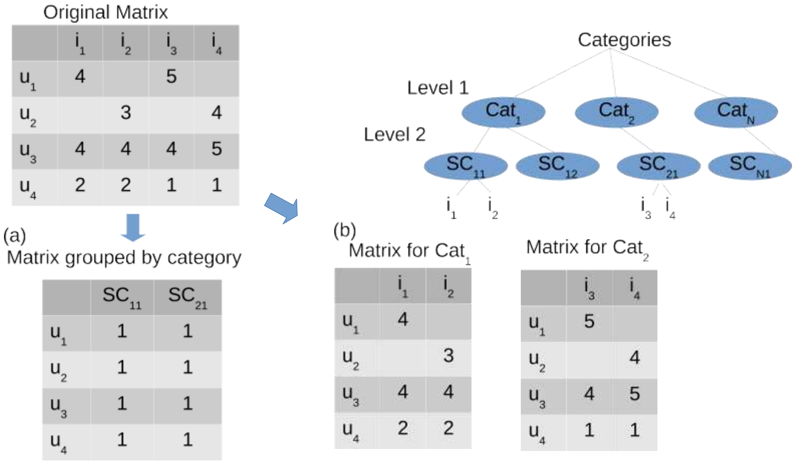
2.2.3 Scoring based on POIs categories

POIs can be categorized depending on the application domain according to a hierarchy. For example, the dataset we used contains a hierarchy of three levels, where the first level contains general categories such as "Arts & Entertainment", "Food", and "Nightlife Spot". The second level contains more specific categories, such as "Arcade" and "Art Gallery" for "Arts & Entertainment"; "South American Restaurant" or "Taco Place" for "Food"; and "Strip Club" and "Whisky Bar" for "Nightlife Spot". Then, the third level contains lower-level categories such as "Indie Movie Theater", "Paella Restaurant", "Rock Club", etc.

Leveraging POIs categories we define strategies for weighting neighbors based on two approaches. The first approach derives from the original preference matrix another matrix that shows places grouped by the lowest category in the hierarchy. This approach enables to find more intersections between users than the original matrix, since we are not considering specific places (a certain restaurant) but the category to which they belong to (e.g. paella restaurant).

The second approach, proposes a specialization of the original preference matrix according to the categories in the top level of the hierarchy. Thus, if the hierarchy has N categories in the first level, we obtain N preference matrices, one for each category. Then, to compute the scoring of neighbors, we need information about the user, the neighbor and the POI category. This second strategy aims at giving more importance to those users that visit places in the same category. Equations 8 and 9 show the scoring functions proposed for these two strategies, where $Jaccard(u,v)$ is Jaccard similarity and $sim(u,v)$ is any other similarity metric.

Figure 3 shows an example of the proposed strategies. In Figure 3(a) the original matrix is transformed into a matrix in which all of the users have coincidences on items belonging to two of the lowest categories SC_{11} and SC_{21}, that could be for example "Indie Movie Theater" in one branch of the hierarchy and "Paella Restaurant" in other branch. Figure 3(b) shows instead the translation of the original matrix to one matrix by category. One matrix collects the ratings on a main category Cat_1, that could be for example "Arts & Entertainment", and other matrix gathers the rating of another category Cat_2, for instance "Food". Thus, both matrices can be used separately depending of the category of the target item to generate predictions.



Original Matrix

	i_1	i_2	i_3	i_4
u_1	4		5	
u_2		3		4
u_3	4	4	4	5
u_4	2	2	1	1

Matrix grouped by category

	SC_{11}	SC_{21}
u_1	1	1
u_2	1	1
u_3	1	1
u_4	1	1

Matrix for Cat_1

	i_1	i_2
u_1	4	
u_2		3
u_3	4	4
u_4	2	2

Matrix for Cat_2

	i_3	i_4
u_1	5	
u_2		4
u_3	4	5
u_4	1	1

Figure 3
Extraction of preference matrices based on categories

$$s4(u,v,i) = s(u,v) = Jaccard(u,v)_{MatrixGroupedByCategory} \tag{8}$$

$$s5(u,v,i) = sim(u,v)_{CategoryMatrix-i} \tag{9}$$

2.2.4 Scoring based on a graph relationships

This strategy proposes the utilization of structural similarity metrics between two nodes in a network. In our domain, we can use the social graph representing friendship relationships between users. Also, we can build a graph in which nodes are users and the relationships among them represent common visited places by the users as the one used in the neighbor selection strategy. In the social graph, two users are similar if they have friends in common, and in the other graph two users are similar if they are related to users that visited the same places. Equations 10 and 11 show how to compute the scoring functions for these strategies, where $N(u)$ and $N(v)$ are the neighbors of u and v in the graph, respectively.

$$s6(u,v,i) = s(u,v) = \sigma_{jaccard(u,v)} = \frac{|N(u) \cap N(v)|}{|N(u) \quad N(v)|} \tag{10}$$

$$s7(u,v,i) = s(u,v) = \sigma_{coseno(u,v)} = \frac{|N(u) \cap N(v)|}{\sqrt{|N(u)||N(v)|}} \tag{11}$$

Both Jaccard and Cosine local similarity can be also applied to calculate the similarity of two users in the graph of common visited placed. We denote these strategies $s8(u,v,i)$ and $s9(u,v,i)$, respectively.

2.2.5 Summary of strategies

To sum up, the strategies we propose and analyze in this article are summarized below:

- $s1(u,v,i)$ generates positive and negative preference matrices based on the discretization of the preference values. This scoring function uses only the positive sentiments, favoring users having coincidences in the positive matrix.
- $s2(u,v,i)$ generates positive and negative preference matrices based on the discretization of the preference values. This scoring function uses both sentiments, favoring users having coincidences in both positive and negative matrices.
- $s3(u,v,i)$ considers areas of POIs visited more frequently by users, favoring users having bigger area intersection, i.e. the closer a person is the more important its opinions.
- $s4(u,v,i)$ considers the hierarchically organized POIs categories and the number of coincidences in the lowest-level categories of POIs visited by the two users.
- $s5(u,v,i)$ considers the hierarchically organized POIs categories as well as the coincides of users in the highest-level category the target item belongs to.

- $s6(u,v,i)$ calculates the Jaccard local similarity index over the social graph (friendship relationships).
- $s7(u,v,i)$ calculates the Cosine local similarity index over the social graph (friendship relationships).
- $s8(u,v,i)$ calculates the Jaccard local similarity index over the graph representing common visited places.
- $s9(u,v,i)$ calculates the Cosine local similarity index over the graph representing common visited places.

3 Experimental results

In this section, the different experiments we carried out to evaluate the performance of the different strategies are proposed. We compared the Mean-Absolute Error (Equation 12) for the different neighbor selection alternatives against the classical user-based collaborative filtering approach. First, in Section 3.1 we describe the characteristics of the dataset used. Then, in Section 3.2 we present the results obtained and analyze them.

$$MAE = \frac{1}{|T|} \sum_{(u,i)\in T} |\overline{r}_{ui} - r_{ui}| \tag{12}$$

The baseline chose for comparing the results of the defined strategies as well as the initial selection of neighbors is the traditional user-based CF approach in which the *k* most similar users are selected by calculating the cosine similarity of preferences between the target user and all users in the system and the selected users are weighted according to the same similarity. For calculating the MAE rates the preference matrix was divided into a 70% for training and 30% for testing, i.e. the error estimation is done over the last group considering the profiles in the former one.

3.1 Dataset description

For the experiments the dataset from [4] was used, containing data collected from one of the most widely used LBSN, *Foursquare*. The dataset has the following information: *Places*, information about the places visited; *Users*, data of the users using the system; *Tips*, information about check-ins made by users; *Friendship*, information on the social relationship between users, and *Categories*; information of the categories of *Foursquare* places. In the dataset there are users from all over the world, but for our experiments only users belonging to the state of New York were considered, as they are greater in quantity. Out of the 47,220 users in the dataset, the 27,000 users from New York were used.

As explained in Section 2, collaborative filtering relies on a rating matrix *M* where each cell represents the preference score (rating) given by a user *i* to item *j*. For extracting such a matrix from the dataset, preferences need to be inferred from the user actions in the dataset since there is not explicit rating given to places. User tips, however, constitute implicit information about the user's preferences and they can be processed in order to interpret the meaning of the text and the sentiment associated to it to obtain a preference score.

To this end, in this work we use an automatic sentiment analysis tool to obtain a numerical value denoting the users' opinion about the places they visited expresses in the tips left. Sentiment analysis or opinion mining is the computational study of people's opinions, appraisals, attitudes, and emotions toward entities, individuals, issues, events, topics and their attributes [21]. This discipline has a lot of activity in recent years since the proliferation of social media and their applications [2]. Given the practical implications of this task in the automatic analysis of the content generated in social media, such as reviews, forum discussions or posts, multiple tools have emerged for extracting sentiment from input texts.

User tips were processed to obtain the preference matrix. For each user, all tips belonging to the same place were concatenated into a single text for tuning the sentiment analysis tool. TextBlob[2] tool was used to obtain the opinion or sentiment corresponding to the text extracted from tips. This tool was used as it offers a simple API for diving into common natural language processing (NLP) tasks. TextBlob is a python library that among other functionalities, contains a sentiment analysis function to extract the sentiment of a given text. The sentiment is expressed in a [−1, 1] range, where -1 means that the sentiment or opinion is negative, and 1 means that the sentiment or opinion is positive. Values between the extremes, indicate different degrees of positiveness or negativeness. Finally, the sentiments were mapped to a five-point scale for completing matrix M. Table 1 shows how the values for sentiments are discretized.

Table1
Discretization of sentiment analysis values extracted from tips

Sentiment value	Preference value
[-1,-0.6)	1
[-0.6,-0.2)	2
[-0.2,0.2)	3
[0.2,0.6)	4
[0.6,1]	5

The process of transforming the preference matrix obtained as described into a graph of common visiting places, illustrated in Figure 1, results in a network containing 37,679 nodes and 440,436 edges. Experiments were run for evaluating the selection of neighbors over this graph and, then, the impact of the different scoring functions for the estimation of preferences.

[2] https://textblob.readthedocs.io/en/dev/index.html

3.2 Results

In the graph of common visiting places built as described in the previous sections, the selection of neighbors is first addressed. For these experiments, different neighborhood sizes were considered, from 5 users up to 300 users. This upper limit was chosen as results tend to stabilize as the neighborhood size grows. Figure 4 shows the results achieved by selecting users at different depth levels. Also, the results are summarized for different neighborhood sizes. In the figure, it is possible to observe that independently of the depth explored, the selection of neighbors based on the graph outperformed the baseline. The best results were obtained using relationships of level 1, i.e. considering the opinion of users that visited the same places as the target user. After level 1, the selection of neighbors does not achieve better results in terms of prediction. Figure 5 shows the average number of users extracted from the graph for determining the neighborhood of each target user. From level 2 to 5 the number of users involved in preference estimation during prediction is higher, but this does not cause a reduction of estimation errors. In average, in the first level an average of 27 users are considered, whereas at level 5 an average of 31661 users are extracted from the graph for making a prediction.

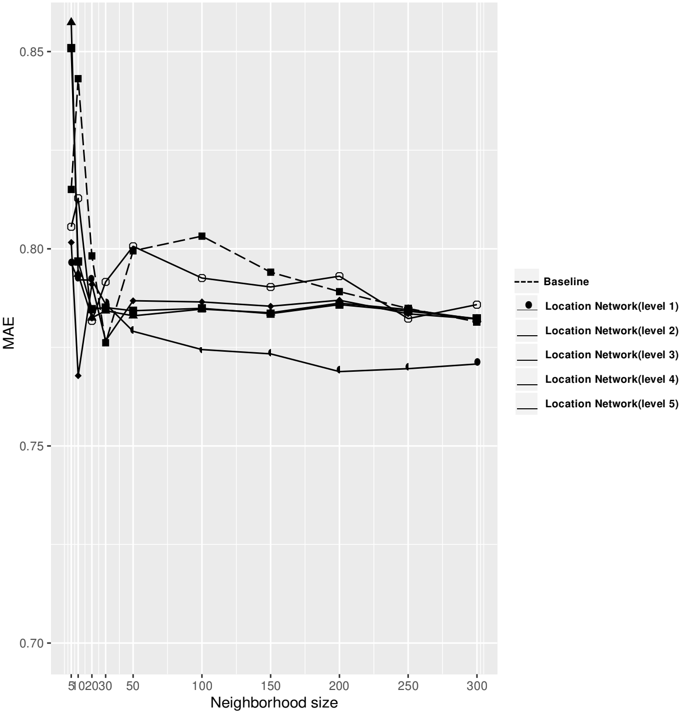


Figure 4
Results of selecting neighbors in the graph of common visited places

Scoring strategies are compared after selecting neighbors from the graph of common visited locations. Figure 6 shows the results obtained using as neighbor selection the

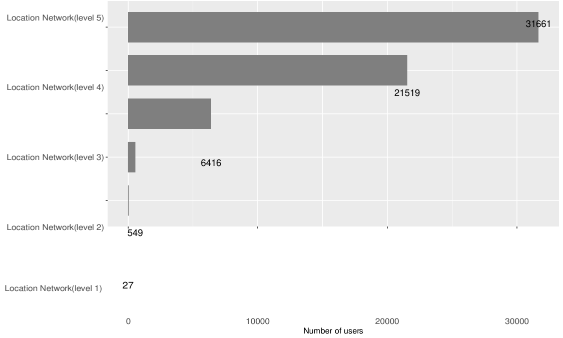


Figure 5
Average number of users compared when selecting the neighborhood

network of common visited places at different levels (from 1 to 5) and the different scoring strategies proposed in this article, compared against the baseline obtained by selecting the *k* most similar neighbors using cosine similarity.

In general, we can observe that the deeper the network is explored the higher the MAE, as noticed in the previous experimentation. This could be due to the fact that considering more neighbors in the preference estimation introduces some noise in the calculus. We can also observe that in level 1 of the network, i.e. direct relationships, all the scoring strategies have less MAE than the baseline, being strategies *s*5 and *s*2 the ones that produce less error differences. In the remaining levels, most of the strategies are worse than the baseline. Strategy *s*5 considers the item category, thus the weight that of each neighbor considered is more specific regarding the item. Strategy *s*2 takes into account the positive and negative coincidences in user preferences, thus a user that has more coincidences both in places and in opinions with the target user has a higher weight.

Regarding level 2 in the network, we can observe that strategies *s*5 and *s*2 achieved less error values than the baseline, but the other scoring strategies increased their error being even higher than the baseline in some configurations. Also, in level 2 the best overall results are achieved for strategy *s*2, implying that including more users (an average of 549 in such level), but appropriately weighted, can lead to a better prediction. In level 3, the tendency is similar than in level 2. In levels 4 and 5 in the network, *s*2 still has a less MAE than the baseline, but *s*5 increased its value being similar to the baseline and other strategies.

4 Related works

POIs recommendation plays an important role in LBSNs as it helps users discover and explore new attractive locations taking advantage of the community-contributed data, such as friendship links between users, check-ins on points-of-interest, comments, geographical information and categories of POIs, all of them reflecting user preferences. LBSNs, therefore, open new possibilities and challenges for recommender systems [5].

Several POI recommendation systems have been proposed in the literature stem-

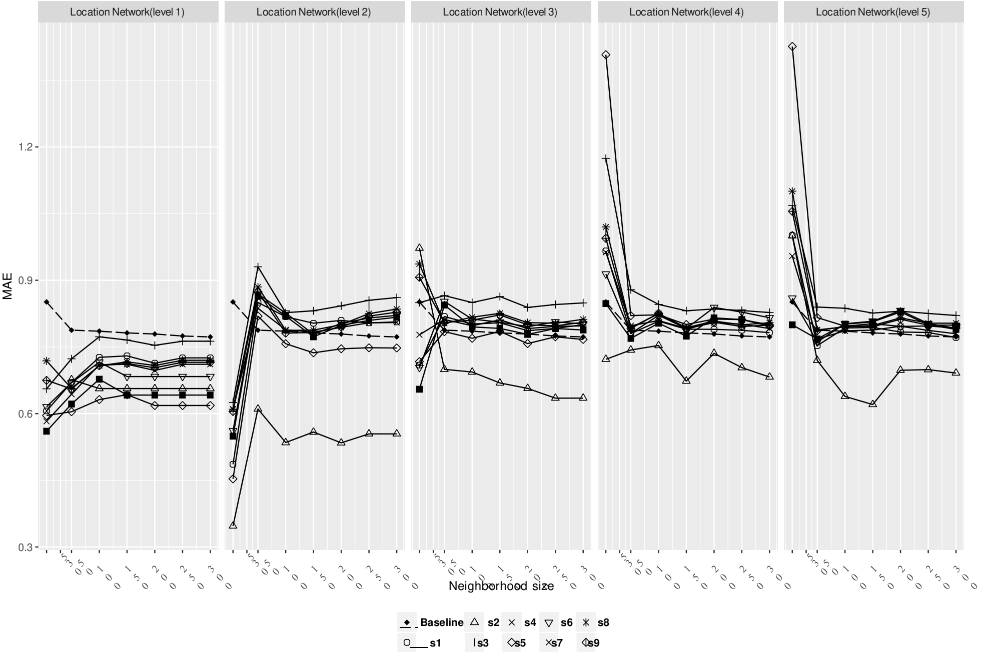


Figure 6
Results of the different neighbor weighting strategies

ming from traditional techniques from the area of recommender systems such as content-based, collaborative filtering and hybrid approaches. LCARS [36], a location-content-aware recommender system, exploits the content information about a user preferred spatial items to produce recommendations in other cities. In [7], the authors propose an approach for detecting the current user context, inferring possible leisure activities and recommend appropriate content on the site (shops, parks, movies). [34] explore text descriptions, photos, user check-in patterns, and venue context for defining a location semantic similarity for venue recommendation. The Sequential PersOnalized spatial items REcommender system (SPORE) [33] fuses the sequential influence of visited spatial items and the personal interests of individual users using a novel latent variable topic-region model. In [19, 18] the problem of cold-start (i.e., recommending locations to new users) is addressed with a hybrid content-aware collaborative filtering approach that exploits the rich semantics (e.g., tweets) that users often shared on social networks. [40] propose a cross-region collaborative filtering (CRCF), to address both long-term content preferences and short-term location preferences of users.

Specifically, in the context of collaborative filtering, there are works that translated user check-ins into a user-item matrix, where each row corresponds to a user visiting history and each column is a POI. LARS [16], a location-aware recommender system, deals with three types of location-based ratings: spatial ratings for non-spatial items, non-spatial ratings for spatial items, and spatial ratings for spatial items. Berjani et al. [8] applied a Regularized Matrix Factorization (RMF) technique for CF-based personalized recommendation of potentially interesting spots. [20] incorporates into the factorization model a spatial clustering phenomenon observed in human mobility behavior on LBSNs. Within the CF framework, [13] use highly available GPS trajectories to enhance visitors with context-aware POI recommendations and [42] extract the user travel experience in the target region to reduce the range of candidate POIs. [37] introduce the temporal behavior of users into a time-aware POI recommendation and [41] propose an opinion-based POI recommendation framework taking advantage of the user opinions on POIs expressed as text-based tips.

Regarding users' weighting, various authors have proposed methods for integrating information about user's connections into recommender systems in the form of "trust-based recommenders". Massa and Avesani [3, 24] proposed a method for using users' statements of their trust in other users' opinions to weight user ratings by estimated trust rather than similarity in user– user CF when producing predictions and recommendations and built a ski mountaineering site around it. Golbeck [10] used a similar integration method, based on a different trust estimation algorithm, for movie recommendation. In [14] the authors present a variation of k-NN, the trusted k-nearest recommenders (or k-NR) algorithm, which allows users to learn who and how much to trust one another by evaluating the utility of the rating information they receive. This way users are no longer weighted according to how close to like-minded are, but according to the quality of the information that they exchange with each other. Rating information is weighted according to trust, a value that reflects a history of interactions rather than a history of similar ratings. There is continued work, however, on various methods for estimating and propagating trust through social networks [3, 24, 26, 43]. Guy et al. [12] also found that users tend to find recommendations of web sites, discussion forums, and other social software more interesting when they were recommended from the user's social connections rather than users with similar preference histories.

Particularly in the context of LBSN, the influence that some users may have on location recommendations for other users on account of social or spatial relationships has been addressed in different works. In [35] the authors consider the social and spacial influence of users under the framework of user-based CF, introducing this influence into a model-based method (a Bayesian CF algorithm). Trust and distrust relationships are used to identify friends for recommendation rather than considering all users. The authors found that geographical influence has a significant impact on the accuracy of POI recommendations, whereas the social friends contribute little to the accuracy. Liu et al. [22] incorporated instance and region level of geographical neighborhood characteristics into the learning of latent features of users and locations. Gao et al. [9] propose the concept of geo-social correlations of users' check-in activities, which considers both social networks and geographical distance to model four types of social correlations (i.e., local friends, distant friends, local non-friends and distant non-friends). These correlations are used for solving the "cold start" location prediction problem. In [32] a local context is defined, modeling the correlation between users and their friends, and a global context, denoting the reputation of users in the social network that is employed to weight the importance of user ratings. Li et al. [17] define three types of friends in LBSNs, social friends, location friends, and neighboring friends. Then, a two-step framework leverage the information of friends to improve POI recommendation in the context of a matrix factorization model.

In contrast to the described works, this paper aims to assess the impact of the different strategies proposed in the scoring of neighbors in user-based CF-based POI recommendation. In a previous work [30], strategies for selecting neighbors were studied, showing the advantage of building a graph of common visited places. The scoring strategies evaluated considered the different elements available in LBSNs, such as places categories and social information.

5 Conclusions

In this article, we have proposed different strategies for neighbor scoring in the context of collaborative filtering for the recommendation places of interest (POIs) in LBSNs. We have combined these strategies with a neighbor selection approach that is based on a graph of common visited places. This strategy outperformed others in the experimental evaluation reported in a previous work [30]. We carried out different experiments to evaluate the strategies proposed against a traditional user based collaborative filtering approach. Some of the strategies obtained very low errors values in the estimation of user preferences such as the strategy that considers the hierarchically organized POIs categories as well as the coincides of users in the highest-level category the target item belong to, and the strategy that is based on coincidences in positive and negative sentiment matrices based on the discretization of the preference values. These findings might be useful for recommender system developers in the context of LBSNs, since they can consider our results to prioritize different aspects or dimensions present in these networks to make recommendations. In future works, we plan to extend the experimentation on larger-scale datasets as well as incorporate other possible elements available in different location-based so- cial networks.

Acknowledgements

The work described in this article has been partially funded by ANPCyT, Argentina, through project PICT 2011-0366.

References

[1] G. Adomavicius and A. Tuzhilin: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, IEEE Transactions on Knowledge and Data Engineering, 2005, 17(6), pp. 734–749.

[2] O. Appel, F. Chiclana, and J. Carter: Main concepts, state of the art and future research questions in sentiment analysis, Acta Polytechnica Hungarica, 2015, 12(3), pp. 87–108.

[3] P. Avesani, P. Massa, and R. Tiella: A trust-enhanced recommender system application: Moleskiing, In Proceedings of the 2005 ACM Symposium on Applied Computing (SAC 2005), Santa Fe, New Mexico, USA, 2005, pp. 1589–1593.

[4] J. Bao, Y. Zheng, and M. Mokbel: Location-based and preference-aware recommendation using sparse geo-social networking data, In Proceedings of the 20th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '12), Redondo Beach, USA, 2012, pp. 199–208.

[5] J. Bao, Y. Zheng, D. Wilkie, and M. Mokbel: Recommendations in location-based social networks: A survey, Geoinformatica, 2015, 19(3), pp. 525–565.

[6] A. Bellogín, P. Castells, and I. Cantador: Neighbor selection and weighting in user-based collaborative filtering: A performance prediction approach, ACM Transactions on the Web, 2014, 8(2), pp. 12:1–12:30.

[7] V. Bellotti, B. Begole, E. H. Chi, N. Ducheneaut, J. Fang, E. Isaacs, T. King, M. W. Newman, K. Partridge, B. Price, P. Rasmussen, M. Roberts, D. J. Schiano, and A. Walendowski: Activity-based serendipitous recommendations with the Magitti mobile leisure guide, In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems Pages (CHI '08), Florence, Italy, 2008, pp. 1157–1166.

[8] B. Berjani and T. Strufe: A recommendation system for spots in location-based online social networks, In Proceedings of the 4th Workshop on Social Network Systems (SNS '11), Salzburg, Austria, 2011, pp. 4:1–4:6.

[9] H. Gao, J. Tang, and H. Liu: gSCorr: Modeling geo-social correlations for new check-ins on location-based social networks, In Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM '12), Maui, Hawaii, USA, 2012, pp. 1582–1586.

[10] J. Golbeck: Generating predictive movie recommendations from trust in social networks, In Proceedings International Conference on Trust Management, Lecture Notes in Computer Science, Springer, 2006, Vol. 3986, pp. 93–104.

[11] J. Golbeck: Introduction to social media investigation: A hands-on approach, chapter Foursquare, Syngress Publishing, 2015, pp. 101–113.

[12] I. Guy, N. Zwerdling, D. Carmel, I. Ronen, E. Uziel, S. Yogev, and S. Ofek-Koifman: Personalized recommendation of social software items based on social relations, In Proceedings of the 3rd ACM Conference on Recommender Systems (RecSys '09), New York, NY, USA, 2009, pp. 53–60.

[13] H. Huang and G. Gartner: Using context-aware collaborative filtering for POI recommendations in mobile guides, In Proceedings of the 8th International

Symposium on Location-Based Services (LBS 2011), Vienna, Austria, 2011, pp. 131–147.

[14] N. Lathia, S. Hailes, and L. Capra: Trust-based collaborative filtering, In Joint iTrust and PST Conferences on Privacy, Trust Management and Security (IFIPTM), 2008, Vol. 263, pp. 119–134.

[15] R. Lee, S. Wakamiya, and K. Sumiya: Discovery of unusual regional social activities using geo-tagged microblogs, World Wide Web, 2011, 14(4), pp. 321–349.

[16] J. J. Levandoski, M. Sarwat, A. Eldawy, and M. F. Mokbel: LARS: A location-aware recommender system, In Proceedings of the 2012 IEEE 28th International Conference on Data Engineering (ICDE '12), Washington, DC, USA, 2012, pp. 450–461.

[17] H. Li, Y. Ge, R. Hong, and H. Zhu: Point-of-interest recommendations: Learning potential check-ins from friends, In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16), San Francisco, CA, USA, 2016, pp. 975–984.

[18] D. Lian, Y. Ge, N. Jing Yuan, X. Xie, and H. Xiong: Sparse bayesian content-aware collaborative filtering for implicit feedback, In Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI-16), New York, NY, USA, 2016, pp. 1732–1738.

[19] D. Lian, Y. Ge, F. Zhang, N. Jing Yuan, X. Xie, T. Zhou, and Y. Rui: Content-aware collaborative filtering for location recommendation based on human mobility data, In Proceedings of the IEEE International Conference on Data Mining (ICDM 2015), 2015, pp. 261–270.

[20] D. Lian, C. Zhao, X. Xie, G. Sun, E. Chen, and Y. Rui: GeoMF: Joint geographical modeling and matrix factorization for point-of-interest recommendation, In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14), New York, NY, USA, 2014, pp. 831–840.

[21] B. Liu and L. Zhang: Mining Text Data, chapter A Survey of Opinion Mining and Sentiment Analysis, Springer, Boston, MA, USA, 2012, pp. 415–463.

[22] Y. Liu, W. Wei, A. Sun, and C. Miao: Exploiting geographical neighborhood characteristics for location recommendation, In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM '14), Shanghai, China, 2014, pp. 739–748.

[23] X. Long and J. Joshi: A HITS-based POI recommendation algorithm for location-based social networks, In IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013), Niagara, Canada, 2013, pp. 642–647.

[24] P. Massa and P. Avesani: Trust-aware collaborative filtering for recommender systems, Lecture Notes in Computer Science, Springer 2004, Vol. 3290, pp. 275–301.

[25] A. Papadimitriou, P. Symeonidis, and Y. Manolopoulos: Friendlink: Link prediction in social networks via bounded local path traversal, In International Conference on Computational Aspects of Social Networks (CASoN 2011), Salamanca, Spain, 2011, pp. 66–71.

[26] A. Popescul, L. H. Ungar, D. M. Pennock, and S. Lawrence: Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments, In Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence (UAI'01), San Francisco, CA, USA, 2001, pp. 437–444.

[27] D. Quercia, N. Lathia, F. Calabrese, G. Di Lorenzo, and J. Crowcroft: Recommending social events from mobile phone location data, In IEEE 10th International Conference on Data Mining (ICDM 2010), Sydney, Australia, 2010, pp. 971–976.

[28] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl: Grouplens: An open architecture for collaborative filtering of netnews, In Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work (CSCW '94), 1994, pp. 175–186.

[29] F. Ricci, L. Rokach, and B. Shapira: Recommender Systems Handbook, chapter Introduction to recommender systems handbook, Springer, 2011, pp. 1–35.

[30] C. Rios, S. Schiaffino, and D. Godoy: On the impact of neighborhood selection strategies for recommender systems in LBSNs, In Proceedings of the 15th Mexican International Conference on Artificial Intelligence (MICAI 2016), Cancun, Mexico, 2016.

[31] S. Scellato, A. Noulas, and C. Mascolo: Exploiting place features in link prediction on location-based social networks, In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11), San Diego, CA, USA, 2011, pp. 1046–1054.

[32] J. Tang, X. Hu, H. Gao, and H. Liu: Exploiting local and global social context for recommendation. In Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI '13), Beijing, China, 2013, pp. 2712–2718.

[33] W. Wang, H. Yin, S. Sadiq, L. Chen, M. Xie, and X. Zhou: SPORE: A sequential personalized spatial item recommender system, In Proceedings of the IEEE 32nd International Conference on Data Engineering (ICDE 2016), Helsinki, Finland, 2016, pp. 954–965.

[34] X. Wang, Y-L. Zhao, L. Nie, Y. Gao, W. Nie, Z-J. Zha, and T-S. Chua: Semantic-based location recommendation with multimodal venue semantics, IEEE Transactions on Multimedia, 2015, 17(3), pp. 409–419.

[35] M. Ye, P. Yin, W-C. Lee, and D-L. Lee: Exploiting geographical influence for collaborative point-of-interest recommendation, In Proceedings of the 34th

International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ′11), Beijing, China, 2011, pp. 325–334.

[36] H. Yin, Y. Sun, B. Cui, Z. Hu, and L. Chen: LCARS: A location-content-aware recommender system, In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ′13), Chicago, IL, USA, 2013, pp. 221–229.

[37] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. Magnenat-Thalmann: Time-aware point-of-interest recommendation, In Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ′13), 2013, pp. 363–372.

[38] R. Zafarani, M. Ali Abbasi, and H. Liu: Social Media Mining: An Introduction, Cambridge University Press, 2014.

[39] A. Zanda, E. Menasalvas, and S. Eibe: A social network activity recommender system for ubiquitous devices, In Proceedings of the 11th International Conference on Intelligent Systems Design and Applications (ISDA 2011), Cordoba, Spain, 2011, pp. 493–497.

[40] C. Zhang and K. Wang: POI recommendation through cross-region collaborative filtering, Knowledge and Information Systems, 2016, 46(2), pp. 369–387.

[41] J-D. Zhang, C-Y. Chow, and Y. Zheng: ORec: An opinion-based point-of-interest recommendation framework, In Proceedings of the 24th ACM International Conference on Information and Knowledge Management (CIKM 2015), Melbourne, Australia, 2015, pp. 1641–1650.

[42] E. Zhou, J. Huang, and X. Xu: A point-of-interest recommendation method based on user check-in behaviors in online social networks, In Proceedings of the 4th International Conference on Computational Social Networks (CSoNet 2015), Lecture Notes in Computer Science, Springer 2015, Vol. 9197, pp. 160–171.

[43] C-N. Ziegler and G. Lausen: Propagation models for trust and distrust in social networks, Information Systems Frontiers, 2005, 7(4), pp. 337–358.

Bipolar Rating Scales: A Survey and Novel Correlation Measures Based on Nonlinear Bipolar Scoring Functions

Ildar Batyrshin[1], Fernando Monroy-Tenorio[1], Alexander Gelbukh[1], Luis Alfonso Villa-Vargas[1], Valery Solovyev[2], Nailya Kubysheva[2]

[1] Centro de Investigación en Computación (CIC), Instituto Politécnico Nacional, Av. Juan de Dios Bátiz, C.P 07738, DF, Mexico

[2] Kazan Federal University, 18 Kremlyovskaya Street, Kazan 420008, Russian Federation

batyr1@cic.ipn.mx, b130294@sagitario.cic.ipn.mx, {gelbukh, lvilla}@cic.ipn.mx, {valery.solovyev, NIKubysheva}@kpfu.ru

Abstract: A bipolar rating scale is a linearly ordered set with symmetry between elements considered as negative and positive categories. First, we present a survey of bipolar rating scales used in psychology, sociology, medicine, recommender systems, opinion mining, and sentiment analysis. We discuss different particular cases of bipolar scales and, in particular, typical structures of bipolar scales with verbal labels that can be used for construction of bipolar rating scales. Next, we introduce the concept of bipolar scoring function preserving linear ordering and the symmetry of bipolar scales, study its properties, and propose methods for construction of bipolar scoring functions. We show that Pearson's correlation coefficient often used for analysis of relationship between profiles of ratings in recommender systems can be misleading if the rating scales are bipolar. Basing on the general methods of construction of association measures, we propose new correlation measures on bipolar scales free from the drawbacks of Pearson's correlation coefficient. Our correlation measures can be used in recommender systems, sentiment analysis and opinion mining for analysis of possible relationship between opinions of users and their ratings of items.

Keywords: rating scale; bipolar scale; recommender system; opinion mining; sentient analysis; correlation; association measure

1 Introduction

In psychology, sociology, medicine, and other fields, rating scales in numerical or verbal form are often used for measuring human preferences, traits, abilities, and attitudes in order to attribute values of items[1] [37, 20, 22, 23, 19, 12]. Different definitions of rating scale have been proposed. Presently, it is more common to consider rating scale to be a linearly ordered set of categories. The number of categories used in rating scales usually varies from 3 to 11 [20, 22, 17]. A set of rating scales can be aggregated in a scale of higher level [20].

Rating scales often have bipolar structure: two poles and opposite categories symmetrically located at the opposite sides of the scale [37, 20, 23, 27, 24, 3, 35, 11]. In such bipolar scales, the negative side of the scale is the inverse mirror of its positive side [11]. An object is evaluated in bipolar scale as being either positive, negative, or neutral.

In many applications of rating scales, a human is considered as a gauge for measuring and rating the strength of the attitudes or attribute values; the results of such "measurements" are usually represented by numbers. The theory of measurement [25, 30] studies the classes of operations allowed on the sets of such numbers, defining ordinal, interval, and ratio measurement scales. Although rating scales typically have symmetric bipolar form defined by a pair of polar terms [23], this symmetry usually is not considered explicitly in measurement scales. Due to the increasing interest in application of bipolar rating scales in recommender systems and sentiment analysis [28, 1, 29, 21, 34, 9, 26], the problem of consideration of scoring functions defined on bipolar scales explicitly taking into account the symmetry of these scales, which we study in this paper, is of particular interest.

As an alternative to the measurement-based approach to analysis of human attitudes, the model-based approach considers models of words, human verbal evaluations of attributes, sentiment scores, and ratings. For example, in fuzzy logic and soft computing [40, 18], the words representing the strength of the attributes, such as *small* or *very good*, are modeled by fuzzy sets and are often defined by parametric membership functions: triangular, trapezoidal, etc. In fuzzy logic based models of decision making, classification, or control, linguistic terms represented by fuzzy sets are usually considered as components of fuzzy inference systems; these fuzzy sets can be adjusted by a machine-learning procedure for the system to provide the optimal, or reasonable, solutions [18]. Herrera and Herrera-Viedma [16] consider an ordered structure of linguistic terms with the negation operator defined on the set of indexes. The semantics of a linguistic term defined in [16] is given by fuzzy sets defined on the segment [0, 1], distributed on it symmetrically or non-symmetrically. Xu [39] considers so-called additive

[1] In this paper, we usually list references in chronological order.

linguistic evaluation scales, where scores are given by fractional indexes of linguistic labels and bipolarity of the scale is represented by the negation also defined on the set of indexes. In recommender systems and collaborative filtering, utility function is typically represented by ratings [1]. The rating scales usually have bipolar form, however, generally speaking, utility can be an arbitrary function. Extrapolations from known to unknown ratings in recommender systems are usually done by (a) specifying heuristics that define the utility function and empirically validating their performance, and (b) estimating the utility function that optimizes certain performance criterion, such as the mean square error [1]. Breese et al. [8] consider two alternative probabilistic models for collaborative filtering: Bayesian classifier and Bayesian networks. Machine-learning algorithms are applied to learn parameters of the models. Liu and Seneff [21] consider the problem of assigning scores for the degree of polar sentiment to phrases related with the considered aspects of items. Based on these scores, they calculate an average rating for the aspect. They proposed linear additive model for calculating the scores of polar sentiments taking values in a positive scale. Taboada et al. [34] present lexicon-based approach to extracting sentiment from texts. They use dictionaries of words annotated with their semantic orientation: polarity and strength. The semantic orientation of words takes positive and negative values. It is supposed that some problems of its calculation could be resolved by fine-tuning sentiment orientation values and modifiers [34].

Several online resources have been developed in recent years for sentiment analysis of texts. For example, SenticNet [9] provides polarity scores for 30,000 concepts [26]. The scores range from −1 (bad) to +1 (good), with neutral scores being around zero.

To analyze the similarity between lists of ratings in recommender systems, Pearson’s correlation coefficient is often used [29]. However, as we will show, this correlation coefficient can be misleading in analysis of ratings from bipolar scales. Shardanand and Maes [32] proposed to use “constrained” Pearson’s coefficient *r* to take into account bipolarity of the rating scale.

In this paper, we give a survey of bipolar rating scales. We consider typical structures of bipolar scales with verbal labels that can be used for construction of bipolar rating scales. We introduce the concept of the bipolar scoring function preserving the linear ordering and the symmetry of bipolar scales, study its properties, and propose methods for construction of bipolar scoring functions. We show that Pearson’s correlation coefficient often used for analysis of relationships between profiles of ratings in recommender systems can be misleading if the rating scales are bipolar. Basing on general methods of construction of association measures, we propose new correlation measures on bipolar scales free from the drawbacks of Pearson’s correlation coefficient.

The paper is organized as follows. In Section 2, we survey different types of bipolar scales discussed in the literature. In Section 3, we present some typical structures of finite fully-labeled bipolar verbal rating scales. In Sections 4 and 5, we consider formal definitions of bipolar rating scales, bipolar scoring functions, and their properties. In Section 6, we propose novel methods of construction of bipolar scoring functions. In Section 7, we introduce new correlation measures on bipolar rating profiles. Finally, in Section 8 we give some discussion and conclusions.

2 Types of Bipolar Scales

Generally speaking, a bipolar scale is a linearly ordered set with minimal and maximal elements considered as two poles of the scale. These poles can be interpreted as negative and positive poles, correspondingly. Usually, bipolar scales have a neutral category located in the center of the scale, and all other categories between the neutral category and the negative and positive poles can be considered as negative and positive categories, correspondingly. In addition to the linear ordering of the bipolar scales, the scale is symmetric: the negative categories are mapped to the corresponding opposite positive categories, and vice versa. The elements of bipolar scales can have numerical scores and linguistic labels.

The symmetry of the bipolar scales can be reflected in the symmetry of the labels or the scores. Typical examples of bipolar scales containing three or five possible responses on questions are considered in [20]. One of these scales with bipolar structure has five categories: *strongly approve*, *approve*, *undecided*, *disapprove*, *strongly disapprove*, with the corresponding scores 5, 4, 3, 2, 1, or scores in reverse order. This scale has a neutral category *undecided* and symmetry between positive and negative categories. Currently, it is more popular to order the scale categories from negative to positive, for example, as follows: *strongly disagree*, *disagree*, *neither agree nor disagree*, *agree*, *strongly agree*, or in correspondence with the ordering of the scores: 1, 2, 3, 4, 5.

Bipolar scales of the following types have been considered in the literature; references given here contain corresponding examples:

(a) scales with a finite number n of categories [20],
(b) scales with infinite number of categories, e.g., in the range from 0 to 1, from −1 to 1, from −10 to 10 [16, 14, 13];
(c) 2-point scale ($n = 2$) [31] or multipoint scale ($n > 2$);
(d) scales with a neutral category [20] or without neutral category [10];
(e) scales with verbal labels only at poles [23] or fully labeled when all response categories are explicitly labeled [20, 38];
(f) scales with symbolic labels of poles [24];

(g) scales with numeric scores explicitly given [20] or not, e.g., given graphically with intervals [23];

(h) scales with positive numeric scores, e.g., 1, …, 5, or from 0 to 1 [20, 16] or with both negative and positive scores, e.g., from −1 to 1 [14, 9];

(i) scales without polarity of verbal labels; e.g., one of the Likert rating scales [20] contains responses: *grade school*, *junior high school*, *high school*, *college*, *graduate and professional school* with the scores 1, 2, 3, 4, 5 typical for bipolar scales;

(j) scales with polarity of verbal categories, but without symmetry with respect to the neutral category, i.e., when some positive verbal category or concept has no corresponding opposite negative verbal category [9, 26];

(k) scales explicitly using negation operation [40, 16, 39] or not [20];

(l) scales with non-numeric scores, e.g., with fuzzy sets [40, 16];

(m) scales with verbal categories ordered from negative categories on the left to positive categories on the right, or in reverse order [20];

(n) scales with numeric scores increased from left to right: 1, …, 5 or in reverse order: 5, …, 1 [20];

(o) scales with scoring function being linear [20] or nonlinear [34, 39] with respect to the indexes of the categories;

(p) scales with several verbal labels or concepts assigned to one gradation of the scale, due to synonymy or labels having equal score value [9, 29];

(q) positive and negative categories separated in two scales [11, 36];

(r) scales in which an attitude or attribute can have a positive and a negative polarity degree at the same time, e.g., degree of membership and degree of non-membership [2, 11];

(s) scales with aggregation of scores from the same scale [37, 14, 4];

(t) scales with aggregation of scores from different scales [20, 21];

(u) scales with calculation of the group opinion as an aggregation of the scores of individuals [37];

(v) scales with calculation of individual scores in higher-level scale as aggregation of individual scores in particular scales [20];

(w) scales measuring [20] or modeling [16, 1] attitudes, preferences, ratings, or utilities.

Below, we consider finite bipolar scales with neutral element as linearly ordered sets with symmetry given by the negation operation. In the following section, we will discuss typical structures of finite bipolar scales fully labeled with verbal labels. Further, we will formally consider two types of mutually related finite bipolar scales with neutral element as sets of integer indexes of categories of bipolar scales. We introduce bipolar scoring functions defined on these sets as models of user preferences.

3 Bipolar Verbal Rating Scales

A verbal rating scale is *bipolar* if it is symmetric with respect to the opposite categories, located on the opposite sides of the scale. Such symmetry can be expressed by a *negation operation N* defined on the scale. Consider an example of a 5-point bipolar scale with verbal labels ordered from left to right: *never* < *seldom* < *sometimes* < *often* < *always* and with negation *N*(*never*) = *always*, *N*(*seldom*) = *often*, *N*(*sometimes*) = *sometimes*, *N*(*often*) = *seldom*, *N*(*always*) = *never*. This negation operation has several formal properties. It is *involutive*, i.e., double negation of any category gives the same category, e.g., *N*(*N*(*seldom*)) = *seldom.* It is *decreasing*, i.e., if a category c_2 has greater rating than a category c_1: $c_1 < c_2$, then the negation of c_1 has greater rating than the negation of c_2: $N(c_1) > N(c_2)$. For example, *never* < *often* implies *N*(*never*) = *always* > *N*(*often*) = *seldom.* Note that this negation operation differs from the linguistic *not* [21, 34].

If the scale has an odd number of gradations, then there is a *center*, *midpoint*, *or neutral category C* in the scale such that $N(C) = C$. Generally, such point is called the *fixed point* of the negation. For the scale considered above, we have *C* = *sometimes.*

Consider a typical structure of 7-point verbal bipolar scale [3]:

$$L = (eap, vap, ap, np, p, vp, ep). \tag{1}$$

L is ordered from left to right: *eap* < *vap* < … < *vp* < *ep*, where *p* and *ap* (*anti-p*) denote opposite adjectives or attributes, correspondingly, and other letters denote: *e* = *extra*, *v* = *very, n* = *neutral*. The negation defines a mapping between the opposite categories, for example: *N*(*eap*) = *ep*, *N*(*vap*) = *vp*, …, *N*(*ep*) = *eap*, with *N*(*np*) = *np* for the neutral category *np.* Here is an example of a bipolar verbal scale with the structure (1):

L = (*awful*, *very bad*, *bad*, *neither good nor bad, good, very good, excellent*),

with *p* = *good, ap* = *bad.* One can consider the structure (1) as a cupboard with shelves with the labels *eap*, *vap*, … , *ep* ordered from the bottom upward. In a particular application, each “shelf” of (1) can be filled by corresponding verbal labels. These shelves can have scores 1, 2, 3, 4, 5, 6, 7 or −3, −2, −1, 0, 1, 2, 3, with neutral category *np* having the score 4 or 0, respectively. In the following sections, we will consider these scores as indexes of the categories of the bipolar scale with $n > 3$ indexes, and the score function will be given as a numeric function defined on the set of these indexes. For identity scoring function, its values will coincide with the indexes, in our case, with 1, …, 7 or with −3, …, 3. Generally, it will be a nonlinear function preserving the symmetry of the bipolar scale.

The shelves of (1) can contain several verbal labels, which are considered synonymous or having equal scores. For example, the shelf *eap* can contain the verbal labels *awful* and *terrible*. A 5-point bipolar scale can be obtained from the structure (1) by eliminating symmetric categories or by symmetric merging of neighboring categories. In such way, one can reduce the structure (1), for example, to the following one:

L= (*vap, ap, np, p, vp*), (2)

with indexes 1, 2, 3, 4, 5 or −2, −1, 0, 1, 2. In the reduced cupboard (2), the shelf *vp* can contain the verbal labels *very good, excellent, perfect,* etc. Deletion of the neutral category *np* will give bipolar verbal scale without center.

Another typical structure of the bipolar scale is as follows [3]:

L= (*hap, map, lap, np, lp, mp, hp*), (3)

where *h*, *m*, and *l* denote *high, middle* and *low* intensities, respectively, of the opposite attributes *ap* and *p*. The categories of the scale (3) are ordered from left to right: $hap < map < \ldots < mp < hp$. The negation defines a mapping between the opposite categories: *N*(*hap*)= *hp*, *N*(*map*) = *mp*, …, *N*(*hp*) = *hap*. Bipolar scales with the number of categories less than 7 can be obtained from (3) by deleting pairs of opposite categories.

Generally, the scale structures (1) and (3) can be extended until 15-points bipolar verbal scale by adding gradations with modifiers *el* (*extra low*)*, vl* (*very low*)*, vh* (*very high*)*, eh* (*extra high*) for adjective *p* and its opposite *ap* as follows:

L= (*ehap, vhap, hap, map, lap, vlap, elap, np, elp, vlp, lp, mp, hp, vhp, ehp*). (4)

Bipolar scales with the number of categories less than 15 can be obtained from (4) by deleting pairs of opposite categories. Below is an example of such 9-point bipolar scale:

L = (*dislike extremely, dislike very much, dislike moderately, dislike slightly, neither like nor dislike, like slightly, like moderately, like very much, like extremely*),

with the center *C* = *neither like nor dislike* and with the reduced structure of (4):

L= (*ehap, vhap, map, lap, np, lp, mp, vhp, ehp*).

An example of a bipolar verbal 13-point scale can be found in [15].

The considered structures are common for the scales used in sociology, psychology, and medicine. However, in opinion mining and in recommender systems the bipolar scales with symmetry of opposite categories can have categories expressed in a variety of forms. Below is Ringo’s scale for rating music [32] in music recommendation system, which is explicitly different from the structures (1) and (3) but implicitly has symmetric form, as in (1), with the center at the category 4:

L = (1. *Pass the earplugs*. 2. *Barely tolerable*. 3. *Eh. Not really my thing*. 4. *Doesn't turn me on, doesn't bother me.* 5. *Good Stuff.* 6. *Solid. They are up there*. 7. *BOOM! One of my FAVORITE few! Can't live without it*).

The considered typical structures of bipolar scales can be used for construction of verbal bipolar scales with symmetry of opposite categories. The verbal labels of these categories can differ from one application to another, but the symmetric structure should be preserved in order to consider the verbal rating scales as bipolar scales.

In the next section, we will give a formal definition of a finite bipolar scale as an ordered set of indexes of categories of a bipolar scale considered above. In the sequel, we will consider bipolar scoring functions that assign numerical values to each point of the bipolar scale, which will be used as a numerical model of this scale. The main properties of bipolar utility functions will also be connected with the negation operation related with the symmetry in the scores assigned to opposite categories of the bipolar scale.

4 Finite Bipolar Scales

Formally, a *bipolar* scale L with n ordered categories $c_1 < \ldots < c_n$ can be represented by an ordered set of indexes of these categories $J = \{1, \ldots, n\}$, $n > 1$, with the *negation* operation $N: J \rightarrow J$ defined by

$$N(j) = n + 1 - j \text{ for all } j \in J. \tag{5}$$

The negation function (5) is a *strictly decreasing*:

$$N(i) > N(j) \text{ if } i < j, \tag{6}$$

and *involutive*:

$$N(N(j)) = j, \quad \text{for all } j \in J. \tag{7}$$

We will assume that the bipolar scale has an odd number of elements, i.e., $n = 2m + 1$ for some positive integer m. In this case, the set J will have a *fixed point* of the negation N, i.e., an element C such that

$$N(C) = C. \tag{8}$$

The property (8) is fulfilled for a unique element

$$C = m + 1, \tag{9}$$

called the *center, neutral*, or *midpoint* of the bipolar scale. From (5) and (6), it follows that a bipolar scale with an even number of elements n has no center. Bipolar scales without center can be obtained from scales with center by deleting the center.

For example, the 5-point bipolar scale (*never, seldom, sometimes, often, always*) can be given by an ordered set of indexes $J = \{1, 2, 3, 4, 5\}$ and with the negation $N(j) = 6 - j$, such that $N(1) = 5$, $N(2) = 4$, $N(3) = 3$, $N(4) = 2$, $N(5) = 1$. This scale has a center $C = 3$.

From (5), $n = 2m + 1$, and (9), we obtain *bipolarity* properties:

$N(j) + j = 1 + n,$

and

$$N(j) + j = 2C, \text{ for all } j \in J. \tag{10}$$

Due to bipolarity, the elements $N(j)$ and j of the scale are symmetrically located with respect to the center and the "poles" 1 and n of the scale J:

$|j - C| = |N(j) - C|, \quad |j - 1| = |N(j) - n|, \quad \text{for all } j \in J.$

We call the ordered set $K = \{-m, \ldots, -1, 0, 1, \ldots, m\}$ the *centered form* of the bipolar scale $J = \{1, \ldots, 2m + 1\}$, $m > 0$. The negation operation $N: K \to K$ on K is defined by:

$$N(k) = -k, \qquad \text{for all } k \in K. \tag{11}$$

Unless it can cause confusion, we will use the same letter N for the negation on J and on K, using the arguments j or k, respectively. It is clear that N on K is a strictly decreasing and involutive function, i.e., $N(N(k)) = k$, for all $k \in K$. This scale has the center $C = 0$ with $N(C) = C = 0$, and the bipolarity (10) also fulfills for the scale K with negation (11):

$$N(k) + k = 2C, \quad \text{for all } k \in K. \tag{12}$$

We call the scale $K = \{-m, \ldots, m\}$, $m > 0$, with the negation defined by (11) a *centered bipolar scale.*

For example, the 5-point bipolar scale $J = \{1, 2, 3, 4, 5\}$ is represented in centered form as $K = \{-2, -1, 0, 1, 2\}$ with the negation N on K defined by $N(-2) = 2$, $N(-1) = 1$, $N(0) = 0$, $N(1) = -1$, $N(2) = -2$ and with the center $C = 0$.

The bipolar scales $J = \{1, \ldots, 2m + 1\}$, $m > 0$, and $K = \{-m, \ldots, m\}$ can be transformed one into the other as follows:

$$k = j - m - 1, \; j = k + m + 1, \quad \text{for all } j \in J \text{ and } k \in K. \tag{13}$$

5 Bipolar Scoring Functions

Here we study the properties of scoring (utility) functions defined on bipolar scales. The scoring functions give possibility to model users with different utility of categories of the same scale. For example, one user prefers to use the gradations of the scale near the poles but another user prefers to use the gradations near the center. For such users, the utility of the same gradations can be different. In addition, in the development of decision-making or recommender systems using bipolar scales, the utility of categories of the scales can be represented by non-linear functions modeling utility of categories in different manner, depending on the application or the task. For such functions, the difference between the neighboring gradations depends on their positions on the scale. For example, for the bipolar scale (*never, seldom, sometimes, often, always*), the difference between the utility of the categories *sometimes* and *often* can be modeled by the number 10, but the difference between the utilities of the categories *often* and *always* can be modeled by the number 50. Such nonlinear utility functions can be used, for example, for modeling a user's ratings in model-based approach to collaborative filtering in recommender systems [1].

In model-based approach to construction of bipolar utility functions, how the user measures these utilities is not important; what matters is how to model these utilities for different users and different tasks. In this approach, instead of the "adequate" measurement of a user's preferences or ratings, we concentrate on the effectiveness of the recommendations or decisions generated by the recommender or decision-making system using these bipolar utility functions. Generally, these utility functions can be parameterized and further tuned or adjusted by some machine-learning procedure for the system to obtain optimal or useful recommendations and decisions. For this, these utility functions should satisfy properties similar to the properties of the bipolar scales that we considered above. In this section, we will consider these properties of bipolar utility functions, and in the following section we will consider different methods of construction of such functions.

Let I denote a bipolar scale J or K with $n = 2m + 1$ categories, $m > 0$. For $J = \{1, \ldots, 2m + 1\}$, we have $C = m + 1$, $N(j) = n + 1 - j$ for all $j \in J,$ and for the scale $K = \{-m, \ldots, -1, 0, 1, \ldots , m\}$, we have $C = 0$, $N(k) = - k,$ for all $k \in K$. For both scales, we have $N(C) = C$. We call the values $P_1 = \min(I)$ and $P_2 = \max(I)$ the negative and the positive poles, correspondingly. For the scale J, we have $P_1 = 1$, $P_2 = n,$ and for the scale K, we have $P_1 = -m$, $P_2 = m.$ For both scales, we have $N(P_1) = P_2$ and $N(P_2) = P_1$.

Definition 1. Let I be a bipolar scale with negation N. A strictly increasing real function $U: I \to R$ is called a *scoring* or *utility function* on I. This function is called a *bipolar scoring function* (BSF) on I if it satisfies the condition

$$U(N(i)) + U(i) = U(P_1) + U(P_2), \qquad \text{for all } i \in I. \qquad \text{(bipolarity)} \qquad (14)$$

For the scales with the center C from $N(C) = C$, we have $U(P_1) + U(P_2) = 2U(C)$ and the bipolarity property can be given by:

$$U(N(i)) + U(i) = 2U(C), \qquad \text{for all } i \in I. \qquad \text{(bipolarity)} \qquad (15)$$

A BSF is called a *centered bipolar scoring function* (CBSF) if $U(C) = 0$.

From (14) and (15), we have:

$$|U(j) - U(C)| = |U(N(j)) - U(C)|,\ \ |U(j) - U(P_1)| = |U(N(j)) - U(P_2)|,\ \text{ for all } j \in J,$$

i.e., for a bipolar scoring function U, the utility values of the opposite categories $U(j)$ and $U(N(j))$ are at the equal distances from the utility value of the neutral category $U(C)$ and at the equal distances from the utility values of the poles.

The definition of the centered bipolar scoring function implies

$$U(N(i)) = -U(i), \qquad \text{for all } i \in I. \tag{16}$$

For a CBSF defined on a centered bipolar scale $K = (-m, \ldots, m)$, $m > 0$, we have: $U(0) = 0$, $U(k) > 0$ if $k > 0$, $U(k) < 0$ if $k < 0$, and

$$U(-k) = -U(k), \qquad \text{for all } k \in K. \tag{17}$$

CBSFs give natural models of utility of categories of bipolar verbal rating scales when these categories have negative and positive sentiments. For example, for the 5-point centered bipolar scale $K = \{-2, -1, 0, 1, 2\}$, one can define a CBSF $U(K) = \{-10, -4, 0, 4, 10\}$ preserving the sign and the symmetry of the bipolar scale K.

Proposition 1. If U is a BSF on I, then the function W: $I \rightarrow R$ defined by

$$W(i) = pU(i) + q \quad \text{for all } i \in I, \tag{18}$$

where $p, q \in R, p > 0$, is also a BSF on I.

Proof. It is clear that W is strictly increasing. Bipolarity of W follows from (18) and bipolarity (15) of U: $W(N(i)) + W(i) = pU(N(i)) + q + pU(i) + q = p(U(N(i)) + U(i)) + 2q = p(2U(C)) + 2q = 2(pU(C) + q) = 2W(C)$. □

In (18), the parameters p and q define scaling and shifting of the utility function U, correspondingly.

Proposition 1 implies that from any BSF U one can obtain a CBSF U_C as follows:

$$U_C(i) = U(i) - U(C) \quad \text{for all } i \in I. \tag{19}$$

From (5) and (9), it follows that the identity function $U(j) = j$, for all $j \in J$, is a BSF. We call this function the *standard bipolar scoring function* (SBSF). For example, SBSF on 7-point bipolar scale $J = \{1, 2, 3, 4, 5, 6, 7\}$ has the values $U(J) = J = \{1, 2, 3, 4, 5, 6, 7\}$. From this function, applying the transformation (18) with parameter values $p = 1$ and $q = -1$, one can obtain another popular bipolar scoring function $U(J) = \{0, 1, 2, 3, 4, 5, 6\}$.

These examples show that the majority of the popular rating scales, including Likert scales [20] and the scales used in recommender systems [32], use SBSFs. In this paper, we are mainly interested in nonlinear BSFs. For example, on 7-point bipolar scale $J = \{1, 2, 3, 4, 5, 6, 7\}$ with the center $C = 4$, a nonlinear bipolar scoring function can be defined as $U(J)= \{0, 15, 40, 50, 60, 85, 100\}$ with $U(C)=$

50. From this BSF, applying the linear transformation (19), one can obtain a CBSF $U_C(J) = \{-50, -35, -10, 0, 10, 35, 50\}$.

Note that using transformations (13) of the indexes of the bipolar scales $J = \{1, \ldots, 2m + 1\}$ and $K = \{-m, \ldots, m\}$, one can transform BSF U_J: $J \rightarrow R$ defined on J into a BSF U_K: $K \rightarrow R$ defined on K, and vice versa, as follows:

$$U_J(j) = U_K(j - m - 1), \quad U_K(k) = U_J(k + m + 1), \quad \text{for all } j \in J \text{ and } k \in K. \tag{20}$$

These functions have the same set of values: $U_J(J) = U_K(K)$. For example, the CBSF $U_K(K) = \{-10, -4, 0, 4, 10\}$ defined on $K = \{-2, -1, 0, 1, 2\}$ is transformed into CBSF $U_J(J)= \{-10, -4, 0, 4, 10\}$ defined on $J = \{1, 2, 3, 4, 5\}$. Such simple transformation can be used in the method of construction of CBSF on J considered in the following section by defining a CBSF as some odd function on K and then changing the set of its arguments (indexes) K by J.

6 Methods for Construction of Bipolar Scoring Functions

As we have shown in the previous section, a BSF on a bipolar scale is a linear transformation of an odd function with zero argument value in the center of the scale. The majority of the traditional rating scales can be considered as particular cases of BSFs when the scoring function is linear. In this section, we consider various heuristic methods of construction of bipolar utility (scoring) functions which are, generally, nonlinear.

6.1 Bipolar Utility Functions Based on the Distribution of a User's Ratings

We suggest a heuristic method for constructing bipolar utility functions using the distribution of bipolar scores obtained from a user's ratings. Our method is based on the observation that the greater the difference between the frequencies of two neighboring categories, the greater the difference in the utilities of these categories.

For simplicity, consider the categories in a centered bipolar scale $K = \{-m, \ldots, -1, 0, 1, \ldots, m\}$. Suppose the user has provided ratings of N items with frequencies $P= (P_{-m}, \ldots, P_{-1}, P_0, P_1, \ldots, P_m)$, with $P_{-m} + \ldots + P_m = N$, where P_k, $k \in \{-m, \ldots, m\}$, is the frequency of the category from the scale K. Our algorithm consists of the following steps.

1. Symmetrize the frequencies:

$PS_k = \frac{P_k + P_{-k}}{2}$, for all $k = -m,\ldots, m$.

2. Calculate the cumulative distribution function:

$$U_{-m} = 0, \quad U_{k+1} = U_k + c|PS_{k+1} - PS_k| + d, \qquad k = -m,\ldots, m-1, \tag{21}$$

where c is a scaling constant and $d = 0$ if $PS_{k+1} \neq PS_k$ for all $k = -m, \ldots, m-1$, otherwise $d > 0$; the constant d is introduced for all categories to have different scores and thus the utility function to be strictly increasing.

3. Normalize the utility function to obtain the values of utility function in the range $[0, M]$, or $[-M, M]$, $M > 0$ and, if necessary, move it from the scale $K = \{-m, \ldots, m\}$ to the scale $J = \{1, \ldots, 2m + 1\}$ by replacing indexes by (20).

It can be shown that the proposed method constructs a BSF satisfying the bipolarity property.

Figure 1 shows an artificial example of construction of bipolar utility functions based on the distribution $P = (5, 35, 48, 60, 30, 20, 2)$ of categories of 7-point bipolar scale in $N = 200$ ratings of a user. The left plot shows the original and symmetrized distributions and right plot shows the bipolar utility function constructed from the symmetrized distribution. We used $d = 0$ in (21) because in the symmetrized distribution all neighboring categories have different frequencies.

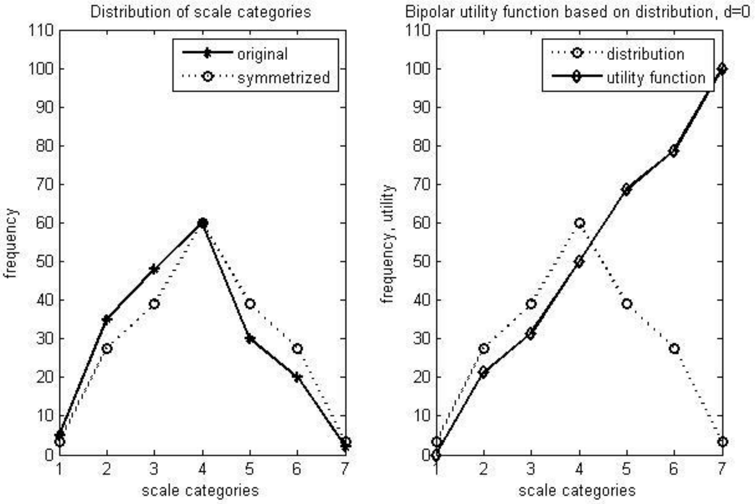


Figure 1.
Construction of bipolar utility function based on the symmetrized distribution of bipolar scale categories obtained in $N = 200$ ratings of a user, with $d = 0$

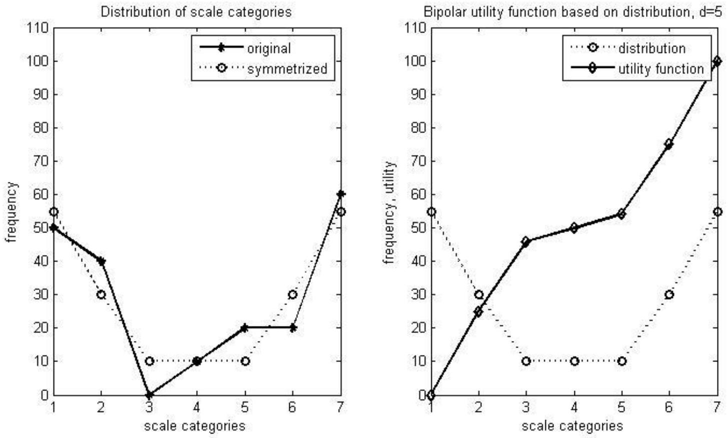


Figure 2
Construction of bipolar utility function based on symmetrized distribution of bipolar scale categories obtained in $N = 200$ ratings of a user, $d = 5$

Similarly, in Figure 2 the distribution is $P = (50, 40, 0, 10, 20, 20, 60)$. In this case, we used $d = 5$ because the symmetrized distribution contains neighboring categories with equal frequencies. In both cases, we used $c = 1$. All bipolar utility functions were normalized to have values in the interval [0,100].

6.2 Generator-Based Utility Functions

Now we suggest a method for constructing CBSF U on the bipolar scale $K = \{-m, \dots, -1, 0, 1, \dots , m\}$, $m > 0$. A bipolar utility function on J can be constructed using (18) and (20). Let G be a positive real value and g: $\{0, \dots, m\} \rightarrow [0, G]$ a strictly increasing function such that $g(0) = 0$, $g(m) = G$. We call this function a *generator* of the function W: $K \rightarrow [- G, G]$ defined by:

$$W(k) = g(k) \qquad \text{for all } k \in \{0, \dots, m\}, \tag{22}$$

$$W(k) = -g(-k) \qquad \text{for all } k \in \{- m, \dots, -1\}. \tag{23}$$

Obviously, the function W constructed by this method is a CBSF. In the model-based approach to modeling user preferences, we can define the generator g by a parametric function and use it to define parametric bipolar utility functions on K or on J.

Here are two examples of parametric generators of bipolar utility functions inspired by the parametric Sugeno negation used in fuzzy logic [33], for $p > -1$:

$$g_1(k) = \frac{Gk(1+p)}{m+pk}, \qquad k = 0, \dots, m, \tag{24}$$

$$g_2(k) = \frac{Gk}{m + pm - pk}, \quad k = 0, \ldots, m. \tag{25}$$

Figure 3 shows the shapes of CBSF W on a 7-point centered bipolar scale $K = (-3, -2, -1, 0, 1, 2, 3)$ obtained from the generators (24) and (25) for different values of the parameter p. For $p = 0$, both generators are linear: $g_1(k) = g_2(k) = Gk/m$.

The generator $g_1(k)$ with positive p and the generator $g_2(k)$ with negative p can be used for modeling bipolar utility functions when the scores are located nearer to the boundaries of the scale; see Figure 3. For such utility functions, the nearer the categories of the scale to the corresponding positive or negative pole, the smaller the difference between the utilities of the neighboring categories. For example, if on the 7-point scale

L = (*awful*, *very bad*, *bad*, *not good not bad, good, very good, excellent*)

we define a bipolar utility function with these properties, then the difference between the utilities of the categories *very good* and *excellent* will be small.

Conversely, the generator $g_1(k)$ with negative p and the generator $g_2(k)$ with positive p can be used for modeling bipolar utility functions when the scores are located near the center of the scale, i.e., the nearer the categories of the scale to the corresponding pole, the larger the difference between the utility values of the neighboring categories. For example, if on the same scale L we define a function with these properties, then the utility value of the category *excellent* will be much greater than that of the category *very good*.

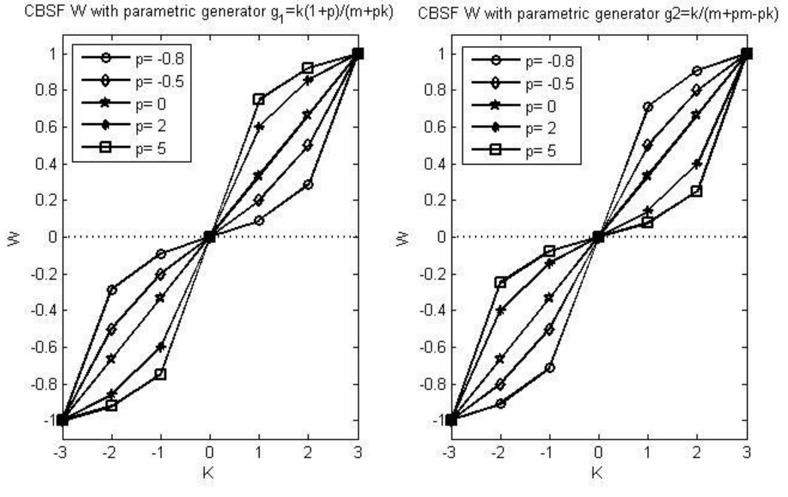


Figure 3

Examples of CBSF on 7-point centered bipolar scale generated by generators $g_1(k)$ (left) and $g_2(k)$ (right) for different values of parameter p

In the model-based approach to analysis of user opinions, the parameters of the utility functions can be tuned for the model to generate the best solutions.

6.3 Selection of Generator Values

A general method of construction of the generator of bipolar utility function given by (22) and (23) can be summarized as follows:

```
$g(0)=0$; $g(m) = G$;
for $k = 1$ to $m - 1$:
    Determine $u \in (g(k-1), G)$;
    $g(k) = u$;
end
```

The function *Determine* depends on the specific method, for example: random selection, selection based on optimization of some criteria, etc.

7 Correlation Measures on the Set of Bipolar Profiles

Pearson's product-moment correlation coefficient

$$corr(x, y) = \frac{\sum_{i=1}^{n}(x_i - x)(y_i - y)}{\sqrt{\sum_{i=1}^{n}(x_i - x)^2}\sqrt{\sum_{i=1}^{n}(y_i - y)^2}} \tag{26}$$

is often used in recommender systems for measuring similarity between profiles [29]. In this section, we show that this correlation coefficient can be misleading if the opinions are measured in bipolar scales. Consider the following utility profiles with the ratings of 10 items in 7-point bipolar scale $J = \{1, 2, 3, 4, 5, 6, 7\}$ with the standard utility function $U(j) = j$ for all j in J:

$x = (7, 5, 5, 7, 7, 7, 5, 7, 5, 5)$,

$y = (5, 7, 7, 5, 5, 5, 7, 5, 7, 7)$,

$z = (3, 1, 1, 3, 3, 3, 1, 3, 1, 1)$.

The profiles x and y have only "positive" (greater than neutral $C = 4$) ratings, so a reasonable association measure A should show positive association between them: $A(x,y) > 0$; however, the correlation coefficient gives $corr(x,y) = -1$. The profiles x and z have almost opposite ("positive" vs. "negative") ratings, so a reasonable association measure should give negative association between them: $A(x,z) < 0$; however, the correlation coefficient gives $corr(x,z) = 1$.

Therefore, we need to introduce correlation (association) measures that, similarly to the correlation coefficient, could show positive and negative associations between profiles of ratings in bipolar scales, but without the drawbacks of Pearson's correlation coefficient, such as shown in the above example. Below we present such measures, based on general results discussed in [5–7].

Let I be a bipolar scale ($I = J$ or $I = K$) with the negation N and with the center C. We call the vector $x = (x_1, \ldots, x_M)$, $x_s \in I$, $s = 1, \ldots, M$, of elements from the bipolar scale I a *rating profile*. We also call the vector $C_X = (C, \ldots, C)$ of the length M the *central profile* of the set X of all rating profiles of the length M. We define the *negation of the profile* x as $N_X(x) = (N(x_1),\ldots, N(x_M))$. Obviously, N_X is an involution on X, i.e., $N_X(N_X(x)) = x$ for all profiles from X, and C_X is a unique fixed point of N: $N_X(C_X) = C_X$. For a bipolar utility function U defined on the bipolar scale I, we call the vector $U_X(x) = (U(x_1),\ldots, U(x_M))$ a *utility profile* of the rating profile x.

Suppose a user evaluates 6 items in the bipolar rating scale $J = \{1, 2, 3, 4, 5\}$ by the vector of ratings $x = (3, 5, 2, 4, 1, 3)$. Suppose $U(J) = \{-10, -3, 0, 3, 10\}$ is the centered bipolar utility function defined on J. Then, the utility profile of the rating profile x is given by $U(x) = (0, 10, -3, 3, -10, 0)$, the negation of the profile x is given by $N_X(x) = (3, 1, 4, 2, 5, 3)$, and on the set X of all rating profiles of the length 6, the central profile is given by $C_X = (3, 3, 3, 3, 3, 3)$.

Consider two rating profiles x and y with the same length. We will define the correlation measure $A_U(x,y)$ as a function of utility profiles $U(x)$ and $U(y)$. In applications, when the users profiles have different lengths, the vectors x and y will contain only ratings of items presented in the profiles of both users.

Definition 2. Let X be the set of all profiles of the length M with ratings from the bipolar scale I with the negation N and the center C. Let U be a bipolar utility function defined on I. A *correlation* (*association*) *measure* on the set $V = X \setminus \{C_X\}$ is a function A_U: $V \times V \to [-1, 1]$ that satisfies for all $x, y \in V$ the following properties:

$$A_U(x, y) = A_U(y, x) \qquad \text{(symmetry)} \qquad (27)$$

$$A_U(x, x) = 1, \qquad \text{(reflexivity)} \qquad (28)$$

$$A_U(x, N(y)) = - A_U(x, y). \qquad \text{(inverse relationship)} \qquad (29)$$

We call a correlation measure A_U *C-separable* if it satisfies the following properties:

$$A_U(x, y) > 0 \text{ if for all } s = 1, \ldots, M \text{ it holds } x_s, y_s > C \text{ or } x_s, y_s < C, \qquad (30)$$

$$A_U(x, y) < 0 \text{ if for all } s = 1, \ldots, M \text{ it holds } y_s < C < x_s \text{ or } x_s < C < y_s. \qquad (31)$$

The properties (27)–(29) were used in [7] in the definition of the correlation (association) measures on the set X with involution N. These properties generalize the properties of Pearson's correlation coefficient applied to M-tuples when the negation of M-tuples of real values is defined by $N(x) = -x = (-x_1, \ldots, -x_M)$. Here we extend the definition of association measures given in [7] on the set of bipolar utility profiles. The properties (30) and (31) are introduced here to avoid the problems with the correlation coefficient defined on bipolar profiles discussed at the beginning of this section. See also C-separability property of association measures on [0,1] considered in [6].

From (28) and (29), we have

$$A_U(x, N(x)) = -1. \tag{32}$$

Definition 2 can be extended from the set V to the set of all profiles X replacing the property (28) by

$$A_U(x, x) = 1 \text{ if } x \neq C_X \tag{33}$$

In this case, for all x in X we have

$$A_U(x,C_X) = A_U(C_X,x) = 0. \tag{34}$$

Consider a method for construction of correlation (association) measures on the set of bipolar utility profiles based on the general methods discussed in [5].

Proposition 2. Let I be a bipolar scale ($I = J$ or $I = K$) with the center C, X be a set of profiles $x = (x_1,\ldots, x_M)$ of the length M, $x_s \in I$, $s = 1, \ldots, M$, with the central profile $C_X = (C, \ldots, C)$, and U be a bipolar utility function on I. Then the following function is a C-separable correlation measure on $X \setminus \{C_X\}$:

$$A_U(x, y) = \frac{1}{2^t}\sum_{s=1}^{M}[|F(x_s)+F(y_s)|^t - |F(x_s)-F(y_s)|^t], \tag{35}$$

where

$$F(x_s) = \frac{U(x_s)-U(C)}{\sqrt[t]{\sum_{s=1}^{M}|U(x_s)-U(C)|^t}}, \tag{36}$$

and $t \geq 1$.

One can easily check that the properties (27)–(31) are satisfied for the function (35).

If the bipolar scoring function U in (36) is centered, i.e., $U(C) = 0$, then (36) is simplified as follows:

$$F(x_s) = \frac{U(x_s)}{\sqrt[t]{\sum_{s=1}^{M}|U(x_s)|^t}}. \tag{37}$$

Proposition 2 implies the following corollary.

Corollary 1. In the conditions of Proposition 2, if $t = 2$ in (35), (36), then the following function is a C-separable correlation measure on $X \setminus \{C_X\}$:

$$A_U(x, y) = \frac{\sum_{s=1}^{M}(U(x_s) - U(C))(U(y_s) - U(C))}{\sqrt{\sum_{s=1}^{M}(U(x_s) - U(C))^2}\sqrt{\sum_{s=1}^{M}(U(y_s) - U(C))^2}}. \tag{38}$$

If the bipolar scoring function U in (38) is centered, then:

$$A_U(x, y) = \cos(U(x), U(y)) = \frac{\sum_{s=1}^{M} U(x_s)U(y_s)}{\sqrt{\sum_{s=1}^{M} U(x_s)^2}\sqrt{\sum_{s=1}^{M} U(y_s)^2}}, \tag{39}$$

where $U(x) = (U(x_1), ..., U(x_M))$, $U(y) = (U(y_1), ..., U(y_M))$.

Since the formulas (37) and (39) require fewer operations than (36) and (38) in calculation of the correlation value $A_U(x,y)$ between a large number of pairs (x, y) of profiles, it is recommended to replace a bipolar utility function U defined on the bipolar scale I by a centered bipolar utility function $U - U(C)$ and then calculate the correlation between the corresponding profiles by (35), (37) or by (39).

Let us calculate correlation $A_U(x, y)$ between the profiles of ratings $x = (7, 5, 5, 7, 7, 7, 5, 7, 5, 5)$, $y = (5, 7, 7, 5, 5, 5, 7, 5, 7, 7)$, $z = (3, 1, 1, 3, 3, 3, 1, 3, 1, 1)$ from the bipolar scale $J = \{1, 2, 3, 4, 5, 6, 7\}$ considered at the beginning of this section. For the standard utility function $U(J) = J$, we obtain $U(x) = x$, $U(y) = y$, $U(z) = z$. We can calculate correlation between three profiles by (38). However, as we have noted above, it is more efficient to replace the bipolar utility function U by the centered utility function $U_C(J) = U(J) - U(C) = J - 4 = K = \{-3, -2, -1, 0, 1, 2, 3\}$ and to use this centered utility function in (39). We obtain: $A_U(x, y) = 0.6$, $A_U(y, z) = -1$, $A_U(x, z) = -0.6$. These values correspond to our propositions $A(x, y) > 0$ and $A(x, z) < 0$ considered at the beginning of this section, and the new correlation measure (38) does not have the drawbacks of Pearson's correlation coefficient. Note that we have $z = N(y)$, and for this reason, according to the property (29) of the correlation measure, we obtain: $A_U(x, z) = A_U(x, N(y)) = -A_U(x, y) = -0.6$; according to the property (32), we obtain $A_U(y, z) = A_U(y, N(y)) = -1$.

The correlation measure (38) obtained here as a particular case of (35) generalizes the constrained correlation coefficient considered in [32] (see formula (5)) using in (38) the standard 7-point utility function $U = J$ with the center $C = 4$ and $U(C) = 4$. Consider (38) in the form (39), when the utility function U is replaced by its centered form $U - U(C)$. As one can see, the formula (39) is a nonlinear function sensitive to the presence of respectively high utility values for the same items in both profiles x and y.

Consider an example for 7-point scale: $K = \{-3, -2, -1, 0, 1, 2, 3\}$. Suppose one uses the standard utility function $U(K) = K$, and two users have the following profiles of rating of four items in the scale K: $x = \{1, 1, 1, 1\}$, $y = \{-1, -1, -1, -1\}$. Due to $U(K) = K$, the corresponding utility profiles will have the same values $U(x) = x$, $U(y) = y$. Since the profiles are opposite: $N(x) = y$, the correlation (39) between them has the value $A(x,y) = -1$. Suppose a new item has the rating 2 from both users. Then, we obtain the new profiles: $x^* = \{1, 1, 1, 1, 2\}$, $y^* = \{-1, -1, -1, -1, 2\}$ with the correlation value $A(x^*, y^*) = 0$. As one can see, addition of the same value 2 to both ratings drastically changes the correlation value from -1 to 0. This situation can be avoided if we use nonlinear bipolar utility function. Let $U(1)= 1$ and $U(2) = 1.5$. Then, we obtain the following utility profiles: $U(x^*)= \{1, 1, 1, 1, 1.5\}$, $U(y^*)= \{-1, -1, -1, -1, 1.5\}$ with correlation between them $A_U(x^*, y^*) = -0.28$. As one can see, the change of the correlation value from -1 to -0.28 is not as drastic for considered nonlinear bipolar utility function as for the standard linear utility function.

Consider another example of profiles with ratings from 7-point rating scale K. Suppose again that we use the standard utility function $U(K) = K$ and two users have equal rating profiles of five items: $x = \{1, 1, 1, 1, 2\}$, $y = \{1, 1, 1, 1, 2\}$. We have $A(x, y) = 1$. Suppose for the sixth item both users have the opposite ratings 3 and -3. For the standard utility function, the correlation between new utility profiles $x^* = \{1, 1, 1, 1, 2, 3\}$ and $y^* = \{1, 1, 1, 1, 2, -3\}$ is drastically changed, from the value $A(x, y) = 1$ to the value $A(x^*, y^*) = -0.059$. Let us change the standard utility function by $U(K) = \{-2, -1.5, -1, 0, 1, 1.5, 2\}$. For this nonlinear bipolar utility function, the correlation between utility profiles changes from $A(x, y) = 1$ to $A_U(x^*, y^*) = 0.22$, which is not as drastic as for the standard utility function.

In the two considered examples, we used the method of construction of bipolar utility functions considered in Section 6.3, which defines the positive part of the centered utility function and symmetrically maps it to the negative part of the scale with 0 in the center of the scale. Similarly, we defined sequentially $U(1) = 1$, $U(2) = 1.5$, and $U(3) = 2$. Another method can be based on a parametric generator such as (24) or (25). For example, using the generator (24) with the parameter $p = 1$, we can obtain the similar results: for nonlinear bipolar utility function $U(K) = \{-2, -1.6, -1, 0, 1, 1.6, 2\}$, the correlation between utility profiles x^* and y^* has the value $A_U(x^*, y^*) = 0.24$.

In both examples, we decreased the absolute utility values of the categories near the poles to avoid the drastic change of the correlation value when both users use near polar ratings for the same items. In general, a parametric utility function can be heuristically adjusted to obtain intuitively validated solutions or optimized by some machine-learning method for the recommender or decision-making system using bipolar rating scales to obtain solutions with better performance.

8 Discussion and Conclusions

The rating scales applied in different application areas usually have bipolar structure, but this bipolarity between opposite categories of the scales often was not explicitly or formally exploited. Our paper introduces explicitly the property of bipolarity in the definition of the general structure of verbal bipolar scales, in the formal definition of the bipolar scale as the linearly ordered set of indexes with negation operation and in the definition of bipolar scoring function on bipolar scale preserving the symmetry of this scale. In the description of the general structure of bipolar scales in Section 3 we based on the paper [3]. The idea of the formal definition of bipolar scale on the set of indexes was partially based on the paper [16] where the linguistic categories of the scale are presented by fuzzy sets. In Section 4 we consider together two mutually related sets of indexes where $J = \{1,\dots, 2m+1\}$ is more traditional and $K= \{-m, \dots, m\}$ is more "natural" for representation of bipolar scales with opposite categories. Most of bipolar scales observed in Sections 1 and 2 can be represented as bipolar scales with general structure considered in Section 3 or formally as bipolar scales considered in Section 4. The concept of bipolar scoring or utility function defined on bipolar scale to the best of our knowledge is new. The bipolar scoring functions include the traditional scoring of n-point rating scales by numbers $1,\dots, n$ as particular case and such scoring functions are called standard bipolar scoring functions. But it seems more interesting instead of the standard scoring functions or instead of the linear utility functions to consider nonlinear utility functions. These utility functions can be given as parametric functions and adjusted by some machine learning procedure to obtain good or optimal results on the output of recommender or decision making system using these bipolar scales. The nonlinear bipolar scoring functions can be useful in modeling the ratings of users or in modeling utility or importance of categories in bipolar scales. The reasons to use nonlinear bipolar utility functions in correlation measure introduced in the paper are discussed in Section 7. The results on association measures considered in Section 7 are based on the papers [5-7]. Here, we use the terms association measure and correlation measure as interchangeable. We extend the property of C-separability from association measure on [0,1] considered in [6] on the set of utility profiles. The general formula for association measure on bipolar utility profiles is based on Minkowski distance and on general results considered in [5] for time series. The formulas (37) and (39) for centered bipolar utility functions are specific for C-separable association measures on bipolar utility profiles. The formula (38) generalizes the constrained correlation coefficient considered in [32] without utility functions. In our future work we plan to apply the results of the paper in collaborative filtering and in analysis of human ratings.

Acknowledgements

This work was supported in parts by the projects 20162204 and 20171344 of SIP IPN, 240844 and 283778 of CONACYT, 15-01-06456A of RFBR and by the

Russian Government Program of Competitive Growth of Kazan Federal University.

References

[1] Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering, 17(6), pp. 734-749.

[2] Atanassov, K. T. (1986). Intuitionistic fuzzy sets. Fuzzy sets and Systems, 20 (1), 87-96.

[3] Batyrshin, I.Z. (1990). On the structure of verbal scales. In: Proceedings of the Second All-Union Conference on Artificial intelligence. Minsk, 1990, vol. 1, pp. 37-40.

[4] Batyrshin, I. (2011). Uncertainties with memory in construction of strict monotonic t-norms and t-conorms for finite ordinal scales: basic definitions and applications. Applied and Computational Mathematics, vol. 10, 3, 2011, pp. 498-513.

[5] Batyrshin, I. (2013). Constructing time series shape association measures: Minkowski distance and data standardization. In 2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence (BRICS-CCI & CBIC), (pp. 204-212). IEEE. https://arxiv.org/abs/1311.1958v3.

[6] Batyrshin, I. Z. (2015). Association measures on [0, 1]. Journal of Intelligent & Fuzzy Systems, 29(3), pp. 1011-1020.

[7] Batyrshin, I. Z. (2015). On definition and construction of association measures. Journal of Intelligent & Fuzzy Systems, 29(6), 2015, pp. 2319-2326.

[8] Breese, J. S., Heckerman, D., & Kadie, C. (1998, July). Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence (pp. 43-52). Morgan Kaufmann Publishers Inc.

[9] Cambria, E., Olsher, D., & Rajagopal, D. (2014). SenticNet 3: a common and common-sense knowledge base for cognition-driven sentiment analysis. In Proceedings of the twenty-eighth AAAI conference on artificial intelligence (pp. 1515-1521). AAAI Press.

[10] Chang, L. (1994). A psychometric evaluation of 4-point and 6-point Likert-type scales in relation to reliability and validity. Applied Psychological Measurement, 18(3), 205-215.

[11] Dubois, D., & Prade, H. (2006). Bipolar representations in reasoning, knowledge extraction and decision processes. In International Conference

on Rough Sets and Current Trends in Computing, pp. 15-26. Springer Berlin Heidelberg.

[12] Friborg, O., Martinussen, M., & Rosenvinge, J. H. (2006). Likert-based vs. semantic differential-based scorings of positive psychological constructs: A psychometric comparison of two versions of a scale measuring resilience. Personality and Individual Differences, 40(5), 873-884.

[13] Goldberg, K., Roeder, T., Gupta, D., & Perkins, C. (2001). Eigentaste: A constant time collaborative filtering algorithm. information retrieval, 4(2), 133-151.

[14] Grabisch, M. (2006). Aggregation on bipolar scales. In Theory and applications of relational structures as knowledge instruments II (pp. 355-371). Springer Berlin Heidelberg.

[15] Gunderman, R. B., & Chan, S. (2013). The 13-point Likert scale: a breakthrough in educational assessment. Academic radiology, 20(11), 2013, pp. 1466-1467.

[16] Herrera, F., & Herrera-Viedma, E. (2000). Linguistic decision analysis: steps for solving decision problems under linguistic information. Fuzzy Sets and systems, 115(1), 67-82.

[17] Hjermstad, M. J., Fayers, P. M., Haugen, et al (2011). Studies comparing numerical rating scales, verbal rating scales, and visual analogue scales for assessment of pain intensity in adults: a systematic literature review. Journal of Pain and Symptom Management, 41(6), 1073-1093.

[18] Jang, J. S. R., Sun, C. T., & Mizutani, E. (1997). Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prentice Hall.

[19] Juniper, E. F., Guyatt, G. H., Ferrie, P. J., & King, D. R. (1999). Development and validation of a questionnaire to measure asthma control. European Respiratory Journal, 14(4), 902-907.

[20] Likert, R. (1932). A technique for the measurement of attitudes. Archives of Psychology. Vol 22, No. 140, 1932, pp. 55. New York.

[21] Liu, J., & Seneff, S. (2009). Review sentiment scoring via a parse-and-paraphrase paradigm. In Proceed. 2009 Conference on Empirical Methods in Natural Language Processing: Vol. 1 (pp. 161-169). Association for Computational Linguistics.

[22] Mosier, C.I. (1941). A psychometric study of meaning. The Journal of Social Psychology, 13(1), 123-140.

[23] Osgood, C.E. (1952). The nature and measurement of meaning. Psychological bulletin, 49(3), 197-237.

[24] Petrenko V.F. (1988). Psychosemantics of consciousness. Moscow, Ripol Klassik. (In Russian: Петренко, В. Ф. (1988). Психосемантика сознания. Рипол Классик).

[25] Pfanzagl, J. (1971). Theory of measurement. Physica. Physica-Verlag Heidelberg.

[26] Poria S, Gelbukh A, Cambria E, Hussain A & Huang G (2014). EmoSenticSpace: A novel framework for affective commonsense reasoning, Knowledge-Based Systems, 69, pp. 108-123.

[27] Pospelov D.A. (1989). Models of Reasoning. Essay in the Analysis of Mental Acts. Radio y Svyaz. Moscow. (In Russian: Поспелов Д.А. Моделирование рассуждений. Опыт анализа мыслительных актов , М.: Радио и связь, 1989, -184с.)

[28] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J. (1994). GroupLens: an open architecture for collaborative filtering of netnews. In Proceedings of the 1994 ACM conference on Computer supported cooperative work, pp. 175-186. ACM.

[29] Ricci, F., Rokach, L., Shapira, B., Kantor P.B. (2011) Recommender Systems Handbook, Springer US.

[30] Roberts, F. S. (1985). Measurement theory. Cambridge University Press.

[31] Schafer, J. H. J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. The adaptive web, 291-324.

[32] Shardanand, U., & Maes, P. (1995). Social information filtering: algorithms for automating “word of mouth”. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 210-217). ACM Press/Addison-Wesley Publishing Co.

[33] Sugeno M. (1974) Thery of Fuzzy Integrals and its Applications. Dissertation. Tokio Institute of Technology.

[34] Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. Computational linguistics, 37(2), 267-307.

[35] Tarassov V.B. (2001). Analysis and modeling of NOT-factors on polar scales. In: Integrated Models and Soft Computing in Artificial Intelligence, Moscow, Fismatlit, pp. 65-71. (In Russian: Тарасов В.Б. Анализ и моделирование НЕ-факторов на полярных шкалах// Интегрированные модели и мягкие вычисления в искусственном интеллекте. − М.: Наука. Физматлит, 2001. − С.65-71).

[36] Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., & Kappas, A. (2010). Sentiment strength detection in short informal text. Journal of the American Society for Information Science and Technology, 61(12), 2544-2558.

[37] Thurstone, L. L. (1928). Attitudes can be measured. American Journal of Sociology, 33(4), 529-554.

[38] Weijters, B., Cabooter, E., & Schillewaert, N. (2010). The effect of rating scale format on response styles: The number of response categories and response category labels. International Journal of Research in Marketing, 27(3), 236-247.

[39] Xu, Z. (2012). Linguistic decision making. Springer Berlin Heidelber.

[40] Zadeh, L. A. The concept of a linguistic variable and its application to approximate reasoning.—I. Information sciences, 8(3), 1975, pp. 199-249; —II. Information sciences, 8(4), 1975, pp. 301-357. —III. Information sciences, 9(1), 1975, pp. 43-80.

Authorship Attribution in Portuguese Using Character N-grams

Ilia Markov[1], Jorge Baptista[2], Obdulia Pichardo-Lagunas[3]

[1]CIC, Instituto Politécnico Nacional (IPN), Av. Juan de Dios Bátiz S/N, Del. Gustavo A. Madero, 07738, Mexico City, Mexico
imarkov@nlp.cic.ipn.mx

[2]Univ. Algarve/FCHS and INESC-ID Lisboa/L2F, Campus de Gambelas, P-8005-139, Faro, Portugal
jbaptis@ualg.pt

[3]UPIITA, Instituto Politécnico Nacional (IPN), Av. Instituto Politécnico Nacional 2580, Del. Gustavo A. Madero, 07340, Mexico City, Mexico
opichardola@ipn.mx

Abstract: For the Authorship Attribution (AA) task, character n-grams are considered among the best predictive features. In the English language, it has also been shown that some types of character n-grams perform better than others. This paper tackles the AA task in Portuguese by examining the performance of different types of character n-grams, and various combinations of them. The paper also experiments with different feature representations and machine-learning algorithms. Moreover, the paper demonstrates that the performance of the character n-gram approach can be improved by fine-tuning the feature set and by appropriately selecting the length and type of character n-grams. This relatively simple and language-independent approach to the AA task outperforms both a bag-of-words baseline and other approaches, using the same corpus.

Keywords: authorship attribution; character n-grams; Portuguese; stylometry; computational linguistics; machine learning

1 Introduction

The Authorship Attribution (AA) task aims at identifying the author of an anonymous target text given a predefined set of candidate authors and corresponding samples of their texts, deemed to be representative of their writing practices (style). In recent years, the AA task triggered an increasing interest due to its impact on marketing, security, and forensic linguistics, where it can help to limit the search space for the author of a text under investigation. From a machine-learning perspective, approaches to the AA task can be viewed as a multi-class, single-label classification problem, in which the set of class labels is known *a priori*. The challenge consists in modelling this classification task so that automatic methods can assign class labels (authors' names) to objects (text samples).

Character *n*-gram features have proved to be highly predictive not only for the AA task [13, 18, 35] but also for similar tasks, such as Author Profiling [20]. Character *n*-grams are language-independent features but they are able to capture lexical and syntactic information, as well as punctuation and capitalization information related with the authors' personal style [6, 14]. Sapkota *et al.* [29] showed that, for the AA task in English, some categories of character *n*-grams perform better than others [29, p. 94]: "These categories are related to the three linguistic aspects hypothesized to be represented by character *n*-grams: morphosyntax (as represented by affix-like *n*-grams), thematic content (as represented by word-like *n*-grams) and style (as represented by punctuation-based *n*-grams)." Hence, the authors report that character *n*-grams that capture affixes and punctuation information, which can be related to morpho-syntactic and stylistic information, perform better than using all character *n*-grams.

This paper focuses on the AA task in the Portuguese language. Work on AA for the Portuguese language is still scarce [13, 33, 36, 37], and several strategies have been put in practice with varying results. From the morphological point of view, Portuguese is a moderately rich language: besides a small set of productive prefixes and suffixes, most of these affixes can only be analysed by resourcing to the language's history; there is a relatively complex verbal inflection system, yielding around 70 inflected forms, several of them homographs; nominal and adjectival is limited to gender and number, within limited set of morphemes. Thus, the settings for the *n*-gram approach to the AA task as used for a language such as English is likely to yield different results in Portuguese. This makes it important to examine which types and/or combinations of character *n*-grams are the most predictive for the Portuguese language.

This paper shows that selecting optimal feature representation, a popular machine-learning algorithm, and combining different types of character *n*-grams, allows for improving previous results in AA task using the same Portuguese corpus. Moreover, appropriate tuning of the size of the feature data set can render significantly lighter the machine-learning processing with only slight variation of accuracy.

(i) Which types and length of character *n*-grams are optimal for the AA task in Portuguese?

(ii) Is it possible to enhance AA performance by selecting an appropriate parameters, that is, feature combinations, feature set size, etc. using only the training corpora?

(iii) Which feature representation and machine-learning algorithm provide the best results for this task?

(iv) Is the conclusion reported in [29], that the best performing model is based solely on affix and punctuation *n*-grams, valid for the Portuguese language?

2 Related Work

Over the last decade, Authorship Attribution has become an important field of study in computational linguistics, among other factors because of its high stake applications in Social Media Forensics [4,5]. The PAN competition[1] is a series of scientific events and shared tasks on digital text forensics, and it is one of the main *fora* regarding the AA and other related tasks. These tasks include Authorship Attribution proper, Authorship Verification (determine if two texts were written by the same author), Clustering (grouping documents by author) and Diarization (identify and group parts of a document written by the same author). Other tasks relate to Author Profiling (by gender, age or personality) and, more recently, Author Obfuscation.

A recent trend on AA and related tasks focused on cross-topic and cross-gender scenarios [35], which is a more realistic context to the development of practical applications of this task. In this section, a selection of the works on the Authorship Attribution task in both the English and Portuguese languages is briefly presented.

Many previous studies focused on finding stylometric features that represent the authors style [18, 26, 30]. [10] present an extensive list of the main features used in the AA task: word-based and punctuation-based features, either discarding or combining function words statistics, using stemming or lemmatization techniques, and term frequency-inverse document frequency (*tf-idf*). A popular approach involved using syntactic information extracted from texts: [37] based their approach on syntactic features such as subject, predicate, and accessories. In [10], Gómez-Adorno *et al.* showed that textual patterns obtained from shortest path walks over integrated syntactic graphs is a useful methodology for the AA task. Textual genre and length are key issues in AA. Most work on social media [2, 3, 33] have to tackle with the limited size of the texts.

Some linguistic-poor approaches are based on character *n*-grams. Many independent works have demonstrated that character *n*-grams are effective features for the AA task [8, 18, 34]. Character *n*-grams are predictive when used in isolation [8] or when combined with other stylometric features [25]. Several studies [22, 35] investigated the impact of varying threshold values in single- and cross-topic AA conditions. The studies conclude that high threshold values are optimal for cross-topic AA. Finally, Sapkota *et al.* [29] introduced the notion, that this paper explores, that different types of *n*-grams may have differential predictive value for the AA task, showing that, for English, using affix+punctuation *n*-gram categories are more predictive than using all *n*-grams.

Different machine-learning techniques also perform in a varying way depending on a large number of factors [12]. Several works explore different machine-learning approaches to the AA task. Support Vector Machines (SVM) is a very popular machine-learning method in the field. [37] based their approach on syntactic features using various fusion methods with SVM. [28] showed that character-level convolutional neural networks outperform state-of-the-art approaches on four out of five examined datasets. Homem and Carvalho [13] explored fuzzy methods to

[1] http://pan.webis.de

determine authorship fingerprints in texts. Posadas-Durán *et al.* [24] showed that doc2vec-based feature representation outperforms the state-of-the-art approaches on the examined corpora.

Related work on the Portuguese Language is still scarce (see [1,19] for an overview). Pavelec *et al.* [23] used discourse connectors (mainly conjunctions) in Brazilian journalistic text as features to model the AA task. Sousa-Silva [32] experimented with different types of stylistic markers (POS-based, punctuation, word length, suffixes, pronouns, and conjunctions) drawing on a corpus of European Portuguese journalistic texts and using SVM, showing that simple quantitative data (word and sentence length, and punctuation), rather than more linguistic rich features, perform remarkably well. Work by Homem and Carvalho [13] also used character *n*-grams ($n = 4$) with a corpus of European Portuguese journalistic texts. Results bellow the threshold of 60% were reported. This corpus has been made available for this paper, so this is the most closely related data available for comparison. In one of the more recent work, [36] examined whether qualitative and quantitative analysis using SVM is an efficient approach to forensic cases of AA. [9] worked on gender classification (Author Profiling task) based on Twitter data in Portuguese. Silva *et al.* [33] focused on idiosyncratic usage on a corpus of social media data using SVM.

3 Character N-gram Features

In this work, the same, language-independent, character *n*-gram categories introduced by Sapkota *et al.* [29] are used. The original definitions for some of the categories are refined in order to make them more complete. This paper also experiments with character 4-grams, considering the usually larger word and affix length in Portuguese. The categories of character *n*-grams can be organized into three main super categories (affix–, word–, and punctuation–related *n*-grams). They are defined in Table 1. As an example, let us consider the following sample sentence (1):

(1) *“Vejo-te na quarta-feira, está bem?”, respondeu o Pedro.*
(“[I]’ll see you on Wednesday, is [that] okay?”, replied Pedro.)

The character *n*-grams ($n = 3$ and 4) for the sample sentence (1) for each of the categories are shown in Tables 2 and 3, respectively. For clarity, spaces are represented by the underscore ‘_’.

Following the work by Sapkota *et al.* [29], three models of *n*-grams are examined:

1. **All-untyped**: when the categories of *n*-grams are ignored; any distinct *n*-gram is a different feature. This corresponds to the more common approach of extracting *n*-grams without classifying them into different categories.

2. **All-typed**: when *n*-grams of all available categories (**affix+word+punct**) are considered. Notice that instances of the same *n*-gram may refer to different features.

3. **Affix+Punct**: when the *n*-grams of the **word** category are excluded.

Table 1
Categories of character *n*-grams introduced by Sapkota *et al.* [29].

Affix character n-grams	
prefix	An *n*-gram that covers the first *n* characters of a word that is at least $n + 1$ characters long.
suffix	An *n*-gram that covers the last *n* characters of a word that is at least $n + 1$ characters long.
space-prefix	An *n*-gram that begins with a space and that does not contain any punctuation mark.
space-suffix	An *n*-gram that ends with a space, that does not contain any punctuation mark, and whose first character is not a space.
Word character n-grams	
whole-word	An *n*-gram that encompasses all the characters of a word, and that is exactly *n* characters long.
mid-word	An *n*-gram that contains *n* characters of a word that is at least $n + 2$ characters long, and that does not include neither the first nor the last character of the word.
multi-word	An *n*-gram that spans multiple words, identified by the presence of a space in the middle of the *n*-gram.
Punctuation character n-grams (abbreviated as **punct**)	
beg-punct	An *n*-gram whose first character is a punctuation mark, but the middle characters are not.
mid-punct	An *n*-gram whose middle character is a punctuation mark (for n =3).
end-punct	An *n*-gram whose last character is punctuation mark, but the first and the middle characters are not.

One of the main conclusions of Sapkota *et al.* [29] was that models based on **affix+punct** features were more efficient than models trained using all the features. In the current paper, these three models were applied in order to examine whether this conclusion is also valid for the Portuguese language.

Moreover, the performance of each category of character *n*-grams is examined separately, and the different models mentioned above are combined, aiming at identifying the most predictive stylometric feature combination for Portuguese.

4 Experimental Settings

In this section, the experimental settings are laid out. First the corpus here used is briefly described, in order to present the criteria adopted in building the two subsets used in the experiments, a *balanced* and an *unbalanced* subcorpus. Next, the method for defining a baseline is presented, using standard evaluation procedures and machine-learning algorithm (SVM), commonly used in this task.

Table 2
Character *n*-grams (*n* = 3) per category for the sample sentence (1), where SC stands for Super Category.

SC	Category	3-grams
affix	*prefix*	Vej qua fei est res Ped
	suffix	ejo rta ira stá deu dro
	space-prefix	_na _qu _es _be _re _o_ _Pe
	space-suffix	te_ na_ tá_ eu_
word	*whole-word*	bem
	mid-word	uar art eir esp spo pon ond nde edr
	multi-word	e_n a_q á_b u_o o_P
punct	*beg-punct*	“Ve -te -fe ,_e ,_r
	mid-punct *	_“_ _-_ _,_ _?_ _”_ _._
	end-punct	jo- ta- ra, em? ro.

* In this work, punctuation marks are separated from adjacent words and from each other by space for this category. This enables to capture their frequency [22].

Table 3
Character *n*-grams (*n* = 4) per category for the sample sentence (1), where SC stands for Super Category.

SC	Category	4-grams
affix	*prefix*	quar feir resp Pedr
	suffix	arta eira ndeu edro
	space-prefix	_na_ _qua _est _bem _res _Ped
	space-suffix	stá_ deu_
word	*whole-word*	Vejo está
	mid-word	uart espo spon pond onde
	multi-word *	te_n na_q tá_b eu_o o_P
punct	*beg-punct*	“Vej -te_ -fei ,_es ,_re
	mid-punct **	_“_ _-_ _,_ _?_ _”_ _._
	end-punct	ejo- rta- ira, bem? dro.

* In the case when the previous word is more than one character long, two characters are considered; otherwise, only one character is considered.
** This is the same as for character 3-grams (see Table 2).

4.1 Corpus

Experiments were conducted using a data set extracted from a large corpus composed of 5,167 newspaper articles (1,489,947 words) in Portuguese, which were retrieved from the on-line edition of the *Público* newspaper. This is the same set of texts that was used by Homem and Carvalho in their paper [13]. For this paper, though, the titles were included in the texts.

The texts in this corpus were written by 87 different authors on 16 distinct topics. The corpus here used is, therefore, a mixed-topic corpus. This corresponds to a realistic scenario, where the texts by candidate authors can be written either on the same topic or on different topics. The topics' classification is derived from the newspaper sections from hence they were retrieved. They include texts from *national* (26%) to *world news* (16%), including a *local news* section (8%); topic-specific sections relate to *sport* (20%), *economy* (11%) and *culture* (1%); thematically mixed sections include: news *highlights* (6%) and the *last page* (3%) section, an *opinion* section, with texts from non-regular contributors' (3%), as well as a *chronicles* section, with texts from regular contributors' (1%); four of the 16 topics have less than 10 texts each. In spite of this topic distribution across the corpus, it should be noticed that, for the purpose of the AA task, some of the thematically mixed sections may also be considered of interest, as long as the number of texts per author is deem sufficient.

The corpus is highly unbalanced in terms of articles/author ratio, with an average of 59.5 texts per author, but only 34 authors having 60 or more texts, one author (chronicle) reaching 169 texts, while the least represented author has only 30 articles. The topic/author ratio is also quite unbalanced: for an average of 3.8 topics per author, only two authors address the maximum of 7 topics, while 9 only write about a single topic (and not always the same). Another issue is that there is a very short number of authors with a relevant number of texts in at least two different topics and, when this is the case, those topics do not always overlap. This situation raises considerable difficulties to cross-topic AA experiments with this corpus. For this paper, cross-topic AA is not addressed.

In the first phase of experiments, a balanced subset of the corpus was used, selecting only those authors who have at least 50 articles each. There were 50 authors with more than 50 articles per author. The corpus was then divided into two subsets and, in order to insure the reproducibility of the results, this splitting was based on the articles' ID, selecting the first 25 for training and the remaining 25 for evaluation. This corpus will be referred to as *balanced*.

In the second phase, the whole corpus was used. It was divided in a similar way as before, that is, the first half of articles per author were used for training and the second half for testing, using the texts' ID sequence. In case of an odd number of articles, the remaining text was added to the evaluation set. These settings were adopted in order to be able to compare this paper's results with those reported in [13], who proceed in the same way. This corpus will be referred to as *unbalanced*.

4.2 Defining a Baseline

The first set of experiments was carried out using the balance corpus. In order to better frame the results obtained from the different methods here applied to the AA task, the bag-of-words (*BoW*) approach was defined as the baseline, which is a common procedure for this task, due to its language-independence and the fact that, in spite of its being a relatively simple and computationally inexpensive method, it already yields a strong (and challenging) baseline. In the BoW approach here used, features fed into the machine-learning algorithm are based on word frequency (punctuation marks are ignored).

In view of the size of the dataset, which comprises BoW 45,707 features, different frequency thresholds (*frq*) values were also experimented, in order to assess the impact of varying the size of the feature set in the performance of the machine-learning algorithms. In this way, frequency thresholds were tested using features with minimum *frq* $\geqslant$ 25, 50, 75, and 100, that is, first using the whole set of features and then all the features with at least 25, 50, 75, and 100 occurrences in the training corpus, thus progressively reducing the size of the feature set. According to [22] and [35], selecting an appropriate size of the feature set is important in cross-topic AA. Different threshold values were experimented in order to examine their impact under mixed-topic conditions in Portuguese.

Following Homem and Carvalho [13], who used the same corpus as this paper does, a 50% training/testing corpus partition was adopted for the evaluation; only the training subcorpus is used in order to find the best parameters for the task at hand. This methodology also follows the practice of PAN international competitions on AA task, where the testing corpus is not made available to the competitors. Besides, experiments were also carried out using an adaptation of the 10-fold cross evaluation method, where the training subcorpus is divided into 10 folds, only 9 are used to train the model and 1 is left out; the model is then evaluated on the testing subcorpus, and the process is repeated leaving another fold out; finally, the results of each training-testing stage are averaged.

Finally, two data representation methods have been compared, namely, the term frequency (*tf*) described above; and a *binary* representation, which indicates just the presence or absence (1 or 0) of a feature in a given document. The *tf-idf* and normalized feature representation methods have also been experimented but they were dismissed as they did not show any positive effect on results. Table 4 shows the results from these experiments with a bag-of-word approach, varying the minimum frequency threshold, the feature representation, and the evaluation method.

Table 4
Baseline results in terms of accuracy (%) using the *bag-of-words* (BoW) approach, with different frequency threshold values, different evaluation procedures, *i.e.* 10-fold cross-validation (*10-fold*) and 50% training/testing evaluation (*50%-test*), and different data representation methods, namely *term frequency* (*tf*) vs. *binary*, using SVM algorithm. The top accuracy values in each experimental setting are shown in bold typeface.

min. feature frequency	**tf**		**binary**		**N of features**
	10-fold	**50%-test**	**10-fold**	**50%-test**	
0 (all features)	59.52	51.36	**70.40**	61.84	45,707
25	**65.52**	**57.04**	68.96	**62.80**	2,933
50	64.08	55.84	67.36	60.16	1,554
75	62.24	55.04	64.00	58.16	1,032
100	59.04	50.80	60.24	52.48	749

As one can see from Table 4, *binary* feature representation systematically outperforms *term frequency* (*tf*) data representation scheme, regardless of the examined threshold and of the evaluation procedures (10-fold cross-validation and 50% training/test partition of the corpus). As expected, the 10-fold cross-validation setting also yields better results than the 50% training/test partition of the corpus.

Also as expected, the different frequency thresholds (*frq*) have a significant impact on the size of the feature set: for *frq* = 25, the 2,933 features corresponds to 6.42% of the entire data set; the number of features (1,554) is almost halved for *frq* = 50;

the reduction in the feature data set (1,032) is less pronounced (66%) in the next threshold (*frq* = 75); and even less important (73%) for *frq* = 100.

Considering the size of the feature set for each frequency threshold, and comparing it with the accuracy obtained for each data representation method (*tf*/*binary*) and for the two evaluation scenarios (10-fold/50%-test); it should be noticed that: (i) In the term frequency (*tf*) settings, the best data size/accuracy combination is achieved with *frq* = 25; an important drop is observed when the entire data set is used instead (6.00% and 5.68%, in the 10-fold and the 50%-test scenarios, respectively); moving from *frq* = 25 to *frq* = 50 or from here to *frq* = 75 has only a minor effect in the performance of the classifiers, resulting in a reduction of accuracy slightly larger in the 10-fold evaluation setting (1.44% and 1.84%) than in the 50%-test setting (1.20% and 0.80%); another important drop (3.20% and 4.24%) occurs when selecting a *frq* = 100. This results can be interpreted in the sense that, when using the term frequency data representation method, low frequency words deteriorate the performance of the classifier, while important information is discarded if only highly frequent words are kept. (ii) In the binary feature representation setting, in 10-fold evaluation scenario the performance progressively decays (a drop of 1.44% from the best-performing all-features scenario to the *frq* = 25, and then, progressively, 1.60% to *frq* = 50, 3.36% to *frq* = 75 till 3.76% for *frq* = 75); on the other hand, in the 50%-test evaluation setting, the all-features scenario actually yields slightly worst results than *frq* = 25; still, in both scenarios, the difference between the two best-performing thresholds is small (1.44% in the 10-fold setting, against 0.96% in the 50%-test).

The best result (70.40% accuracy) was achieved using binary feature representation in a 10-fold cross-validation setting and taking all features into consideration. This could be interpreted as evidence that even low frequency words, usually associated with topic-specific information, provide useful information to the classifier. However, low frequency lexical features may be considered as too topic-specific, which may lead to unintended extraction of topic or domain information, instead of capturing the characteristics of the authors' style.

5 Different Character N-gram Approaches

Next, the *n*-gram methods were applied to the AA task using the same settings as described above, first in a 3-grams scenario and then in a 4-grams scenario. For each scenario, tree different models were built: (i) using only untyped *n*-grams; (ii) using only typed *n*-grams; and (iii) combining affix and punctuation *n*-grams, following the proposal of Sapkota *et al.* [29]. The 3-gram approach replicates previous experiments in the literature [20,22,29], while 4-grams were introduced to investigate wether it would be better suited for Portuguese, as it presents a moderately rich morphology. For each model, the minimum frequency threshold variation (from *frq* = 0 to 100 with step 25) was tested.

5.1 Character 3-gram and 4-gram Models

Tables 5 and 6 present the results from the 3-gram and 4-gram scenarios, using the three different models presented above, and, for each model, the same frequency threshold values were tested. Again, the same feature representation and evaluation methods were examined. These tables also show, for each setting, the size of the feature set (number of features).

Table 5
Accuracy of character 3-gram models (untyped, typed, and affix+punctuation), across different frequency thresholds, using two feature representation methods and two evaluation procedures (this Table's headings are the same as those of Table 4). The size of the feature set in each specific setting is also provided. The best performing model for each type of feature set is highlighted in bold typeface. In case two models yield the same result, the one with the smaller number of features is selected.

Model	**min. feature frequency**	**tf**		**binary**		**N of features**
		10-fold	**50%-test**	**10-fold**	**50%-test**	
untyped	0 (all features)	61.20	55.60	69.68	63.20	24,400
	25	64.27	58.00	69.52	**63.36**	6,330
	50	64.64	59.20	**69.76**	62.32	4,659
	75	**65.28**	**59.28**	69.60	62.56	3,938
	100	64.64	58.96	69.12	62.64	3,443
typed	0	61.92	55.44	70.64	64.16	27,686
	25	64.80	59.28	70.32	64.32	7,283
	50	64.88	59.60	**70.72**	63.36	5,413
	75	65.36	**60.00**	70.32	**63.60**	4,559
	100	**65.36**	59.84	69.92	63.20	3,965
affix+ punct	0	60.80	54.40	**69.28**	**60.96**	16,275
	25	64.56	56.72	69.04	60.48	3,798
	50	64.16	**57.52**	68.56	60.16	2,730
	75	**65.36**	56.32	67.76	59.84	2,255
	100	64.40	57.04	68.40	59.36	1,942

As in the case of the BoW approach, binary feature representation always outperforms term frequency. The remaining of this paper will then focus on the results from binary feature representation, though the corresponding results for term frequency are also presented. Typed character 3-grams are slightly more predictive than either untyped or affix+punctuation, the later being showing the worst performance (though the difference is only marginal). The best 10-fold cross-validation result was obtained with the threshold *frq* = 50. However, accuracy variation across thresholds in each model and even between different models is minimal.

Similarly to a 3-gram scenario, in a 4-gram scenario (Table 6) binary feature representation systematically provide higher results than term frequency. Also, the typed 4-gram model is slightly more predictive that both the untyped and the affix+punctation models. Again, the affix+punctation model is the worst performing.

The conclusion by Sapkota *et al.* [29] that using only affix+punctuation *n*-grams is more predictive than using all *n*-grams does not seem to be valid for the Portuguese language. This may indicate that *word* character *n*-gram category, which is

Table 6
Accuracy of character 4-gram models (untyped, typed, and affix+punctuation), across different frequency thresholds, using two feature representation methods and two evaluation procedures (this Table's headings are the same as those of Table 4). The size of the feature set in each specific setting is also provided. The best performing model for each type of feature set is highlighted in bold typeface. In case two models yield the same result, the one with the smaller number of features is selected.

Model	**min. feature frequency**	**tf**		**binary**		**N of features**
		10-fold	**50%-test**	**10-fold**	**50%-test**	
untyped	0 (all features)	60.96	54.64	69.28	**66.16**	92,646
	25	67.04	60.88	70.48	65.92	15,547
	50	**67.12**	**61.76**	70.24	65.60	10,535
	75	66.96	61.44	70.48	65.44	8,125
	100	66.88	60.56	**70.48**	64.72	6,717
typed	0	61.92	54.48	70.08	**66.56**	75,969
	25	66.72	60.08	**70.64**	66.40	13,741
	50	**67.12**	**61.52**	70.40	65.92	9,251
	75	66.40	61.36	70.24	65.44	7,100
	100	66.64	60.48	69.60	64.72	5,803
affix+ punct	0	61.92	53.92	69.36	64.56	42,463
	25	**67.20**	59.12	69.36	64.72	6,823
	50	66.32	59.28	**69.76**	64.88	4,559
	75	64.96	**59.84**	69.04	**64.96**	3,528
	100	65.68	59.28	68.08	62.96	2,89

considered to be more closely related to thematic content, should not be discarded when dealing with Portuguese. However, as the differences between models are only marginal, more experiments are required to verify this conclusion.

This is somehow strengthen by the fact that, in the BoW approach, low frequency words, which are related to topic-specific information, still contribute to AA accuracy in mixed-topic settings. However, as mentioned above, the results may be biased, since the approach may be capturing the topic information and not the style of the author. In the case of 4-grams, any threshold above 0 improves the results. According to [22, 35], higher frequency threshold values provide better results in cross-topic AA. In the mixed-topic corpus used in this work, varying the frequency threshold does not seem to significantly improve the results; however, it allows for an important reduction of the size of the feature set without loss of accuracy.

Comparing the performance of 3- and 4-gram models, the difference is minimal using the 10-fold cross-validation evaluation method, while when using the 50%-test settings the results of the 4-gram models are approximately 3%–4% higher.

In view of these differences, the untyped 5-gram model was also examined in order to be able to establish the optimal length of character n-grams. Results are shown in Table 7. They are only slightly lower than those obtained when using 4-grams. In fact, the typed n-gram approach proposed by Sapkota *et al.* [29] is maximally efficient for n-gram models with the maximum length of 4, since when using typed 5-grams, many character n-grams are not captured by the proposed categories. Moving from a 4-gram to a 5-gram model does not seem to have much impact on the

Table 7
Accuracy of untyped character 5-gram models, across different frequency thresholds, using two feature representation methods and two evaluation procedures (this Table's headings are the same as those of Table 4). The size of the feature set in each specific setting is also provided. The best performing model is highlighted in bold typeface. In case two models yield the same result, the one with the smaller number of features is selected.

min. feature	**tf**		**binary**		**N of**
frequency	**10-fold**	**50%-test**	**10-fold**	**50%-test**	**features**
0 (all features)	57.76	50.64	68.80	64.00	242,932
25	67.12	58.16	**70.16**	64.40	24,553
50	**67.12**	**59.36**	69.68	**64.40**	14,448
75	65.76	58.24	69.04	64.16	10,173
100	65.28	58.40	68.48	63.60	7,776

Portuguese corpus, probably because average word length is larger than in English. Still, the size of the feature set significantly increases, which makes this a suboptimal approach. As the differences in the overall performance of 4- and 5-gram models are only marginal, more experiments may be necessary.

Comparing the results above (Tables 5–7) with the bag-of-words baseline approach (Table 4), one can see that most *n*-gram models outperform the BoW approach, even if the differences are small. Focusing only on the binary feature representation, the only cases when the best *n*-gram models were unable to yield better results than the best baseline models were: (i) the untyped 3-grams, 10-fold; (ii) the affix+punctuation 3-grams, 10-fold, 50%-test; (iii) the affix+punctuation 4-grams, 10-fold; (iv) the untyped 5-grams, 10-fold. This confirms that the BoW approach is already quite a challenging benchmark for the AA task. In the next sections, different strategies will be put in place to improve the results reported so far.

5.2 Exploring Typed and Untyped Character N-grams

Based on the previous experiments, and focusing only on the binary feature representation and on the 10-fold cross-validation evaluation method, the threshold $frq \geqslant 50$ was selected for the next experiments, since 3 out of the 6 best models were obtained using this threshold. By choosing this threshold an average reduction of 88% of the entire feature set is achieved.

Other experiments were carried out by cutting out the most frequently occurring words in the training corpus, namely by discarding the 50 most frequent words, and then by successively cutting 2%, 5%, and 10% of the most frequent words. This strategy has proved to be helpful in related tasks, such as Author Profiling [20]. However, in the AA task, the most frequent words, which are stop-words for the most part, are considered of a great importance [14]. This conclusion is also valid for Portuguese, since discarding the most frequent *n*-gram features did not led to improvements in accuracy (for lack of space, results are not provided here).

Next, the contribution of each category of character *n*-grams is examined separately (Table 8). To do so, each category was discarded one by one and the performance of typed 3- and 4-grams was evaluated. If the result is improved (italics in Table 8), the examined category is not predictive; otherwise (bold typeface Table 8), it is a predictive category.

Table 8
Results in terms of accuracy (%) per category using typed character 3- and 4-grams, threshold ⩾ 50, binary representation, and SVM algorithm. Three best predictive 3- and 4-gram categories are in bold typeface; three worst predictive 4-gram categories are in italics.

Feature set	3-grams		N of	4-grams		N of
	10-fold	**50%-test**	**features**	**10-fold**	**50%-test**	**features**
All categories	70.72	63.36	5,413	70.40	65.92	9,251
All – prefix	**68.80**	**62.64**	4,419	**70.24**	**65.52**	8,082
All – suffix	**69.76**	63.52	4,768	*70.80*	*66.24*	8,205
All – space-prefix	70.72	63.52	4,941	**69.92**	**64.96**	8,113
All – space-suffix	70.08	63.52	5,186	70.32	*66.32*	8,629
All – whole-word	70.24	63.28	5,303	70.48	*66.00*	9,088
All – mid-word	**69.76**	**62.80**	3,447	70.40	**65.04**	6,265
All – multi-word	70.48	**63.12**	4,806	*70.80*	65.92	7,708
All – beg-punct	69.84	63.36	5,280	**70.00**	65.60	9,021
All – mid-punct	70.32	63.12	5,400	70.32	65.84	9.238
All – end-punct	70.56	63.76	5,167	*70,96*	65.76	8,910

After establishing the best and worst performing 3- and 4-gram categories, a process of feature selection was undertaken. First, the most predictive 3-gram categories and their combinations were added to the model of All-typed character 4-grams. Results are presented in Table 9. The best model corresponds to combining All-typed character 4-grams with prefix 3-grams and middle-punctuation 3-grams categories. This best-performing model was then selected for the next step of feature selection. Next, the worst predictive 4-gram categories and their combinations were discarded from the best combination established in the previous experiment. The results are shown in Table 10. This strategy does not seem to improve the 10-fold cross-validation accuracy, which indicates that, in these settings, all 4-gram categories contribute to the overall accuracy. However, there is a slight improvement in the 50%-test settings when some 4-gram categories are discarded, namely and in decreasing order of accuracy: Best – multi-word (0.72%), Best – end-punctuation (0.24%), Best – suffix– end-punctuation and Best – multi-word – end-punctuation (both 0.16%). As results from this feature selection procedure did not improve accuracy, the strategy of combining different models was tested next.

Next, typed, untyped, 3- and 4-gram models were combined to find the most predictive stylometric feature combination. Typed and untyped *n*-grams are different features, since typed *n*-grams are tagged with the corresponding category. One of the reasons why the combination of typed and untyped *n*-grams can enhance the performance is that some typed *n*-grams, being divided into multiple categories, are discarded by the high threshold (*frq* ⩾ 50), while untyped *n*-grams are still able to exceed this threshold; *e.g.*, in the phrase *"com compaixão"* (with compassion) the untyped 3-gram `com` appears 2 times, but it corresponds to two distinct typed 3-grams: one *whole-word* 3-gram and another *prefix* 3-gram. Moreover, punctuation

Table 9
Results in terms of accuracy (%) combining typed character 4-grams with the three best predictive 3-gram categories and their combinations. Threshold ⩾ 50, binary representation, SVM algorithm.

Feature set	10-fold	50%-test	N of Features
Best: All typed character 4-grams	70.40	**65.92**	9,251
All + prefix 3-grams	70.40	65.44	10,245
All + suffix 3-grams	70.56	65.44	9,896
All + mid-word 3-grams	70.56	65.28	11,217
All + prefix + suffix 3-grams	69.76	65.20	10,890
All + prefix + mid-word 3-grams	**70.88**	64.88	12,211
All + suffix + mid-word 3-grams	70.16	65.52	11,862
All + prefix + suffix + mid-word 3-grams	70.64	65.36	12,856

Table 10
Results in terms of accuracy (%) using the best feature combination from Table 9 (all + prefix + mid-word 3-grams) as baseline and discarding the worst predictive 4-gram categories (Table 8) and their combinations. Threshold ⩾ 50, binary representation, SVM algorithm.

Feature set	10-fold	50%-test	N of Features
Best: All + prefix + mid-word 3-grams	70.88	64.88	12,211
Best – suffix 4-grams	**70.88**	64.56	11,165
Best – multi-word 4-grams	70.80	**65.60**	10,668
Best – end-punct 4-grams	70.72	65.12	11,870
Best – suffix – multi-word 4-grams	70.72	64.40	9,622
Best – suffix – end-punct 4-grams	70.72	65.04	10.824
Best – multi-word – end-punct 4-grams	70.48	65.04	10,327
Best – suffix – multi-word – end-punct 4-grams	70.64	64.08	9.281

marks are separated from adjacent characters by space and included in the middle-punctuation category of typed *n*-grams, which produces different *n*-grams [22]; *e.g.*, in the sample sentence (1), the instance `em?` constitutes just one untyped 3-gram, but it corresponds to two typed 3-grams: `em?` (end-punct) and `_?_` (mid-punct). Results are shown in Table 11. Using untyped 3- and 4-grams in combination with typed 3-grams yielded the best performance so far (72.16%). Thus, this combination was selected for further experiments described in the next subsection.

5.3 Further Experiments

5.3.1 Introducing Some Pre-Processing Steps

Pre-processing has proved to be a useful strategy for AA and related tasks [11, 20, 22, 31]. In this paper, two pre-processing steps were examined: (i) replacing digits and (ii) discarding text inside quotations, before extracting character *n*-gram features. The first method consists in replacing each digit by '0' (ex., 12,345 → 00,000) aiming at capturing the number format but not the actual number [22]; the second procedure aims at discarding information that does not reflect the author's writing style. Finally, the two proposed steps were combined. Results are shown in

Table 11
Results in terms of accuracy (%) combining typed and untyped n-grams with $n = 3$ and 4. Threshold $\geqslant 50$, binary representation, SVM algorithm.

Model	10-fold	50%-test	N of Features
Untyped 3-grams + untyped 4-grams	71.68	66.00	15,194
Untyped 3-grams + untyped 4-grams + typed 3-grams	**72.16**	65.20	20,607
Untyped 3-grams + untyped 4-grams + typed 4-grams	71.60	66.08	24,445
Typed 3-grams + untyped 3-grams	70.96	63.84	10,072
Typed 3-grams + untyped 4-grams	71.52	65.36	15,948
Typed 3-grams + typed 4-grams	71.68	65.60	14,651
Typed 3-grams + typed 4-grams + untyped 3-grams	71.60	65.52	19,310
Typed 3-grams + typed 4-grams + untyped 4-grams	71.68	**66.24**	25,186
Typed 4-grams + untyped 4-grams	70.72	65.76	19,786
Typed 4-grams + untyped 3-grams	71.68	66.00	13,910
Typed 4-grams + typed 3-grams + untyped 3-grams + untyped 4-grams	71.68	65.92	29,845

Table 12. The proposed pre-processing steps were unable to enhance the best 10-fold cross-validation result achieved in the previous stage; however, they provided a slight improvement in the 50%-test accuracy (0.48%).

Table 12
Results in terms of accuracy (%) after applying different pre-processing steps to the best feature combination from Table 11. Threshold $\geqslant 50$, binary representation, SVM algorithm.

Pre-processing	10-fold	50%-test	N of Features
Previous best: untyped 3-grams + untyped 4-grams + typed 3-grams	72.16	65.20	20,607
Replacing digits	71.76	65.68	20,299
Discarding quotes	**72.16**	65.28	20,566
Replacing digits + discarding quotes	71.68	**65.68**	20,260

Other experiments were also carried out, namely, by converting all texts to lowercase and by replacing whole numbers by '0' (ex., 12,345 → 0) and year mentions by 'YYYY' (ex., 2016 → YYYY). These experiments, however, did not lead to improvements in accuracy (for lack or space, the results are not provided here). In future work, the impact of other pre-processing steps will be investigated, such as discarding or anonymizing named entities [7].

5.3.2 Using Other Machine-Learning Algorithms

So far, WEKA's [12] implementation of Support Vector Machines (SVM) algorithm was used. This algorithm with default parameters is considered among the best for the AA task in both the English and the Portuguese languages [23, 33, 36, 37].

Multinomial Naive Bayes (NBM) classifier, which is known to provide high results for text classification tasks [15, 22], was also examined. The J48 and Naive Bayes algorithms have also been examined but they were dismissed as they consistently showed lower results than the SVM algorithm.

Table 13 presents the results of comparing SVM with NBM performance trained on the best combination of *n*-gram categories (untyped 3-grams + untyped 4-grams + typed 3-grams) and without applying any pre-processing steps. In this case, NBM is 2.16% less accurate than SVM under 10-fold cross-validation, but it slightly outperforms SVM in the 50%-test setting. However, additional experiments using NBM classifier with various threshold values and models (untyped, typed, and affix+punctuation), as well as with different feature combinations, showed that the results obtained using NBM classifier are consistently lower than when using SVM (the results are not provided due to lack of space).

Table 13
Results in terms of accuracy (%) using SVM and NBM machine-learning algorithms with the best combination of *n*-gram categories (untyped 3-grams + untyped 4-grams + typed 3-grams), a frequency threshold *frq* $\geqslant$ 50, with binary representation and without any pre-processing.

Machine-learning algorithm	**10-fold**	**50%-test**	**N of Features**
SVM	**72.16**	65.20	20,607
NBM	70.00	**66.48**	20,607

5.3.3 Using Unbalanced Corpus

Finally, the best-performing model (untyped 3- and 4-grams + typed 3-grams, shown in Table 11) was applied to the *unbalanced* corpus (see Section 4.1, *in fine*) with the optimal parameters, selected from the best 10-fold cross-validation results, in order to compare this approach with that of Homem and Carvalho [13]. The baseline experiment was also conducted using the optimal settings for the bag-of-words (BoW) approach: frequency threshold *frq* = 0 and binary feature representation (see Table 4). Table 14 shows the results obtained using SVM and NBM classifiers.

Table 14
Results in terms of accuracy (%) when using the unbalanced corpus without any pre-processing: (1^{st} row) the bag-of-words (BoW) baseline approach (frequency threshold *frq* $\geqslant$ 0, binary representation, and SVM algorithm); (2^{nd} and 3^{rd} rows) comparing SVM and NBM algorithms using the best feature combination from Table 11 with the frequency threshold *frq* $\geqslant$ 50 and binary representation.

Unbalanced corpus	**10-fold**	**50%-test**	**N of Features**
Bag-of-words, SVM (baseline)	62.53	57.27	68,429
Untyped 3-grams + untyped 4-grams + typed 3-grams (SVM)	**64.99**	**60.84**	29,276
Untyped 3-grams + untyped 4-grams + typed 3-grams (NBM)	59.98	57.16	29,276

Using the best model with the unbalanced corpus yielded an accuracy of 64.99% under 10-fold cross-validation and 60.84% in the 50%-test setting. The SVM algorithm, when compared with the results obtained with the balanced corpus, shows a 7.17% and 4.36% drop in accuracy (under 10-fold cross-validation and in the 50%-test setting, respectively), while the accuracy of the BoW baseline approach drops 7.87% and 4.57% (10-fold and 50%-test, respectively). In spite of the sophisticated algorithm proposed by Homem and Carvalho [13], who also used the unbalanced corpus, their results were systematically below the 60% threshold. The BoW baseline approach with the optimal parameters selected in this paper showed results similar to those reported by these authors, while this paper approach based on (typed and untyped) character *n*-grams outperforms them.

Conclusions

The Authorship Attribution (AA) task aims at identifying the author of a text based on text samples from known authors. This paper demonstrated that character *n*-gram features are highly predictive for the AA task in Portuguese. It showed that the combination of character *n*-grams of different types and length, along with an appropriate selection of threshold values, feature representation, and machine-learning algorithm, allows one to achieve high performance in this task. The best result was achieved when training SVM classifier on the combination of untyped character 3-grams, untyped character 4-grams, and typed character 3-grams, using binary feature representation and considering only those features that occur at least 50 times in the training corpus. This language-independent approach, with a commonly used SVM algorithm, outperformed both the bag-of-words baseline and previous approaches using the same corpus.

Varying frequency threshold values did not lead to significant improvements on the results. However, it allowed an average reduction of 88% of the entire feature set without loss of accuracy. Moreover, the paper demonstrated that the parameters selected by 10-fold cross-validation using only the training corpus provide near-optimal results when used in the 50%-test setting. The paper also showed that the conclusion of Sapkota *et al.* [29] that affix+punctuation character *n*-grams perform better than when using all *n*-grams is not valid for the Portuguese language. Still, more experiments using other corpora are required to verify this conclusion.

Finally, it was demonstrated that feature representation is an important aspect for the AA task in Portuguese. Binary feature representation, in all examined cases, provided higher results than term frequency (an average increase of 4.71% under 10-fold cross-validation and of 5.44% in the 50%-test setting using the balanced corpus). Therefore, in future work, alternative feature representation techniques will be tested, such as second order representation [17] or doc2vec-based feature representation [16]. The later has proved to provide good results for AA in English [24] and in related tasks [21]. The proposed approach will also be examined under cross-topic and cross-genre AA conditions.

Acknowledgements

This work was supported by the Mexican Government (Conacyt projects 240844 and 20161958, SIP-IPN 20171813, 20171344, and 20172008, SNI, COFAA-IPN) and by the Portuguese Government, through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013. The authors express their gratitude to

J.P. Carvalho and his collaborators, who kindly made the corpus available for this paper.

References

[1] Almeida, D.: Atribuição de autoria com propósitos forenses. ReVEL Revista Virtual de Estudos de Linguagem, vol. 12, no. 23, 2014, pp. 148–186

[2] Barbon, S., Igawa, R., and Zarpelão, B.: Authorship verification applied to detection of compromised accounts on online social networks. Multimedia Tools and Applications, 2016, pp. 1–21

[3] Bhargava, M., Mehndiratta, P., and Asawa, K.: Stylometric analysis for authorship attribution on twitter. Proceedings of the 2nd Intl. Conf. on Big Data Analytics, Springer, LNCS, vol. 8302, 2013, pp. 37–47

[4] Chaski, C.: Best Practices and Admissibility of Forensic Author Attribution. Journal of Lay and Policy, vol. 21, no. 2, 2013, pp. 1333–1376

[5] Coulthard, M. and Johnson, A. (Eds.): The Routledge Handbook of Forensic Linguistics, Routledge, 2010

[6] Daelemans, W.: Explanation in computational stylometry. Proceedings of the 14th Intl. Conf. on Intelligent Text Processing and Computational Linguistics, CIC-Ling 2013, 2013, pp. 451–462

[7] Dias, F., Baptista, J., and Mamede, N.: Automated Anonymization of Text Documents. IEEE World Congress Computational Computational Inteligence/Intelligence Methods for NLP, 2016, pp.1287-1294

[8] Escalante, H., Solorio, T., and Montes-y-Gómez, M.: Local histograms of character n-grams for authorship attribution. Proceedings of ACL–HLT 2011, 2011, pp. 288–298

[9] Filho, J., Pasti, R., and de Castro, L.: Gender classification of twitter data based on textual meta-attributes extraction. New Advances in Information Systems and Technologies, Springer, 2016, vol. 444, pp. 1025–1034

[10] Gómez-Adorno, H., Sidorov, G., Pinto, D., Vilariño, D., and Gelbukh, A.: Automatic authorship detection using textual patterns extracted from integrated syntactic graphs. Sensors, vol. 16, no. 9, 2016

[11] Gómez-Adorno, H., Markov, I., Sidorov, G., Posadas-Durán, J.-P., Sanchez-Perez, M., and Chanona-Hernandez, L.: Improving feature representation based on a neural network for author profiling in social media texts. Computational Intelligence and Neuroscience, vol. 2016, 2016, 13 pages

[12] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I.: The WEKA data mining software: An update. SIGKDD Explorations, vol. 11, no. 1, 2009, pp. 10–18

[13] Homem, N. and Carvalho, J.P.: Web user identification with fuzzy fingerprints. Proceedings of the IEEE Intl. Conf. on Fuzzy Systems, IEEE Xplorer, 2011, pp. 2622–2629

[14] Kestemont, M.: Function words in authorship attribution. From black magic to theory? Proceedings of the 3rd Workshop on Computational Linguistics for Literature, EACL 2014, 2014, pp. 59–66

[15] Kibriya, A., Frank, E., Pfahringer, B., and Holmes, G.: Multinomial naive Bayes for text categorization revisited. Proceedings of the 17th Australian Joint Conf. on Advances in AI, AI 2004, 2005, pp. 488–499

[16] Le, Q. and Mikolov, T.: Distributed representations of sentences and documents. Proceedings of the 31st Intl. Conf. on Machine Learning, ICML 2014, 2014, pp. 1188–1196

[17] López-Monroy, A., Montes-y-Gómez, M., Escalante, H., Villaseñor-Pineda, L., and Stamatatos, E.: Discriminative subprofile-specific representations for author profiling in social media. Knowledge-Based Systems, vol. 89, 2015, pp. 134–147

[18] Luyckx, K. and Daelemans, W.: Authorship attribution and verification with many authors and limited data. Proceedings of the 22nd Intl. Conf. on Computational Linguistics, COLING 2008, 2008, pp. 513–520

[19] Marquilhas, R., and Cardoso, A.: O estilo do crime: A análise de texto em estilística forense. XXVII Encontro Nacional da Associação Portuguesa de Linguística – Textos selecionados, 2011, pp. 416–436

[20] Markov, I., Gómez-Adorno, H., and Sidorov, G.: Language- and subtask-dependent feature selection and classifier parameter tuning for author profiling. Working Notes Papers of the CLEF 2017 Evaluation Labs. CEUR Workshop Proceedings, vol. 1866, CLEF and CEUR-WS.org, 2017

[21] Markov, I., Gómez-Adorno, H., Posadas-Durán, J.-P., Sidorov, G., and Gelbukh, A.: Author profiling with doc2vec neural network-based document embeddings. Proceedings of the 15th Mexican Intl. Conf. on Artificial Intelligence, MICAI 2016, vol. 10062, Part II, LNAI, Springer, 2017, pp. 117–131

[22] Markov, I., Stamatatos, E., and Sidorov, G.: Improving cross-topic authorship attribution: The role of pre-processing. Proceedings of the 18th Intl. Conf. on Computational Linguistics and Intelligent Text Processing, CICLing 2017, LNCS, Springer, 2017, in press

[23] Pavelec, D., Justino, E., and Oliveira, L.: Author identification using stylometric features. Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial, vol. 11, no. 36, 2007, pp. 59–66

[24] Posadas-Durán, J.-P., Gómez-Adorno, H., Sidorov, G., Batyrshin, I., Pinto, D., and Chanona-Hernández, L.: Application of the distributed document representation in the authorship attribution task for small corpora. Soft Computing, 2016, pp. 1–13

[25] Qian, T., Liu, B., Chen, L., and Peng, Z.: Tri-training for authorship attribution with limited training data. Proceedings of ACL 2014, 2014, pp. 345–351

[26] Ramnial, H., Panchoo, S., and Pudaruth, S.: Authorship attribution using stylometry and machine learning techniques. Intelligent Systems Technologies and Applications, Springer, Advances in Intelligent Systems and Computing, vol. 384, 2016, pp. 113–125

[27] Rocha, A., Scheirer, W., Forstall, C., Cavalcante, T., Theophilo, A., Shen, B., Carvalho, A., and Stamatatos, E.: Authorship attribution for social media forensics. IEEE Transactions on Information Forensics and Security, 2016

[28] Ruder, S., Ghaffari, P., and Breslin, J.: Character-level and multi-channel convolutional neural networks for large-scale authorship attribution. arXiv preprint arXiv:1609.06686, 2016

[29] Sapkota, U., Bethard, S., Montes-y-Gómez, M., and Solorio, T.: Not all character n-grams are created equal: A study in authorship attribution. Proceedings of NAACL–HLT 2015, 2015, pp. 93–102

[30] Schwartz, R., Tsur, O., Rappoport, A., and Koppel, M.: Authorship attribution of micro-messages. Proceedings of EMNLP 2013, 2013, pp. 1880–1891

[31] Sidorov, G., Ibarra Romero, M., Markov, I., Guzman-Cabrera, R., Chanona-Hernández, L., and Velásquez, F.: Detección automática de similitud entre programas del lenguaje de programación Karel basada en técnicas de procesamiento de lenguaje natural. Computación y Sistemas, vol. 20, no. 2, 2016, pp. 279–288

[32] Sousa-Silva, R., Sarmento, L., Grant, T., Oliveira, E., and Maia, B.: Comparing Sentence-Level Features for Authorship Analysis in Portuguese. Computational Processing of the Portuguese Language, Springer, LNAI vol. 6001, 2010, pp. 51–54

[33] Sousa-Silva, R., Laboreiro, G., Sarmento, L., Grant, T., Oliveira, E., and Maia, B.: ‘twazn me!!! ;(’ Automatic authorship analysis of micro-blogging messages. Proceedings of the 16th Intl. Conf. Application of Natural Language to Information Systems, Springer, vol. 6716, 2011, pp. 161–168

[34] Stamatatos, E.: Author identification using imbalanced and limited training texts. Proceedings of the 18th Intl. Conf. on Database and Expert Systems Applications, DEXA 2007, 2007, pp. 237–241

[35] Stamatatos, E.: On the robustness of authorship attribution based on character n-gram features. Journal of Law & Policy, vol. 21, no. 2, 2013, pp. 427–439

[36] Teles, L.: Atribuição de autoria em linguística forense: uma análise combinada para identificação de autor através do texto. Master thesis, Universidade de Lisboa, 2016

[37] Varela, P., Justino, E., Bortolozzi, F., and Oliveira, L.: A computational approach based on syntactic levels of language in authorship attribution. IEEE Latin America Transactions, vol. 14, no. 1, 2016, pp. 259–266

Algorithm for Extraction of Subtrees of a Sentence Dependency Parse Tree

Juan-Pablo Posadas-Durán[1], Grigori Sidorov[2], Helena Gómez-Adorno[2], Ildar Batyrshin[2], Elibeth Mirasol-Mélendez[3], Gabriela Posadas-Durán[1], Liliana Chanona-Hernández[1]

[1]Instituto Politécnico Nacional (IPN), Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Zacatenco (ESIME-Zacatenco),
Av. Luis Enrique Erro S/N 07738, Mexico City, Mexico.

[2]Instituto Politécnico Nacional, Centro de Investigación en Computación,
Av. Juan de Dios Bátiz 07738, Mexico City, Mexico.

[3]Instituto Politécnico Nacional, Escuela Nacional de Medicina y Homeopatía ,
Guillermo Massieu Helguera 239, 07320, Mexico City, Mexico.

E-mail: jdposadas@esimez.mx, sidorov@cic.ipn.mx, batyr1@cic.ipn.mx,
hgomeza1400@alumno.ipn.mx, emirasolm0800@alumno.ipn.mx,
gposadasd@ipn.mx, lchanonah@ipn.mx

Abstract: In this paper, we introduce an algorithm for obtaining the subtrees (continuous and non-continuous syntactic n-grams) from a dependency parse tree of a sentence. Our algorithm traverses the dependency tree of the sentences within a text document and extracts all its subtrees (syntactic n-grams). Syntactic n-grams are being successfully used in the literature (by ourselves and other authors) as features to characterize text documents using machine learning approach in the field of Natural Language Processing.

Keywords: syntactic n-grams; subtrees extraction; tree traversal; linguistic features

1 Introduction

Stylometry is an active research field that studies how to model the style of an author from a linguistic point of view by proposing reliable features based on the use of the language. These features, known as style markers, characterize the writing style of an author and are used to solve various tasks in the field of Natural Language Processing like authorship attribution [1], author profiling [2, 3, 4], author verification [5], author clustering [6], and plagiarism detection [7, 8, 9], among others.

The problem of authorship characterization can be tackled using different approaches. The majority of the proposed methods use n-gram based, morphological and lexical characteristics that only exploit the superficial information of a text. Let us remind that n-grmas are sequences of elements as they appear in the texts, they can be formed by words/lemmas/stems, characters, or POS tags.

Traditional approaches ignore syntactic features despite the fact that the syntactic information is topic independent and therefore is robust to characterize style of an author. Note that the surface representation is prone to noise, for example, insertion of a subordinate clause or an adjective changes the n-grams, while the syntactically based features handle this problem correctly.

In this paper, we present an algorithm, which uses syntactic information contained in dependency trees to extract complete syntactic n-grams. The main idea is that we form n-grams by following the paths in the dependency tree, instead of taking them in the order of their appearance at the surface level. Dependency trees represent the syntactic relations between words, which form the sentence. The proposed algorithm builds complete syntactic n-grams in a general manner. It considers both types of syntactic n-grams: continuous [10] and non-continuous [11], which is called complete syntactic n-grams. The difference between them is that non-continuous n-grams have bifurcations, i.e., the corresponding subtrees have several branches, while continuous n-grams represent exactly one branch. The reason for this distinction is the supposition that there is different linguistic reality in each case. In addition, syntactic n-grams can be formed by various types of elements, like words/lemmas/stems, POS tags, dependency tags or a combination of them. Note that dependency tags are not used in traditional n-grams.

The algorithm extracts the syntactic n-grams from a dependency tree by performing a two-stage procedure. The first stage the algorithm conducts a breadth-first search of the tree and finds all the subtrees of height equal to 1. In the second stage, the algorithm traverses the tree in postorder replacing the node occurrence in a subtree with the subtrees from higher levels where the node is the root. The extracted subtrees correspond to syntactic n-grams of the tree.

We implemented our algorithm in Python and made it freely available at our website[1]. Note that though we presented the idea of syntactic n-grams in our previous works, until now we did not present the description of the algorithm used for their extraction. It is worth mentioning that the implementation of the algorithm was freely available for three years and it was used by other researches who used syntactic n-grams.

The rest of the paper is organized as follows. The concept of complete syntactic n-grams is discussed in Section 2. Section 3 describes the use of the syntactic n-grams in various problems related to Natural Language Processing. The algorithm for extraction of syntactic n-grams (all subtrees of a dependency parse tree) is presented in Section 4. Finally, in the last section of the paper, we draw conclusions and discuss directions of future work.

[1] `http://www.cic.ipn.mx/~sidorov/MultiSNgrams_3.py`

2 The Idea of Syntactic N-grams

As we mentioned in the previous section, the concept of syntactic n-grams (sn-grams) was first introduced in [11, 10] as an alternative idea to the well-known representation based on character n-grams or word n-grams. Standard character or word n-grams are sequences of elements extracted from a given text by using an imaginary window of size *n*, which slides over the text with certain offset, typically equal to 1. On the other hand, syntactic n-grams are the paths of size *n* generated by following the branches of a dependency tree, i.e., they do not depend on the surface order of elements. Syntactic n-grams are extracted by traversing the sentence dependency trees of a text and correspond to all subtrees of the dependency tree. Syntactic n-grams capture the syntactic relations between words in a sentence.

Two important differences can be observed between syntactic and traditional n-grams: (1) syntactic n-grams are able to capture information (internal information), which traditional n-grams cannot access (they use only the surface information), (2) each syntactic n-gram always has a meaning from a linguistic point of view (i.e., there is always underlying grammar that was used for parsing), unlike the traditional n-grams, which do not have it in many cases (i.e., there are many n-grams that just represent noise because there position of one near the other is a pure coincidence).

The syntax of a text is the way, in which words relate to each other to express some idea as well as the function that they have within a text. The relations that exist between the words of a sentence can be represented by the two grammatical formalisms: dependency or constituency grammars [12]. In modern research, the dominant approach is dependency analysis, though these formalisms are equivalent, i.e., their representations can be transformed one into another.

The dependency grammar shows the relations between pairs of words, where one of them is the head word and the other word is the dependent. It is represented as a tree structure that starts with a root node (generally the verb of the sentence), which is the head word of more general order. Then, the arc are used between the head and the dependent words according to dependency relations between them. The dependent words, in turn, are considered as head words and their dependent words are added, thus generating a new level in the tree. The structure described above is known as dependency tree [13].

In order to illustrate the concept of syntactic n-grams and the differences between them and the standard n-grams lets consider the sentence: ”*Victor sat at the counter on a plush red stool*”. We get the following standard word 3-grams using the sliding imaginary window method: *“Victor sat at”, “sat at the”, “at the counter”, “the counter on”, “counter on a”, “on a plush”, “a plush red”, “plush red stool”.*

To extract the syntactic n-grams of any sentence it is necessary first to process the sentence by a syntactic parser. We processed the sentence using the Stanford CoreNLP toolkit [14] and as the result we obtain the lemmas, POS tags, dependency relations tags and the relations between the elements of the sentence.

A dependency tree structure $T = (V, E)$ with root v_0 is obtained from the processed sentence, where the set of nodes $V = \{v_0, v_1, \ldots, v_i\}$ correspond to the words of the sentence and the set of branches $E = \left\{e_0, e_1, \ldots, e_j\right\}$ correspond to dependency relations between words.

Table 1 shows the standard syntactic information that can be gathered from the example sentence. Note that the column *Dependent* denotes the set of nodes of dependent words, the column *Head* corresponds to the nodes of the head word and the row *Leaves* shows the set of nodes of words without dependents (leaves). The root node (v_0) is denoted by the tag *root* in the *SR* column. Figure 1 shows graphical representation of the dependency tree T for the example sentence. It depicts the relations between words using a black arrow where the tail of the arrow denotes the head word, and the head of the arrow denotes the dependent word.

Table 1
Syntactic information obtained from the sentence "*Victor sat at the counter on a plush red stool*"

Id	**Word**	**Lemma**	**POS**	**Head**	**SR**	**Dependent**
1	Victor	Victor	NNP	2	nsubj	
2	sat	sit	VBD	0	root	$[1,3,6]$
3	at	at	IN	2	prep	$[5]$
4	the	the	DT	5	det	
5	counter	counter	NN	3	pobj	$[4]$
6	on	on	IN	2	prep	$[10]$
7	a	a	DT	10	det	
8	plush	plush	JJ	10	amod	
9	red	red	JJ	10	amod	
10	stool	stool	NN	6	pobj	$[7,8,9]$
Leaves (nodes without children): $[1,4,7,8,9]$						

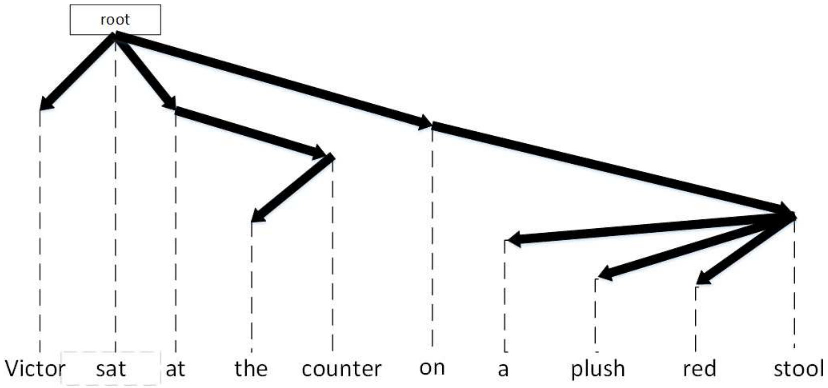


Figure 1
Dependency tree of the sentence "*Victor sat at the counter on a plush red stool*"

The set of labels of dependency relations used by the CoreNLP toolkit is described in [15] and the set of POS tags is described in [16].

The syntactic n-grams can be homogeneous or heterogeneous. We call them homogeneous when they are constructed of the same type of elements, for example, only of words, or only of POS tags. They are heterogeneous, when various types of elements are combined in the same syntactic n-gram.

For the example sentence, we extract syntactic n-grams of size 3 considering the homogeneous case of words. The syntactic n-grams are extracted by traversing the dependency tree and identifying all the subtrees with exactly three nodes. The syntactic n-grams are codified using the metalanguage proposed in [11]. The metalanguage is simple: the head element is on the left of a square parenthesis and inside there are the dependent elements; the elements at the same level are separated by a coma. The syntactic n-grams extracted are: *[Victor,at], sat[Victor,on], sat[at,on], sat[on[stool]], sat[at[counter]], on[stool[plush]], on[stool[a]], on[stool[red]], stool[a,plush], stool[a,red], stool[plush, red], sat[at[counter]], at[counter[the]].*

If we consider heterogeneous syntactic n-grams, then we combining elements of different nature. This kind of sn-grams is obtained by setting one type of information for the head element and use a different type for the rest of elements. The syntactic analysis offers the possibility to work with words, lemmas, POS tags and dependency relation tags (DR tags). So, there are 12 possible pairs for heterogeneous syntactic n-grams that can be extracted [11, 17]: (words, lemmas), (words, POS), (words, DR), (lemmas, words), (lemmas, POS), (lemmas, DR), (POS, words), (POS, lemmas), (POS, DR), (DR, words), (DR, lemmas), and (DR, words).

Table 2 shows the heterogeneous syntactic n-grams (words, POS) of size 3 extracted from the previous example sentence along with the homogeneous syntactic n-grams of words. The heterogeneous syntactic n-grams are able to identify new patterns due to the fact that they combine information from different contexts, for example the sn-grams *sat[NNP, IN], sat[IN [NN]], on[NN[JJ]]* are patterns that occur more frequently compared to homogeneous sn-grams.

Table 2
Comparing sn-grams of words vs. sn-grams of (word, POS)

Words	(words, POS)
sat[Victor,at]	sat[NNP,IN]
sat[Victor,on]	sat[NNP,IN]
sat[at,on]	sat[IN,IN]
sat[on[stool]]	sat[IN[NN]]
sat[at[counter]]	sat[IN[NN]]
on[stool[plush]]	on[NN[JJ]]
on[stool[a]]	on[NN[DT]]
on[stool[red]]	on[NN[JJ]]
stool[a,plush]	stool[DT,JJ]
stool[a,red]	stool[DT,JJ]
stool[plush,red]	stool[JJ,JJ]
sat[at[counter]]	sat[IN[NN]]
at[counter[the]]	at[NN[DT]]

The heterogeneous sn-grams are not restricted neither to the way of combining the syntactic information (one type for the head and another type for the dependents) nor to the type of information mentioned before (words, lemmas, POS tags and DR tags). In general, it is possible to use different kind of information at each level of the subtree, for example, we can use words in the head, POS tags for the elements at level 1, lemmas for the ones at level 2 and so on.

3 Syntactic N-grams as Features for Natural Language Processing Tasks

The written language provides various levels of language description: semantic, syntactic, morphological, lexical, etc. In different works [18, 19], textual feature representation are classified into one of the following categories: character level, lexical level, syntactic level, semantic level, and format level. The syntactic n-grams fall into the syntactic level category, which is one of the least used levels of text representation for automatic analysis.

Although written language offers a wide range of possibilities for characterization, most of the works in the literature focus on morphological and lexical information because of their direct availability. Some examples of text features that have been proposed in the literature are: size of the sentences, size of tokens, token frequency, frequency characters, richness of used vocabulary, character n-grams, word n-grams, among others [20].

As we already mentioned, the main idea of syntactic n-grams is to follow the path in a syntactic tree for obtaining the sequence of elements, instead of using the surface order. It allows to bridge syntactic knowledge with the machine learning methods in modern Natural Language Processing.

In the first paper on syntactic n-grams, continuous syntactic n-grams was introduced in [21, 10] for text classification tasks. The authors evaluated the sn-grams in the task of authorship attribution and compared their performance against traditional n-grams of characters, words, and POS tags. The corpus used in their experiments includes English texts from the Project Gutenberg. For the classification purposes, they used three algorithms: Support Vector Machines (SVM), Naive Bayes (NB) and Decision Trees (J48). The sn-grams features gave better results with SVM classifier over the other traditional features.

There were two research works on grammatical error correction using syntactic n-grams. The first work, developed by Sidorov [22], presents a methodology that applies a set of simple rules for correction of grammatical errors using sn-grams. The methodology achieved acceptable results on the CONLL Shared Task 2013. The second work, by Hernandez et al. [23], used a language model based on syntactic 3-grams and 2-grams extracted from dependency trees generated from 90% of the English Wikipedia. Their system ranked 11th on the CONLL Shared Task 2014.

Author profiling (AP) is another task related to the authorship attribution. The aim of AP is to determine author's demographics based on a sample of his writing. In [5] the authors present an approach to tackle the AP task at PAN 2015 competition [24]. The approach relies on syntactic based n-grams of various types in order to predict the age, gender and personality traits of the author of a given tweet. The obtained results indicate that the use of syntactic n-grams along with other specific tweet features (such as number of retweets, frequency of hashtags, frequency of emoticons, and usage of referencing URLs) are suitable for predicting personal traits. However, their usage is not that successful when predicting the age and gender.

In [25], the authors explore the use of syntactic n-grams for the entrance exams question answering task at CLEF. They used syntactic n-grams as features extracted from Syntactic Integrated graphs and Levenshtein distance as the similarity measure between n-grams, measured either in characters or in elements of n-grams. Their experiments show that the soft cosine measure along with syntactic n-grams provides better performance in this case study.

In the paper by Calvo et al. [26] the authors compared the constituency based syntactic n-grams against the dependency based syntactic n-grams for paraphrase recognition. They presented a methodology that combines sn-grams with different NLP techniques (synonyms, stemming, negation handling and stopwords removal). Both types of sn-grams were evaluated independently and compared against state-of-the-art-approaches. Syntactic n-grams outperformed several works in the literature and achieved an overall better performance compared with traditional n-grams in paraphrase recognition. In most cases, syntactic constituent n-grams yielded better scores than syntactic dependency n-grams.

The research work of Laippala et al. [27], studies the usefulness of syntactic n-grams for corpus description in Finnish, including literature texts, Internet forum discussion for social media and newspapers' websites. Their results suggests that in comparison with traditional feature representation, syntactic n-grams offer both complementary information generalizing beyond individual words to concepts and information depending on syntax not reached by lexemes.

A recent work on statistical machine translation (SMT) [28] proposes a relational language model for dependency structures that is suited for languages with a relatively free word order. The authors empirically demonstrate the effectiveness of the approach in terms of perplexity and as a feature function in string-to-tree SMT from English to German and Russian. In order to tune the log-linear parameters of the SMT they use a syntactic evaluation metric based on syntactic n-grams, which increases the translation quality when coupled with a syntactic language model.

In the book "Prominent Feature Extraction for Sentiment Analysis" [29], the authors explore semantic, syntactic and common-sense knowledge features to improve the performance of sentiment analysis systems. This work applies sn-grams for sentiment analysis for the first time. The authors show that syntactic n-grams are more informative and less arbitrary as compared to traditional n-grams. In their experiments, syntactic n-gram feature set produced an F-measure of 86.4% with BMNB classifier for movie review dataset. This feature set performed well as compared to other simple feature extraction techniques like unigrams, bigrams, bi-tagged, and dependency features.

As it can be seen, syntactic n-grams are being used successfully in a wide range of NLP tasks. In order to increment the research on the use of syntactic n-grams as feature representation we introduce the detailed description of the algorithm for extracting complete syntactic n-grams from a sentence parse tree.

4 Algorithm for Extraction of Syntactic N-grams

We mentioned the algorithm for the extraction of sn-grams in our previous works, for example, in [17], but it is the first time that we give it's detailed description. The algorithm handles homogeneous and heterogeneous variants of continuous and non-continuous syntactic n-grams. The algorithm is implemented in Python, we provide the full implementation of this algorithm at our website (see Section 1).

We established in Section 2 that syntactic n-grams are extracted from a dependency tree structure and they correspond to the subtrees of a tree, i.e., given a dependency tree $T = (V,E)$ with the root v_0, all syntactic n-grams are the set of subtrees $ST = \{st_0, st_1, \ldots, st_k\}$ of the tree with the restriction that each st_i must be of size n. Basically, the algorithm traverses the dependency tree in order to find all the subtrees.

The algorithm consists of two stages: first stage performs a breadth-first search over the tree and extracts all the subtrees of height equal to 1, second stage traverses the tree in postorder replacing the node occurrence in a subtree of lower level with the subtrees from higher levels where the node is the root so that subtrees with height greater than 1 are extracted. The algorithm discriminates those subtrees that do not satisfy the restriction of size n.

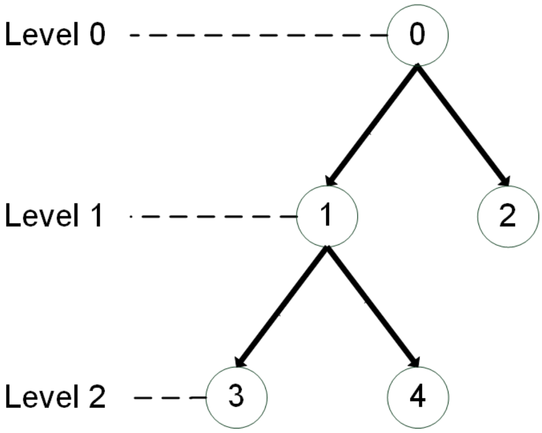


Figure 2
Sample tree

To illustrate how the algorithm extracts the syntactic n-grams, let consider the tree T shown in Figure 2. We express the subtrees according to the proposed metalanguage in [11]. If we perform the first stage of the algorithm to the tree T, we obtain at level

0 the subtrees *0[1], 0[2], 0[1,2]* and *1[3], 1[4], 1[3,4]* at level 1. Note that all the subtrees extracted in the first stage of the algorithm have a height equal to 1.

Then we continue performing the second stage of the algorithm and traverse the tree *T* in preorder replacing the nodes of the subtrees in level 0 with the subtrees in level 1 that has the node as the root element. Observe that only the node 1 satisfies the condition and can be modified in the subtrees of the lower level. After accomplished the second stage, we obtain the following subtrees: *0[1[3]], 0[1[4]], 0[1[3,4]], 0[1[3],2], 0[1[4],2], 0[1[3,4],2]*.

All the possible subtrees of size greater than 1 are generated by the algorithm. These subtrees may become into syntactic n-grams after adding linguistic information. The case of syntactic n-grams of size equal to 1 represent the words of the sentence and are not meaningful for the scope algorithm.

The algorithm is organized into seven functions. The primary function named *EXTRACT_SNGRAMS* is described in the Algorithm 8, and it coordinates the call of the other functions so the two stages of the described algorithm are performed. It is the interface that user utilize to extract the sn-grams.

The function *EXTRACT_SNGRAMS* calls to *GET_SUBTREES* function described in Algorithm 1 that performs the first stage of the proposed algorithm. The *GET_ SUBTREES* function in turn calls the functions *NEXTC* (Algorithm 2) and *NUM_COMBI NATION* (Algorithm 3), both auxiliary functions.

Then the second stage of the algorithm is realized by the function *COMPOUND_SN GRAMS* (Algorithm 5). Finally functions *LEN_SNGRAM* (Algorithm 4) and *PREPARE_SN GRAMS* (Algorithm 6 and 7) rewrite the subtrees extracted into one of the types of sn-grams.

The input of the algorithm is a plain text that contains syntactic information of a sentence and returns the syntactic n-grams together with their frequency of occurrence. The algorithm requires a syntactic parser, which is an external tool often used in many Natural Language Processing problems.

Various syntactic parsers are available, like Standford CoreNLP, Freeling, Connexor, to name some, and although all of them retrieve the same syntactic information, there is no standard for the output. It is merele a technical detail, but we would like to mention that the our code handles the outputs of the Stanford parser. For Freeling, we generated another Python script, which converts its output to the output of the Stanford parser.

The output generated by the Stanford CoreNLP has the following format: in one section it contains the words, lemmas and POS tags of a sentence and in another section it contains the dependency relation tags and dependency tree structure codified as a list of pairs: head node and dependent node.

Algorithm 1 presents the function *GET_SUBTREES*, which receives an adjacency table with the syntactic information of a sentence (as shown in Table 1) and returns a list of codified subtrees bounded in size by two parameters (minimum and maximum size). The function also requires the indexes of the nodes that can be roots of the

Algorithm 1 Function *GET_SUBTREES*

Parameters: *sentence* (syntactic information matrix), *subroots* (possible roots of subtrees), *min_size* (minimum size), *max_size* (maximum size), *max_num_children* (maximum number of children to be consider for a node).

Output: the list of the subtrees' indexes for a sentence dependency tree.

```
function GET_SUBTREES(sentence, min_size, max_size)
    Vars: $unigrams[\,], combinations[\,], counter = 0, aux[\,]$
    for all $node$ in $subroots$ do
        if $max\_num\_children != 0$ then
            $aux \leftarrow [\,]$
            $counter \leftarrow 0$
            for all $child$ in $sentence.children[node]$ do
                if $min\_size < 2$ or $max\_size == 0$ then
                    $unigrams.add([child])$
                end if
                $aux.add(child)$
                $counter \leftarrow counter + 1$
                if $counter > max\_num\_children$ then
                    $aux.pop()$
                    $combinations.add($
                        $NEXTC(node, sentence.children[node], sentence.leaves))$
                    $counter \leftarrow 0$, $aux \leftarrow [\,]$
                    $aux.add(child)$
                end if
            end for
            if $length(aux) > 0$ then
                $combinations.add($
                    $NEXTC(node, sentence.children[node], sentence.leaves))$
            end if
        else
            $combinations.add($
                    $NEXTC(node, sentence.children[node], sentence.leaves))$
            for all $child$ in $sentence.children[node]$ do
                if $min\_size < 2$ or $max\_size == 0$ then
                    $unigrams.add([child])$
                end if
            end for
        end if
    end for
    return $unigrams, combinations$
end function
```

subtrees (difference between set of nodes and set of leaves). Note that the extracted subtrees are codified keeping the natural order of occurrence of words (from left to

Algorithm 2 Function *NEXTC*

Parameters: *idx* (index of the node), *children* (children nodes), *leaves* (leaves nodes).

Output: the list with the indexes of all subtrees in the tree.

```
function NEXTC(idx, children, leaves)
    Vars: $ngram[\,], options[\,], combination[\,], list[\,], val\_max, m$
    for all $r$ in $[1, length(children)]$ do
        for all $j$ in $[1, r+1]$ do
            $combination[j-1] \leftarrow j-1$
        end for
        $options \leftarrow [\,], ngram \leftarrow [\,], ngram.add(idx, -delizq-)$
        for all $z$ en $[0, r]$ do
            $ngram.add(children[combination[z]])$
            if $children[combination[z]] \notin leaves$ then
                $options.add(children[combination[z]])$
            end if
            $ngram.add(-delsep-)$
        end for
        $ngram.add(-delder-), list.add(ngram, options)$
        $top \leftarrow NUM\_COMBINATION(length(children), r)$
        for all $j$ in $[2, top+1]$ do
            $m \leftarrow r, val\_max \leftarrow length(children)$
            while $combination[m-1]+1 == val\_max$ do
                $m \leftarrow m-1, val\_max \leftarrow val\_max-1$
            end while
            $combination[m-1] \leftarrow combination[m-1]+1$
            for all $k$ in $[m+1, r+1]$ do
                $combination[k-1] \leftarrow combination[k-2]+1$
                $options \leftarrow [\,], ngram \leftarrow [\,]$
                $ngram.add(value, -delizq-)$
                for all $z$ in $[0, r]$ do
                    $ngram.add(children[combinations[z]])$
                    if $children[combinations[z]] \notin leaves$ then
                        $options.add(children[combinations[z]])$
                    end if
                    $ngram.add(-delsep-)$
                end for
                $ngram.add(-delder-), list.add(ngram, options)$
            end for
        end for
    end for
    return $list$
end function
```

right) and the node indexes are assigned also following the same order.

The function *GET_SUBTREES* requires another function described in Algorithm 2, named *NEXTC*, which receives as input an index (that will be considered as the root node of the subtrees), a list containing indexes of its children nodes, the list of leaves nodes, and the minimum and maximum size of the subtrees. It returns the list of all extracted subtrees that contain the given root and satisfy the specified size.

Note that the extracted sn-grams are codified using the word indexes and the metalanguage proposed in [11]. In Algorithm 2 the tags $-delizq-$ and $-delder-$ denote the parenthesis [ ] in the metalanguage, which means the new level of the subtree, while the tag $-delsep-$ refers to the element *coma* which separate nodes at the same level.

An important aspect is the one related with the complexity of the algorithm. The problem has high complexity (higher than polynominal), so some sentences may be difficult to process by the algorithm because of their nature, especially those cases where nodes with a high degree (many children) are found. For example, in the sentences in which facts or things are listed, a node can have many children. In this case, the algorithm execution time will be unacceptable for practical purposes. Although in practice it is rare to find nodes with more than three children, the algorithm uses the parameter *max_num_children* to limit the number of children of a node and proceeds as follows: if the number of children is greater than the value of the parameter then only the first *max_num_children* are taken from left to right, discarding the rest of children. We set the default value of this parameter to 5.

Algorithm 3 shows the function *NUM_COMBINATION* that calculates the number of combinations of size r that can be obtained from a given list of elements sz. The algorithm 3 calculates the combinations $C(sz,r)$ implementing the equation 1:

$$C(sz,r) = \frac{sz!}{(sz-r)!r!}. \tag{1}$$

Algorithm 4 presents a function that receives a codified sn-gram using the proposed metalanguage and returns the size of the sn-gram calculated by adding the number of times the square parenthesis [and coma , appears in the sn-gram. The parameter *sngram* is the variable of string type and the function *count* () is the standard method that returns the number of times the argument occurs in the string.

Algorithm 5 introduces the function *COMPOUND_SNGRAMS* that generates new subtrees by the composition of subtrees, i.e., given a subtree with the root node v_i, it substitutes the node by the complete subtree into another subtree that contains it. The algorithm receives as parameters an adjacency table with the syntactic information of a sentence, set of initial subtrees with height equal to 1 (root node is at level 0), the minimum and maximum size of the subtrees. The function return as output a new set of subtrees obtained as the composition of subtrees (height greater than 1).

Algorithm 6 presents a function named *PREPARE_SNGRAM* that receives as parameters the sn-gram codified with the nodes indexes, the syntactic information matrix, an integer value that indicates the type of sn-gram (homogeneous or heterogeneous)

Algorithm 3 Function *NUM_COMBINATION*

Parameters: *sz* (number of elements), *r* (size of the combinations).

Output: The number of combinations of size *r* from a set of *sz* elements.

```
function NUM_COMBINATION(sz, r)
    Vars: $numerator, divisor, aux$
    if $sz == r$ then
        $numerator \leftarrow 1$
    else
        $numerator \leftarrow sz$
    end if
    for all $i$ in $[1, sz]$ do
        $numerator \leftarrow numerator \times (sz - i)$
    end for
    $aux \leftarrow r$
    for all  $i$ in $[1, r]$ do
        $aux \leftarrow aux \times (r - i)$
    end for
    $divisor \leftarrow sz - r$
    for all  $i$ in $[1, sz - r]$ do
        $divisor \leftarrow divisor \times (sz - r - i)$
    end for
    $numerator \leftarrow numerator / (aux \times divisor)$
    return $numerator$
end function
```

Algorithm 4 Function *LEN_SNGRAM*

Parameters: *sngram* (representation of syntatic n-grams).

Output: Size of the sn-gram (number of nodes that contains the sn-gram)

```
function LEN_SNGRAM(sngram)
    Vars: $n$
    $n \leftarrow 1$
    $n \leftarrow n + sngram.count([\,)$
    $n \leftarrow n + sngram.count(,)$
    return $n$
end function
```

and the information to be used (words, lemmas, POS or DR tags), as an output the function returns the sn-gram codified with the syntactic information instead of the nodes indexes.

Parameter *op* of the function *PREPARE_SNGRAM* indicates the type of sn-grams to extract: values from 0 to 3 refer to homogeneous sn-grams (of words, lemmas, POS and DR tags respectively), values from 4 to 6 refer to heterogeneous sn-grams with words as head elements, values from 7 to 9 refer to sn-grams with lemmas as head

Algorithm 5 Function *COMPOUND_SNGRAMS*

Parameters: *container* (first set of subtrees), *sentence* (syntactic information matrix), *min_size* (minimum size), *max_size* (maximum size).
Output: New set of subtrees.

```
function COMPOUND_SNGRAMS(container,sentence, min_size, max_size)
    Vars: $newsngrams[\,], combinations[\,], candidates[\,], value.$
    for all $item \in container$ do
        if $length(item) > 0$ then
            $combinations.add(item)$
        end if
        if $item$ does not contain $sentence.root\_idx$ then
            $candidates.add(item)$
        end if
    end for
    while $length(candidates) > 0$ do
        $candidate \leftarrow candidates.pop[0], value \leftarrow candidate.pop[0]$
        $value \leftarrow candidate.pop[0]$
        for all $combination \in combinations$ do
            if $value \in combination[1]$ then
                $position \leftarrow$ first occurrence of $value$ in $combination[0]$
                $sngram \leftarrow combination$
                $sngram.pop(position)$
                $sngram.add(position, candidate)$
                if $LEN\_SNGRAM(sngram) \in [min\_size, self.max\_size + 1]$ then
                    $newsgrams.add(sngram)$
                end if
                if $LEN\_SNGRAM(sngram) < max\_size$ then
                    if $sngram$ contains $sentence.root\_idx$ then
                        $combinations.add(sngram)$
                    else
                        $combinations.add(sngram)$
                        $candidates.add(sngram)$
                    end if
                end if
            end if
        end for
    end while
    return $newsngrams$
end function
```

elements, values from 10 to 12 refer to sn-grams with POS tags as head elements and values from 13 to 15 refer to sn-grams with DR tags as head elements.

PREPARE_SNGRAM is a recursive function that in each invocation translates an

Algorithm 6 Function *PREPARE_SNGRAM*

Parameters: *line* (sn-gram index codification), *sentence* (syntactic information matrix), *op* (type of n-gram).

Output: an sn-gram codified with the corresponding information (words, lemmas, POS and DR tags), either heterogeneous or homogeneous.

```
function PREPARE_SNGRAM(line, sentence, op)
    Vars: $ngram$
    $ngram \leftarrow$ “”
    for all $item \in line$ do
        if $data\_type(item)$ is $str$ then
            $ngram \leftarrow ngram + item$
        else if $data\_type(item)$ is $int$ then
            if $op == 0$ then
                $ngram \leftarrow ngram + sentence.word[item]$
            else if $op == 1$ then
                $ngram \leftarrow ngram + sentence.lemma[item]$
            else if $op == 2$ then
                $ngram \leftarrow ngram + sentence.pos[item]$
            else if $op == 3$ then
                $ngram \leftarrow ngram + sentence.rel[item]$
            else if $op == 4$ then
                $ngram \leftarrow ngram + sentence.word[item]$
                $op \leftarrow 1$
            else if $op == 5$ then
                $ngram \leftarrow ngram + sentence.word[item]$
                $op \leftarrow 2$
            else if $op == 6$ then
                $ngram \leftarrow ngram + sentence.word[item]$
                $op \leftarrow 3$
            else if $op == 7$ then
                $ngram \leftarrow ngram + sentence.lemma[item]$
                $op \leftarrow 0$
            else if $op == 8$ then
                $ngram \leftarrow ngram + sentence.lemma[item]$
                $op \leftarrow 2$
            else if $op == 9$ then
                $ngram \leftarrow ngram + sentence.lemma[item]$
                $op \leftarrow 3$
            else if $op == 10$ then
                $ngram \leftarrow ngram + sentence.pos[item]$
                $op \leftarrow 0$
            else if $op == 11$ then
                $ngram \leftarrow ngram + sentence.pos[item]$
                $op \leftarrow 1$
```

Algorithm 7 Function *PREPARE_SNGRAM* (cont.)

```
            else if $op == 12$ then
                $ngram \leftarrow ngram + sentence.pos[item]$
                $op \leftarrow 3$
            else if $op == 13$ then
                $ngram \leftarrow ngram + sentence.rel[item]$
                $op \leftarrow 0$
            else if $op == 14$ then
                $ngram \leftarrow ngram + sentence.rel[item]$
                $op \leftarrow 1$
            else if $op == 15$ then
                $ngram \leftarrow ngram + sentence.rel[item]$
                $op \leftarrow 2$
            end if
        else
            $ngram \leftarrow ngram + PREPARE\_SNGRAM(item, sentence, op)$
        end if
    end for
    return $ngram$
end function
```

index of a sn-gram element into a linguistic type of information. For the cases of heterogeneous sn-grams, the function changes the value of the parameter *op*, so the elements other than the head are codified with a different type of information.

Finally, Algorithm 8 contains the main function that performs the extraction of syntactic n-grams named *EXTRACT_SNGRAMS*. The parameters that the function receives are the adjacency table with the syntactic information of a sentence, the integer variable that indicates the type of sn-grams to be extracted (heterogeneous or homogeneous, and the syntactic information to be used), the minimum and maximum size of the subtrees. As output, the function returns the extracted syntactic n-grams.

Conclusions

In this paper we presented the detailed description of the algorithm for extracting complete syntactic n-grams (heterogeneous and homogeneous) from syntactic trees. Syntactic n-grams allow obtaining full description of the information expressed in the syntactic trees that correspond to the sentences of texts. They are suitable as feature representation for several NLP problems, because they explore directly the syntactic information and allow to introduce it into machine learning methods, for example, identify more accurate patterns of how a writer uses the language.

We also presented a current state-of-the-art on usage of syntactic n-grams as features for natural language processing problems. In future research, we are planning to evaluate the syntactic n-grams as features for other NLP tasks such as question answering and sentiment analysis. For the authorship attribution task, we are considering to complement the sn-grams with other features from the literature such as

Algorithm 8 Function EXTRACT_SNGRAMS

Parameters: *sentence* (matrix with the syntactic information), *min_size* (minimum size), *max_size* (maximum size), *op* (type of sn-gram).
Output: the list containing the extracted sn-grams.

```
function EXTRACT_SNGRAMS(sentence, min_size, max_size, op)
    Vars: $unigrams[\,], combinations[\,], aux[\,], sngrams[\,]$
    $unigrams, combinations \leftarrow GET\_SUBTREES(sentence, min\_size, max\_size)$
    if size of $(unigrams) > 0$ then
        $sngrams.add(sentence.root\_idx)$
        $sngrams.add(unigrams)$
    end if
    for $item$ in $combinations$ do
        if $self.min\_size <> 0$ OR $self.max\_size <> 0$ then
            $size \leftarrow LEN\_SNGRAM(PREPARE\_SNGRAM(item, op))$
            if $size >= min\_size$ and $size <= max\_size$ then
                $sngrams.add(item)$
            end if
            if $size < max\_size$ then
                $aux.add(item)$
            end if
        else:
            $sngrams.add(item)$
        end if
    end for
    if $min\_size <> 0$ OR $max\_size <> 0$ then
        $COMPOUND\_SNGRAMS(aux, sentence, min\_size, max\_size)$
    else
        $COMPOUND\_SNGRAMS(combinations, sentence, min\_size, max\_size)$
    end if
    return $sngrams$
end function
```

character n-grams, word n-grams, typed character n-grams in order to build a more accurate authorship attribution methodology.

With respect to the algorithm for the extraction of syntactic n-grams, we would like to implement different filter functions such as removing or keeping stop words in n-grams, n-grams of nouns, n-grams of verbs, etc. For example, with these functions, we will be able to extract syntactic n-grams only using stop words or nouns. We believe that syntactic n-grams of stop words will be an efficient feature set for the authorship identification problem given that it has been shown before that stop words play a crucial role in this task [30].

Acknowledgments

This work was partially supported by the Mexican Government and Instituto Politécnico Nacional (CONACYT project 240844, SNI, COFAA-IPN, SIP-IPN projects 20151406, 20161947, 20161958, 20151589, 20162204, 20162064).

References

[1] J. Diederich, J. Kindermann, E. Leopold, and G. Paass, "Authorship attribution with support vector machines," *Applied intelligence*, vol. 19, no. 1, pp. 109–123, 2003.

[2] J. Posadas-Durán, H. Gómez-Adorno, I. Markov, G. Sidorov, I. Batyrshin, A. Gelbukh, and O. Pichardo-Lagunas, "Syntactic n-grams as features for the author profiling task," in *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum*, 2015.

[3] I. Markov, H. Gómez-Adorno, G. Sidorov, and A. Gelbukh, "Adapting cross-genre author profiling to language and corpus," in *Proceedings of the CLEF*, pp. 947–955, 2016.

[4] I. Markov, H. Gómez-Adorno, J.-P. Posadas-Durán, G. Sidorov, and A. Gelbukh, "Author profiling with doc2vec neural networkbased document embeddings," in *Proceedings of the 15th Mexican International Conference on Artificial Intelligence (MICAI 2016). Lecture Notes in Artificial Intelligence*, In press.

[5] J.-P. Posadas-Durán, G. Sidorov, I. Batyrshin, and E. Mirasol-Meléndez, "Author verification using syntactic n-grams," in *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum*, 2015.

[6] E. Stamatatos, M. Tschuggnall, B. Verhoeven, W. Daelemans, G. Specht, B. Stein, and M. Potthast, "Clustering by authorship within and across documents," in *Working Notes Papers of the CLEF*, 2016.

[7] M. A. Sanchez-Perez, G. Sidorov, and A. Gelbukh, "The winning approach to text alignment for text reuse detection at PAN 2014," in *Working Notes for CLEF 2014 Conference*, pp. 1004–1011, 2014.

[8] M. A. Sanchez-Perez, A. F. Gelbukh, and G. Sidorov, "Adaptive algorithm for plagiarism detection: The best-performing approach at PAN 2014 text alignment competition," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 6th International Conference of the CLEF Association*, pp. 402–413, 2015.

[9] M. A. Sánchez-Pérez, A. F. Gelbukh, and G. Sidorov, "Dynamically adjustable approach through obfuscation type recognition," in *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015.*, 2015.

[10] G. Sidorov, F. Velasquez, E. Stamatatos, A. Gelbukh, and L. Chanona-Hernández, “Syntactic n-grams as machine learning features for natural language processing,” *Expert Systems with Applications*, vol. 41, no. 3, pp. 853–860, 2013.

[11] G. Sidorov, “Non-continuous syntactic n-grams,” *Polibits*, vol. 48, no. 1, pp. 67–75, 2013.

[12] S. Galicia-Haro and A. Gelbukh, *Investigaciones en Anlisis Sintctico para el español*. Instituto Politcnico Nacional, 2007.

[13] H. Beristáin and H. Beristáin, *Gramática estructural de la lengua española*. Universidad Nacional de México, 2001.

[14] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, “The Stanford CoreNLP natural language processing toolkit,” in *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55–60, 2014.

[15] M.-C. De Marneffe and C. D. Manning, “Stanford typed dependencies manual,” tech. rep., Technical report, Stanford University, 2008.

[16] B. Santorini, *Part-of-speech tagging guidelines for the Penn Treebank Project (3rd revision)*. 1990.

[17] J.-P. Posadas-Duran, G. Sidorov, and I. Batyrshin, “Complete syntactic n-grams as style markers for authorship attribution,” in *LNAI*, vol. 8856, pp. 9–17, Springer, 2014.

[18] E. Stamatatos, “A survey of modern authorship attribution methods,” *Journal of the American Society for Information Science and Technology*, vol. 60, no. 3, pp. 538–556, 2009.

[19] P. Juola, “Future trends in authorship attribution,” in *Advances in Digital Forensics III* (P. Craiger and S. Shenoi, eds.), vol. 242 of *IFIP International Federation for Information Processing*, pp. 119–132, Springer Boston, 2007.

[20] E. Stamatatos, “A survey of modern authorship attribution methods,” *Journal of the American Society for information Science and Technology*, vol. 60, no. 3, pp. 538–556, 2009.

[21] G. Sidorov, F. Velasquez, E. Stamatatos, A. Gelbukh, and L. Chanona-Hernández, “Syntactic dependency-based n-grams as classification features,” in *Mexican International Conference on Artificial Intelligence MICAI 2012*, pp. 1–11, Springer, 2012.

[22] G. Sidorov, “Syntactic dependency based n-grams in rule based automatic english as second language grammar correction,” *International Journal of Computational Linguistics and Applications*, vol. 4, no. 2, pp. 169–188, 2013.

[23] S. D. Hernandez and H. Calvo, “Conll 2014 shared task: Grammatical error correction with a syntactic n-gram language model from a big corpora.,” in *CoNLL Shared Task*, pp. 53–59, 2014.

[24] F. Rangel, F. Celli, P. Rosso, M. Potthast, B. Stein, and W. Daelemans, "Overview of the 3^{rd} author profiling task at PAN 2015," in *CLEF 2015 Labs and Workshops, Notebook Papers* (L. Cappelato, N. Ferro, G. Jones, and E. S. Juan, eds.), vol. 1391, CEUR, 2015.

[25] G. Sidorov, A. Gelbukh, H. Gómez-Adorno, and D. Pinto, "Soft similarity and soft cosine measure: Similarity of features in vector space model," *Computación y Sistemas*, vol. 18, no. 3, pp. 491–504, 2014.

[26] H. Calvo, A. Segura-Olivares, and A. García, "Dependency vs. constituent based syntactic n-grams in text similarity measures for paraphrase recognition," *Computación y Sistemas*, vol. 18, no. 3, pp. 517–554, 2014.

[27] V. Laippala, J. Kanerva, and F. Ginter, "Syntactic ngrams as keystructures reflecting typical syntactic patterns of corpora in finnish," *Procedia-Social and Behavioral Sciences*, vol. 198, pp. 233–241, 2015.

[28] R. Sennrich, "Modelling and optimizing on syntactic n-grams for statistical machine translation," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 169–182, 2015.

[29] B. Agarwal and N. Mittal, *Prominent Feature Extraction for Sentiment Analysis*. Springer, 2016.

[30] E. Stamatatos, "Plagiarism detection using stopword n-grams," *Journal of the American Society for Information Science and Technology*, vol. 62, no. 12, pp. 2512–2527, 2011.

Automatic Generation of Summary Obfuscation Corpus for Plagiarism Detection

Sabino Miranda-Jiménez[1], Efstathios Stamatatos[2]

[1]CONACYT / INFOTEC– Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación, Circuito Tecnopolo Sur 112, 20313, Aguascalientes, México, sabino.miranda@infotec.mx

[2]Department of Information and Communication Systems Engineering, University of the Aegean, Karlovasi, 83200, Samos, Greece, stamatatos@aegean.gr

Abstract: In this paper, we describe an approach to create a summary obfuscation corpus for the task of plagiarism detection. Our method is based on information from the Document Understanding Conferences related to years 2001 and 2006, for the English language. Overall, an unattributed summary used within someone else's document is considered a kind of plagiarism because the main author's ideas are still in a succinct form. In order to create the corpus, we use a Named Entity Recognizer (NER) to identify the entities within an original document, its associated summaries, and target documents. After, these entities, together with similar paragraphs in target documents, are used to make fake suspicious documents and plagiarized documents. The corpus was tested in plagiarism competition.

Keywords: corpus generation; plagiarism detection; obfuscation strategies

1 Introduction

The gigantic number of digital documents produced every day and the information available online, have made it easy to reuse data (sentences, excerpts, etc.) from others' work into one's own documents without citing the corresponding source of information; thus, plagiarism comes into the picture. Plagiarism is the reuse of someone else's ideas, processes, results or words without explicitly acknowledging the author's work and source [1].

In recent years, plagiarism detection has received much attention from the community in terms of published papers and systems developed, for example, PAN contests, in plagiarism detection task [2, 3]. In order to evaluate the systems developed, it is required corpus designed for this purpose. Traditionally, an intrinsic evaluation is conducted to evaluate the performance of systems [4, 5], i.e., given a set of suspicious documents, a system must determine whether a whole document or sections of the document are plagiarized from other sources.

Thus, different corpora have been developed using, for example, obfuscation strategies such as author obfuscation [6], which consists in distorting the most frequent words for an author, replacing each word with one of their synonyms. Also, the use of paraphrasing was proposed in [7, 8], for example, using Wikipedia articles to creating the corpus of original and suspicious documents (fragments); in order to obfuscate fragments, it uses random strategies to shuffle words in the extracted fragments, in addition, Part-of-Speech features are used to preserve syntactic structure in fragments. A more sophisticated strategy uses SemEval dataset of semantic textual similarity [9], here, a pair of semantically similar sentences are used to create simulated plagiarism cases; both source and plagiarized fragments are constructed by SemEval dataset sentences.

In this work, we argue, in detail, the development of a corpus for plagiarism intrinsic evaluation used in evaluating systems of plagiarism detection [2]. The creation of the corpus is based on information (news dataset) from text summarization field. This dataset is usually used to evaluate the performance of summarization systems [10], i.e., the summary generated by systems is compared against abstractive/extractive summaries created manually by human experts. We chose abstractive summaries (no simple concatenation of sentences) related to this dataset because it could be considered as plagiarism of author's ideas. Our strategies are based on entities mentioned within news documents and theirs associated summaries and its similarity among target (suspicious) paragraphs/documents in order to mask the information.

In the following sections, we describe our approach in detail, in Section 2, the selection of documents from datasets used, and how the obfuscation strategies are applied. The resulting corpus and the performance of systems on our approach are presented in Section 3. Conclusions and future work are discussed in Section 4.

2 Obfuscation Approach

We propose a kind of plagiarism based on information comes from news and theirs associated summaries. In general, we consider that an unattributed summary used within a document of someone else is a kind of plagiarism, because the ideas of the author are still in a condensed form. For this work, we use summaries made by human experts who paraphrased the original news given in the dataset.

Roughly speaking, our method uses two news datasets: one for extracting summaries, and the other one for making the related suspicious documents. In the following subsections, we describe the creation of the corpus.

2.1 DUC Datasets

As we mentioned, the creation of the corpus is based on two news datasets from Document Understanding Conferences (DUC) in 2001 [10] and 2006 [11]. DUC competition provides these datasets for evaluating the performance of automatic text summarization systems; the datasets is for the English language. Generally, contests for evaluating the performance of systems reused their dataset for several years because of the cost of manual generation of datasets. DUC competition used essentially the same source of documents for DUC-2001 and DUC-2002, and another source of documents from DUC-2003 to DUC-2006.

In order not to choose the same documents (news) that we want to plagiarize and fake, we selected the datasets of DUC-2001 and DUC-2006 as our source of documents. The datasets are described as follows.

DUC-2001 news dataset comprises documents related to Wall Street Journal of years 1987-1992, AP newswire (1989-1990), San Jose Mercury News (1991), Financial Times (1991-1994), LA Times, and Foreign Broadcast Information Service (FBIS). For this dataset, four generic summaries for each document (news) with length of approximately 50, 100, 200, and 400 words were created manually by human assessors. The summary is considered as an abstractive document since no simple concatenation was implemented, i.e., the main ideas of the author, in a condensed form, are within the summary.

DUC-2006 news dataset comprises documents related to Associated Press newswire (1998-2000), New York Times newswire (1998-2000), and Xinhua News Agency (English version, 1996-2000).

Both datasets have similar topics but do not deal with the same news. Table 1 shows the number of documents selected as starting dataset. The news considered as original documents, was selected with at least a 400 word length. The associated summaries were summaries of a 100 word length, because this document length has enough information to deal with faking suspicious paragraphs. The target news was considered for documents with at least a 600 word length.

Table 1

Starting dataset for summary plagiarism task

Source	No. documents
DUC-2001: Original news	237
DUC-2001: Summaries per each news	2
DUC-2006: Target news	527

2.2 Similarity Measure

One of our strategies is using a similarity measure to identify similar paragraphs and similar documents. Therefore, we use an easy measure to calculate the similarity of two objects. In particular, we used the well-known measure, the Dice similarity or coefficient, that is simple but it has a good quality [12]. The Dice similarity is defined as follows, in equation 1.

$$Sim_{Dice}(X,Y)=2\frac{X\cap Y}{|X|+|Y|} \tag{1}$$

In equation 1, let be X and Y documents, the similarity is between 0 and 1; 0 means no similarity, and 1 stands for maximum similarity. |X| means the cardinality of the document, and X represents the set of words of the document. For example, given two documents, X and Y, defined below, the Dice similarity for these documents is 0.1360; it means that 13.6% of X document is similar to Y document. In order to calculate the Dice similarity, we only used content words as elements to be compared, i.e., we discarded punctuation marks and words such as *a*, *to*, *on*, *or*, etc., known as Stop words. Stop words are considered that do not contribute to defining the content of the document [13].

X: *The British cattle industry was under siege, while many nations of the European Union were imposing or discussing bans on imports, fearing "mad cow" could be transmitted to humans.*

Y: *The controversial practice of feeding ground animal remains to pigs and poultry is to be outlawed across the European Union from January as part of a continent-wide effort to stamp out a rising wave of consumer panic over Mad Cow disease.*

2.3 Named Entity Recognition

Another key strategy is the entities mentioned in documents; entities identified are used by the main method. In news genre, entities are common, because data are facts about events, places, persons, dates, organizations, etc. In this genre, it could be easy to identify original documents and theirs associated summaries considering the occurrences of entities. Thus, in order to obfuscate the information using entities, we use a tool to extract them from texts.

The Stanford Named Entity Recognizer (NER) [14] is used to identify seven categories (entities) in documents: time, location, organization, person, money, percent, and date. The entity information and similarity measure are used to create fake documents, as well as fake paragraphs related to entities identified in documents. Figure 1 shows how the entities are identified, defined between XML-based tags, for example, **<LOCATION>** and **</LOCATION>**; and

<ORGANIZATION> and **</ORGANIZATION>**. The information between these tags is the entities to be distorted in target documents.

Investigators from the **<LOCATION>**United States**</LOCATION>** and <LOCATION>Egypt</LOCATION> will review part of the flight control system in the tail of **<ORGANIZATION>**Boeing**</ORGANIZATION>**'s 767 airplane as part of the investigation into the crash of <ORGANIZATION> EgyptAir</ORGANIZATION> Flight 990, the chairman of the <ORGANIZATION>National Transportation Safety Board</ORGANIZATION> said <DATE>Friday</DATE>. The disclosure comes just a couple days after the chairman of <ORGANIZATION>EgyptAir</ORGANIZATION> told a news conference in <LOCATION>Cairo</LOCATION> that something happened to the tail of the <ORGANIZATION>Boeing</ORGANIZATION> 767 that caused it to go into a near supersonic dive before the plane broke up and crashed into the sea. Safety board chairman <PERSON>Jim Hall</PERSON> said investigators from his agency and the <ORGANIZATION>Egyptian Civil Aviation Authority</ORGANIZATION> will examine the 767's elevator system, as well as perform a metallurgic examination of the plane's engine pylon components.

Figure 1

Entities identified by NER

2.4 Obfuscation Method

The creation of the summary obfuscation corpus is based on documents of two datasets of DUC competition. DUC-2001 dataset serves as original documents, i.e., these documents are the information to be plagiarized, and DUC-2006 dataset serves as target documents, these documents serve for two goals: the first one is to create plagiarized documents and the second one is to create suspicious documents, i.e., fake plagiarized documents, using the named entities and close documents related to the original documents according to their similarity.

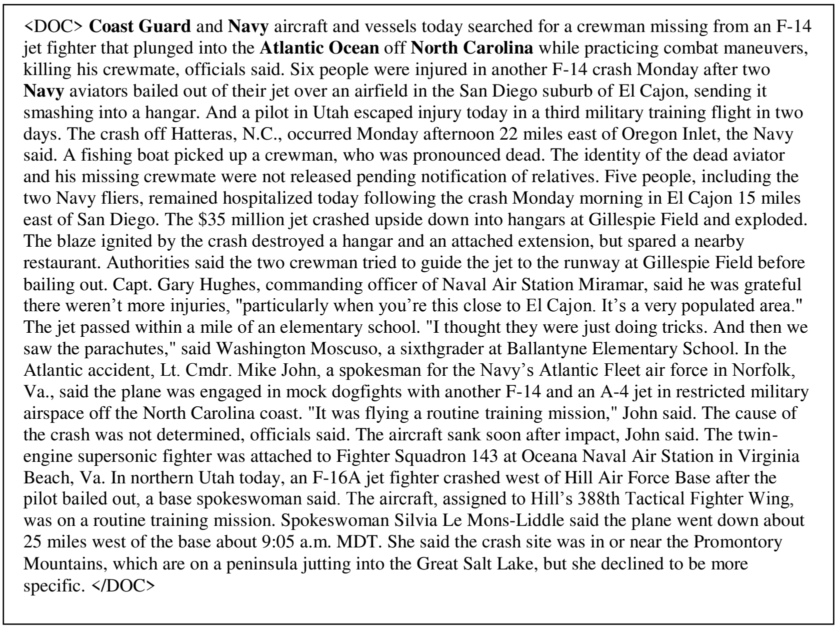
In order to achieve the goals, our method includes three main stages: preprocessing of DUC datasets, candidate document selection, and data obfuscation.

2.4.1 Preprocessing of DUC Datasets

The first stage is selecting the documents from DUC datasets; initially, this information is used to measure the performance of text summarization systems. Thus, there are source documents and four or five associated summaries manually created by human experts.

As we mentioned, on one hand, the original documents were selected from DUC-2001 dataset. Each document was selected based on its length, the document size is greater than 400 words, and two associated summaries of 100 words were selected (see section 2.1); this set of documents we will refer to as *original documents*. Figure 2 illustrates an example of an original document and Figure 3 shows its associated summaries. We have two summaries for each original document.

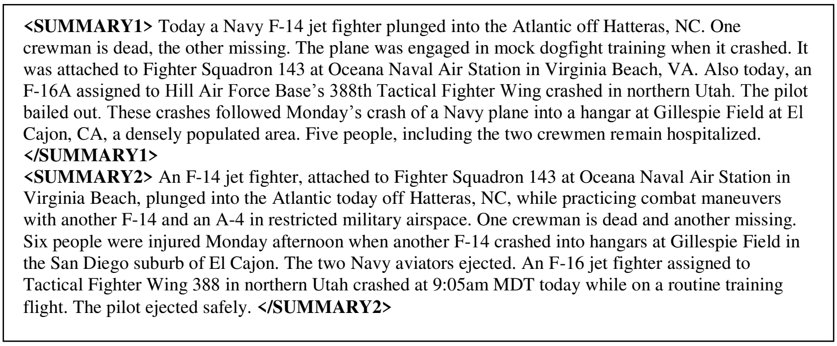
On the other hand, the documents from DUC-2006 play the role of suspicious documents. Similarly, the documents selected were based on the length; the document size is greater than 600 words in order to have enough text to add fake paragraphs or plagiarized paragraphs; this set of documents we will refer to as *candidate documents*. In both cases, all HTML tags were removed and only the text body is used.



<DOC> **Coast Guard** and **Navy** aircraft and vessels today searched for a crewman missing from an F-14 jet fighter that plunged into the **Atlantic Ocean** off **North Carolina** while practicing combat maneuvers, killing his crewmate, officials said. Six people were injured in another F-14 crash Monday after two **Navy** aviators bailed out of their jet over an airfield in the San Diego suburb of El Cajon, sending it smashing into a hangar. And a pilot in Utah escaped injury today in a third military training flight in two days. The crash off Hatteras, N.C., occurred Monday afternoon 22 miles east of Oregon Inlet, the Navy said. A fishing boat picked up a crewman, who was pronounced dead. The identity of the dead aviator and his missing crewmate were not released pending notification of relatives. Five people, including the two Navy fliers, remained hospitalized today following the crash Monday morning in El Cajon 15 miles east of San Diego. The $35 million jet crashed upside down into hangars at Gillespie Field and exploded. The blaze ignited by the crash destroyed a hangar and an attached extension, but spared a nearby restaurant. Authorities said the two crewman tried to guide the jet to the runway at Gillespie Field before bailing out. Capt. Gary Hughes, commanding officer of Naval Air Station Miramar, said he was grateful there weren't more injuries, "particularly when you're this close to El Cajon. It's a very populated area." The jet passed within a mile of an elementary school. "I thought they were just doing tricks. And then we saw the parachutes," said Washington Moscuso, a sixthgrader at Ballantyne Elementary School. In the Atlantic accident, Lt. Cmdr. Mike John, a spokesman for the Navy's Atlantic Fleet air force in Norfolk, Va., said the plane was engaged in mock dogfights with another F-14 and an A-4 jet in restricted military airspace off the North Carolina coast. "It was flying a routine training mission," John said. The cause of the crash was not determined, officials said. The aircraft sank soon after impact, John said. The twin-engine supersonic fighter was attached to Fighter Squadron 143 at Oceana Naval Air Station in Virginia Beach, Va. In northern Utah today, an F-16A jet fighter crashed west of Hill Air Force Base after the pilot bailed out, a base spokeswoman said. The aircraft, assigned to Hill's 388th Tactical Fighter Wing, was on a routine training mission. Spokeswoman Silvia Le Mons-Liddle said the plane went down about 25 miles west of the base about 9:05 a.m. MDT. She said the crash site was in or near the Promontory Mountains, which are on a peninsula jutting into the Great Salt Lake, but she declined to be more specific. </DOC>

Figure 2

Example of an original document



<SUMMARY1> Today a Navy F-14 jet fighter plunged into the Atlantic off Hatteras, NC. One crewman is dead, the other missing. The plane was engaged in mock dogfight training when it crashed. It was attached to Fighter Squadron 143 at Oceana Naval Air Station in Virginia Beach, VA. Also today, an F-16A assigned to Hill Air Force Base's 388th Tactical Fighter Wing crashed in northern Utah. The pilot bailed out. These crashes followed Monday's crash of a Navy plane into a hangar at Gillespie Field at El Cajon, CA, a densely populated area. Five people, including the two crewmen remain hospitalized. **</SUMMARY1>**

<SUMMARY2> An F-14 jet fighter, attached to Fighter Squadron 143 at Oceana Naval Air Station in Virginia Beach, plunged into the Atlantic today off Hatteras, NC, while practicing combat maneuvers with another F-14 and an A-4 in restricted military airspace. One crewman is dead and another missing. Six people were injured Monday afternoon when another F-14 crashed into hangars at Gillespie Field in the San Diego suburb of El Cajon. The two Navy aviators ejected. An F-16 jet fighter assigned to Tactical Fighter Wing 388 in northern Utah crashed at 9:05am MDT today while on a routine training flight. The pilot ejected safely. **</SUMMARY2>**

Figure 3

Example of two associated summaries

2.4.2 Candidate Document Selection

The second stage consists in selecting the best document group from the candidate documents for each original document. In order to achieve this goal, we follow the next steps. First, for an original document is calculated the similarity with all documents in the candidate dataset. In order to do this, we used Dice similarity (equation 1) to identify the probable candidates to be obfuscated. The minimum threshold for similarity is 10 percent of the original document in order to guarantee at least a degree of similarity. Second, all nominee documents are ranked according to Dice similarity in descending order. After that, the top 10 documents are selected, as documents to be obfuscated (target documents). In addition, a nominee document could not be used more than ten times in order to give other documents the chance to be chosen. Two of the target documents are used for plagiarism and the remaining ones are for creating suspicious documents.

2.4.3 Data Obfuscation

The third stage consists in obfuscating the information of the target documents selected in the previous stage (Sec. 2.4.2). To achieve this goal, there are three steps: entity extraction, similar paragraph identification, and fake and plagiarized text insertion.

First, Stanford NER is applied to extract entities for each original document and its associated summaries, as well as the entities for each related target document. Second, paragraphs of target documents are selected according to most similar content by Dice similarity, and similar dispersion of entity types between the original document and the target document. Third, in order to insert the plagiarized summary, a random selection of the place in the target document is performed among the selected paragraphs in the previous step. In addition, two paragraphs are selected to be noisy areas, i.e., replacing the entities from the candidate paragraph with entities of the most similar paragraph from the original document, according to the content and the entity dispersion. The function of noisy areas is to mislead potential methods that take entities to identify plagiarism.

The entities extracted are used as round-robin approach, that is, a circular list, in order to continue replacing entities in the target paragraph until entities are exhausted. Figure 7 shows an example of plagiarized document with noisy areas. In the case of the generation of suspicious documents the entity identification and the replacement of named entities are applied in the same way. Fake, suspicious documents have a similar structure to plagiarized documents, without the plagiarized section. We can see in Figure 6, the original text, and we can see, in Figure 7, how the entities were replaced in the noisy area (entities are in bold), and the text is still readable, but it is obfuscated.

3 Results

The statistics of the resulting corpus are shown in Table 2. The corpus has 2370 documents. The corpus consists of two plagiarized documents and eight fake suspicious documents per each original document (237 documents). There are 496 plagiarized documents. A plagiarized document consists of the text, two noisy areas, and a plagiarized text (a summary), see Figure 7. The XML-based tags are only for informative purposes. Also, there are 1896 fake suspicious documents. A fake document consists of the text and two noisy areas, similar to the structure of Figure 7, without plagiarized section.

Table 2

Statistics of Obfuscation Summary Corpus

Source	No. documents
Original documents	237
Fake suspicious documents	1896
Plagiarized documents	474

Figure 4 and Figure 5 show the structure of annotations for fake suspicious documents and plagiarized documents respectively. The annotated documents are to identify what documents are plagiarized and what documents are only suspicious. In the case of a plagiarized document, there are key features such as the feature called *name* with value *plagiarism* that indicates that the current document has plagiarism; *source_offset* indicates the place where the plagiarism starts; *source_length* is the total of plagiarized characters; and *source_reference* indicates the file name. In the case of fake suspicious documents, this information is absent, see Figure 4. Note that XML-based tags in original documents, fake suspicious documents and plagiarized documents are only for informative purposes; these tags are not present in the final documents.

```
<document reference="suspicious-document00790.txt">
<feature name="about" authors="DUC2006" title="news" language="en" />
<feature name="md5" value="250487dd32850d9b89e1b094392609dc" language="en" />
</document>
```

Figure 4

Annotations for fake suspicious documents

```
<document reference="suspicious-document0010.txt">
<feature name="about" authors="DUC2001, DUC2006" title="news" language="en" />
<feature name="md5" value="9f6e52a04880aac50e92a3d300356a27" language="en" />
<feature name="plagiarism" type="artificial" obfuscation="high" this_language="en"
this_offset="3210" this_length="632" source_reference="source-document0001.txt"
source_language="en" source_offset="1" source_length="7224" />
```

Figure 5

Annotations for plagiarized documents

3.1 Evaluation of Systems

The corpus with the approach described in this paper was used in PAN competition for text alignment for plagiarism detection [2]. The task on this corpus is to determine whether the document contains plagiarized sections given a set of suspicious documents.

Table 3 shows the performance of systems using the summary obfuscation corpus and other two strategies used in the competition: Random obfuscation and Cyclic translation obfuscation, for more details of the implementation of the strategies see [2]. The performance of the systems was measured by *PlagDet* score. Basically, PlagDet is a measure that considers F1 score (harmonic mean of precision and recall) and a kind of normalization considering detections of passage cases with plagiarism and passages confirmed with plagiarism, this measure was designed for this purpose in PAN competitions, for more details of this measure see [2].

According to the performance of the systems with these datasets, it is hard for the systems to identify correctly the plagiarized documents as we can see in the low values obtained with our corpus (Summary Obfuscation) for all systems.

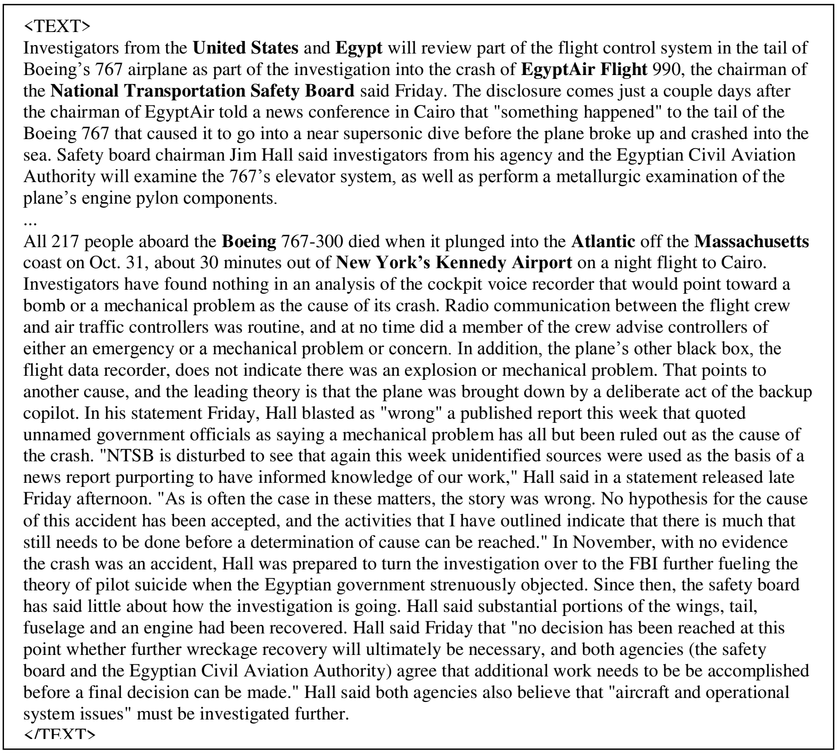
Table 3

Evaluation of text alignment systems related to plagiarism detection

Team	Random	Cyclic translation	Summary Obfuscation
Suchomel [16]	0.75276	0.67544	**0.61011**
Kong [17]	0.83242	0.85212	**0.43399**
R. Torrejón [18]	0.74711	0.85113	**0.34131**
Saremi [19]	0.65668	0.70903	**0.11116**
Shrestha [20]	0.66714	0.62719	**0.11860**
Gillam [21]	0.0419	0.01224	**0.00218**
Jayapal [22]	0.18148	0.18181	**0.05940**

In general, the performance for participating systems in plagiarism detection was weak. One system was a little higher than 60%, the remaining systems were below 45%. We notice intuitively, that the low performance of the systems is due to the strategies implemented and are not trivial, i.e., the plagiarized text is an abstractive summary made manually by human experts. An abstractive summary is not a simple concatenation of sentences or excerpts from an original document; often, it is a complete paraphrasing of the text using several operations to abstract the text [15]. According to the results of the performance of systems, this approach presents great challenges to systems, when the text is a sort of plagiarized version of an author's ideas.

Figure 6



<TEXT>
Investigators from the **United States** and **Egypt** will review part of the flight control system in the tail of Boeing's 767 airplane as part of the investigation into the crash of **EgyptAir Flight** 990, the chairman of the **National Transportation Safety Board** said Friday. The disclosure comes just a couple days after the chairman of EgyptAir told a news conference in Cairo that "something happened" to the tail of the Boeing 767 that caused it to go into a near supersonic dive before the plane broke up and crashed into the sea. Safety board chairman Jim Hall said investigators from his agency and the Egyptian Civil Aviation Authority will examine the 767's elevator system, as well as perform a metallurgic examination of the plane's engine pylon components.

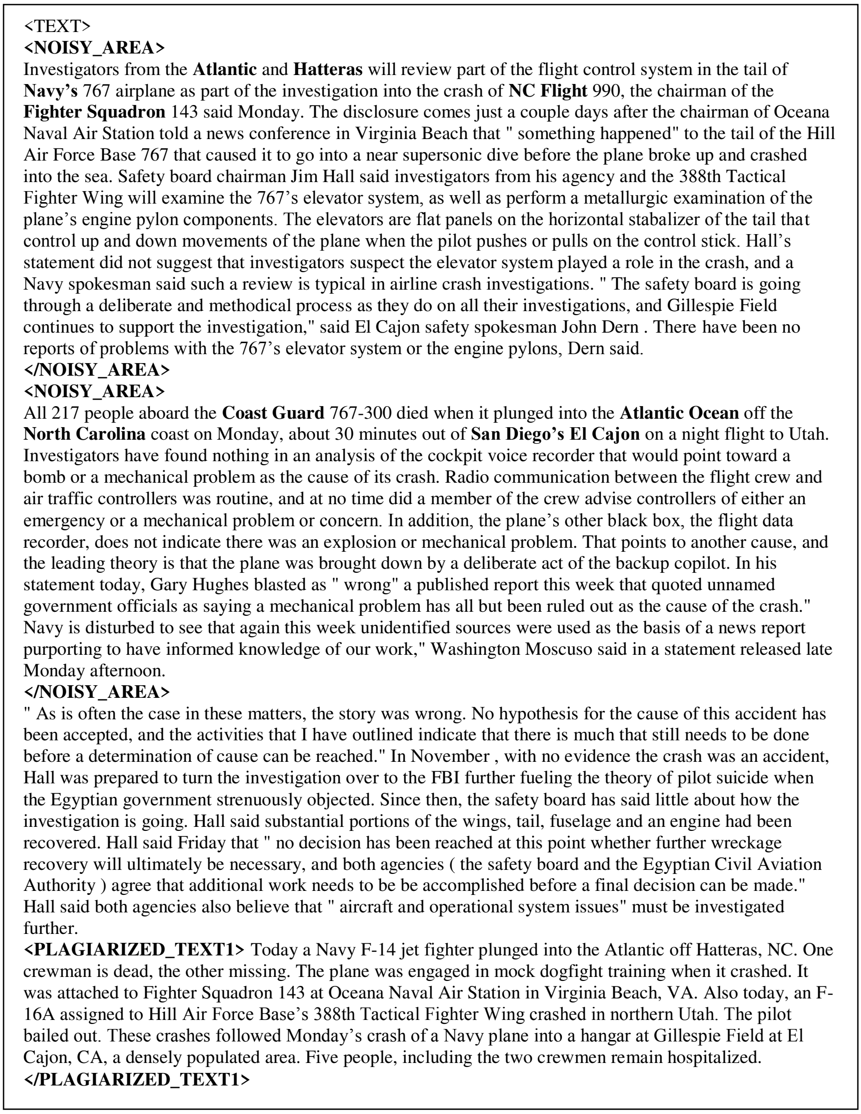
...

All 217 people aboard the **Boeing** 767-300 died when it plunged into the **Atlantic** off the **Massachusetts** coast on Oct. 31, about 30 minutes out of **New York's Kennedy Airport** on a night flight to Cairo. Investigators have found nothing in an analysis of the cockpit voice recorder that would point toward a bomb or a mechanical problem as the cause of its crash. Radio communication between the flight crew and air traffic controllers was routine, and at no time did a member of the crew advise controllers of either an emergency or a mechanical problem or concern. In addition, the plane's other black box, the flight data recorder, does not indicate there was an explosion or mechanical problem. That points to another cause, and the leading theory is that the plane was brought down by a deliberate act of the backup copilot. In his statement Friday, Hall blasted as "wrong" a published report this week that quoted unnamed government officials as saying a mechanical problem has all but been ruled out as the cause of the crash. "NTSB is disturbed to see that again this week unidentified sources were used as the basis of a news report purporting to have informed knowledge of our work," Hall said in a statement released late Friday afternoon. "As is often the case in these matters, the story was wrong. No hypothesis for the cause of this accident has been accepted, and the activities that I have outlined indicate that there is much that still needs to be done before a determination of cause can be reached." In November, with no evidence the crash was an accident, Hall was prepared to turn the investigation over to the FBI further fueling the theory of pilot suicide when the Egyptian government strenuously objected. Since then, the safety board has said little about how the investigation is going. Hall said substantial portions of the wings, tail, fuselage and an engine had been recovered. Hall said Friday that "no decision has been reached at this point whether further wreckage recovery will ultimately be necessary, and both agencies (the safety board and the Egyptian Civil Aviation Authority) agree that additional work needs to be be accomplished before a final decision can be made." Hall said both agencies also believe that "aircraft and operational system issues" must be investigated further.
</TEXT>

Example of a candidate document

Conclusions

We have described an approach for generating a summary obfuscation corpus to be used in plagiarism detection tasks. We used, as source of information, two datasets (DUC contest) from different years to avoid the same news. We considered the use of abstractive summaries within document as a case of plagiarism. We focused on, mainly, identifying entities and the dispersion of them through paragraphs, selecting similar documents and paragraphs to obfuscate the given summary into a plagiarized document; also twisting paragraphs replacing entities into noisy areas, for both plagiarized and fake documents. This approach was used in PAN competition for testing the performance of plagiarism detection systems.



<TEXT>
<NOISY_AREA>
Investigators from the **Atlantic** and **Hatteras** will review part of the flight control system in the tail of **Navy's** 767 airplane as part of the investigation into the crash of **NC Flight** 990, the chairman of the **Fighter Squadron** 143 said Monday. The disclosure comes just a couple days after the chairman of Oceana Naval Air Station told a news conference in Virginia Beach that " something happened" to the tail of the Hill Air Force Base 767 that caused it to go into a near supersonic dive before the plane broke up and crashed into the sea. Safety board chairman Jim Hall said investigators from his agency and the 388th Tactical Fighter Wing will examine the 767's elevator system, as well as perform a metallurgic examination of the plane's engine pylon components. The elevators are flat panels on the horizontal stabalizer of the tail that control up and down movements of the plane when the pilot pushes or pulls on the control stick. Hall's statement did not suggest that investigators suspect the elevator system played a role in the crash, and a Navy spokesman said such a review is typical in airline crash investigations. " The safety board is going through a deliberate and methodical process as they do on all their investigations, and Gillespie Field continues to support the investigation," said El Cajon safety spokesman John Dern . There have been no reports of problems with the 767's elevator system or the engine pylons, Dern said.
</NOISY_AREA>
<NOISY_AREA>
All 217 people aboard the **Coast Guard** 767-300 died when it plunged into the **Atlantic Ocean** off the **North Carolina** coast on Monday, about 30 minutes out of **San Diego's El Cajon** on a night flight to Utah. Investigators have found nothing in an analysis of the cockpit voice recorder that would point toward a bomb or a mechanical problem as the cause of its crash. Radio communication between the flight crew and air traffic controllers was routine, and at no time did a member of the crew advise controllers of either an emergency or a mechanical problem or concern. In addition, the plane's other black box, the flight data recorder, does not indicate there was an explosion or mechanical problem. That points to another cause, and the leading theory is that the plane was brought down by a deliberate act of the backup copilot. In his statement today, Gary Hughes blasted as " wrong" a published report this week that quoted unnamed government officials as saying a mechanical problem has all but been ruled out as the cause of the crash." Navy is disturbed to see that again this week unidentified sources were used as the basis of a news report purporting to have informed knowledge of our work," Washington Moscuso said in a statement released late Monday afternoon.
</NOISY_AREA>
" As is often the case in these matters, the story was wrong. No hypothesis for the cause of this accident has been accepted, and the activities that I have outlined indicate that there is much that still needs to be done before a determination of cause can be reached." In November , with no evidence the crash was an accident, Hall was prepared to turn the investigation over to the FBI further fueling the theory of pilot suicide when the Egyptian government strenuously objected. Since then, the safety board has said little about how the investigation is going. Hall said substantial portions of the wings, tail, fuselage and an engine had been recovered. Hall said Friday that " no decision has been reached at this point whether further wreckage recovery will ultimately be necessary, and both agencies (the safety board and the Egyptian Civil Aviation Authority) agree that additional work needs to be be accomplished before a final decision can be made." Hall said both agencies also believe that " aircraft and operational system issues" must be investigated further.
<PLAGIARIZED_TEXT1> Today a Navy F-14 jet fighter plunged into the Atlantic off Hatteras, NC. One crewman is dead, the other missing. The plane was engaged in mock dogfight training when it crashed. It was attached to Fighter Squadron 143 at Oceana Naval Air Station in Virginia Beach, VA. Also today, an F-16A assigned to Hill Air Force Base's 388th Tactical Fighter Wing crashed in northern Utah. The pilot bailed out. These crashes followed Monday's crash of a Navy plane into a hangar at Gillespie Field at El Cajon, CA, a densely populated area. Five people, including the two crewmen remain hospitalized.
</PLAGIARIZED_TEXT1>

Figure 7

Example of plagiarized document with two noisy areas

The results of the performance of systems showed that paraphrasing in a succinct way becomes great challenges to identify plagiarism.

As future work, we plan to apply our approach to multi-lingual and multi-document news, in the context of MultiLing competition [23, 24, 25]. In those datasets, there are same summaries in several languages such as Arabic, English, Greek, Hebrew and Spanish, but summaries are not a literal translation, they were created by native speakers from original documents in the English language. In this sense, we could work on a corpus for cross-lingual plagiarism detection, based on abstractive summaries, i.e., a summary in a language "A" could be obfuscated in a language "B", a language "C", etc., considering the summary in "B" and "C" as a plagiarism of the summary in "A".

References

[1] A. Barrón-Cedeño, M. Vila, M. A. Martí, and P. Rosso: Plagiarism Meets Paraphrasing: Insights for the Next Generation in Automatic Plagiarism Detection, Computational Linguistics, Vol. 39, No. 4, 2013, pp. 917-947

[2] M. Potthast, M. Hagen, T. Gollub, M. Tippmann, J. Kiesel, P. Rosso, E. Stamatatos, and B. Stein: Overview of the 5th International Competition on Plagiarism Detection, in CLEF Conference on Multilingual and Multimodal Information Access Evaluation, 2013, pp. 301-331

[3] P. Rosso, F. Rangel, M. Potthast, E. Stamatatos, M. Tschuggnall, and B. Stein: Overview of PAN'16—New Challenges for Authorship Analysis: Cross-genre Profiling, Clustering, Diarization, and Obfuscation, in Experimental IR Meets Multilinguality, Multimodality, and Interaction. 7th International Conference of the CLEF Initiative, N. Fuhr, P. Quaresma, B. Larsen, T. Gonçalves, K. Balog, C. Macdonald, L. Cappellato, and N. Ferro, Eds. Berlin Heidelberg New York: Springer, 2016, pp. 332-350

[4] M. Potthast, B. Stein, A. Barrón-Cedeño, and P. Rosso: An Evaluation Framework for Plagiarism Detection, in Proceedings of the 23rd International Conference on Computational Linguistics: Posters, ser. COLING '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 997-1005

[5] M. Kuznetsov, A. Motrenko, R. Kuznetsova, and V. Strijov: Methods for Intrinsic Plagiarism Detection and Author Diarization, in CLEF 2016 Evaluation Labs and Workshop – Working Notes Papers, 5-8 September, Évora, Portugal, K. Balog, L. Cappellato, N. Ferro, and C. Macdonald, Eds. Berlin Heidelberg New York: CEUR-WS.org, 2016, pp. 912-919

[6] M. Potthast, M. Hagen, and B. Stein: Author Obfuscation: Attacking the State of the Art in Authorship Verification, in Working Notes Papers of the CLEF 2016 Evaluation Labs, ser. CEUR Workshop Proceedings. CLEF and CEUR-WS.org, 2016, pp. 716-749

[7] M. Mansoorizadeh, T. Rahgooy, M. Aminiyan, and M. Eskandari: Author Obfuscation using WordNet and Language Models, in CLEF 2016 Evaluation Labs and Workshop – Working Notes Papers, 5-8 September, Évora, Portugal, K. Balog, L. Cappellato, N. Ferro, and C. Macdonald, Eds. Berlin Heidelberg New York: CEUR-WS.org, Sep. 2016, pp. 939-946

[8] S. Mohtaj, H. Asghari, and V. Zarrabi: Developing Monolingual English Corpus for Plagiarism Detection using Human Annotated Paraphrase Corpus, in CLEF 2015 Evaluation Labs and Workshop – Working Notes Papers, 8-11 September, Toulouse, France, L. Cappellato, N. Ferro, G. Jones, and E. San Juan, Eds. CEUR-WS.org, 2015

[9] E. Agirrea, C. Baneab, D. Cerd, M. Diabe, A. Gonzalez-Agirrea, R. Mihalceab, G. Rigaua, J. Wiebef, and B. C. Donostia: Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation, Proceedings of SemEval, 2016, pp. 497-511

[10] DUC-2001. (2001) The document understanding conference [Online] Available: http://duc.nist.gov/pubs.html#2001

[11] DUC-2006. (2006) The document understanding conference [Online] Available: http://duc.nist.gov/pubs.html#2006

[12] V. Thada and D. V. Jaglan: Comparison of Jaccard, Dice, Cosine Similarity Coefficient to Find Best Fitness Value for Web-retrieved Documents using Genetic Algorithm, International Journal of Innovations in Engineering and Technology (IJIET), Vol. 2, No. 4, 2013, pp. 202-205

[13] C. D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval. New York, NY, USA: Cambridge University Press, 2008

[14] J. R. Finkel, T. Grenager, and C. Manning: Incorporating Non-Local Information into Information Extraction Systems by Gibbs Sampling, in Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ser. ACL ’05. Stroudsburg, PA, USA: Association for Computational Linguistics, 2005, pp. 363-370

[15] S. Miranda-Jiménez, A. Gelbukh, and G. Sidorov: Conceptual Graphs as Framework for Summarizing Short Texts, International Journal of Conceptual Structures and Smart Applications (IJCSSA), Vol. 2, No. 2, 2014, pp. 55-75

[16] Š. Suchomel, J. Kasprzak, and M. Brandejs: Diverse Queries and Feature Type Selection for Plagiarism Discovery, in CLEF 2013 Evaluation Labs and Workshop – Working Notes Papers, 23-26 September, Valencia, Spain, P. Forner, R. Navigli, and D. Tufis, Eds. CELCT, 2013

[17] L. Kong, H. Qi, C. Du, M. Wang, and Z. Han: Approaches for Source Retrieval and Text Alignment of Plagiarism Detectio, in CLEF 2013 Evaluation Labs and Workshop – Working Notes Papers, 23-26 September, Valencia, Spain, P. Forner, R. Navigli, and D. Tufis, Eds. CELCT, 2013

[18] D. Rodríguez Torrejón and J. Martín Ramos: Text Alignment Module in CoReMo 2.1 Plagiarism Detector, in CLEF 2013 Evaluation Labs and Workshop – Working Notes Papers, 23-26 September, Valencia, Spain, P. Forner, R. Navigli, and D. Tufis, Eds. CELCT, 2013

[19] M. Saremi and F. Yaghmaee: Submission to the 5th International Competition on Plagiarism Detection, PAN/CLEF 2013, CELCT, Semnan University, Iran, 2013, pp. 352-365

[20] P. Shrestha and T. Solorio: Using a Variety of n-Grams for the Detection of Different Kinds of Plagiarism, in CLEF 2013 Evaluation Labs and Workshop – Working Notes Papers, 23-26 September, Valencia, Spain, P. Forner, R. Navigli, and D. Tufis, Eds. CELCT, 2013

[21] L. Gillam: Guess Again and See if They Line Up: Surrey's Runs at Plagiarism Detection, in CLEF 2013 Evaluation Labs and Workshop – Working Notes Papers, 23-26 September, Valencia, Spain, P. Forner, R. Navigli, and D. Tufis, Eds. CELCT, 2013

[22] A. Jayapal and B. Goswami: Submission to the 5th International Competition on Plagiarism Detection, PAN/CLEF 2013, CELCT, From Nuance Communications, USA, 2013, pp. 352-365

[23] M. Elhadad, S. Miranda-jiménez, J. Steinberger, and G. Giannakopoulos: Multidocument Multilingual Summarization Corpus Preparation, Part 2: Czech, hebrew and spanish, in In MultiLing 2013 Workshop in ACL, Bulgaria, 2013, pp. 13-19

[24] G. Giannakopoulos: Multi-Document Multilingual Summarization and Evaluation Tracks in Acl 2013 Multiling Workshop, in Proceedings of the MultiLing 2013 Workshop on Multilingual Multidocument Summarization, Bulgaria, 2013, pp. 20-28

[25] G. Giannakopoulos, J. Kubina, F. Meade, J. Conroy, J. M. Bowie,J. Steinberger, B. Favre, M. Kabadjov, U. Kruschwitz, M. Poesio. Multiling 2015: Multilingual Summarization of Single and Multi-Documents, On-Line Fora, and Call-Center Conversations. Proceedings of SIGDIAL, Prague, 2015, pp. 270-274

Automatic Detection of Semantic Primitives with Bio-inspired, Multi-Objective, Weighting Algorithms

Obdulia Pichardo-Lagunas

Instituto Politécnico Nacional, UPIITA, Av. IPN, s/n, 07320, Mexico City, Mexico, opichardola@ipn.mx

Grigori Sidorov

Instituto Politécnico Nacional, Centro de Investigación en Computación, Av. Juan de Dios Batiz, s/n, 07320, Mexico City, Mexico, sidorov@cic.ipn.mx

Alexander Gelbukh

Instituto Politécnico Nacional, Centro de Investigación en Computación, Av. Juan de Dios Batiz, s/n, 07320, Mexico City, Mexico, gelbukh@cic.ipn.mx

Nareli Cruz-Cortés

Instituto Politécnico Nacional, Centro de Investigación en Computación, Av. Juan de Dios Batiz, s/n, 07320, Mexico City, Mexico, nareli@cic.ipn.mx

Alicia Martínez-Rebollar

Centro Nacional de Desarrollo Tecnológico en Cómputo (CENIDET), Interior Internada Palmira, s/n, Palmira, 62490, Cuernavaca, Mexico,
amartinez@cenidet.edu.mx

Abstract: This paper proposes the usage of computational techniques that allow for automatic analysis of the vocabulary contained in an explanatory dictionary. It is proposed for the extraction of a set of words, called semantic primitives, which are considered those allowing the creation of a system used to establish definitions in dictionaries. The proposed approach is based on the representation of a dictionary as a directed graph and the combination of a multi-objective differential evolution algorithm with the PageRank weighting algorithm. The differential evolution algorithm extracted a set of primitives that fulfill two objectives: minimize the set size and maximize its degree of representation (PageRank), allowing the creation of a computational dictionary without cycles in its definitions. We experimented with a RAE dictionary of Spanish. Our results present improvement over other algorithms that are representative of the state-of-the-art.

Keywords: lexicography; computational lexicography; semantic primitives; defining vocabulary; explanatory dictionary; multiobjective bioinspired algorithms; differential evolution; weighting algorithms; PageRank.

1 Introduction

Traditional explanatory dictionaries are aimed at human readers. However, if an explanatory dictionary is to be used by computers, some important differences must be considered. Dictionaries for computers are important mainly because a large number of problems related to Computational Linguistics (CL) need to be addressed. Some of those problems are automatic translations and the generation of abstracts and the alignment of texts, among many others. In all these tasks we deal with *semantics*, therefore, it is quite beneficial to use vocabularies containing pre-coded information about deep relations among words and not only the isolated words.

The automatic construction of dictionaries for computer use is usually done starting from a traditional explanatory dictionary. However, traditional dictionaries have a major problem, that is, the existence of *cycles* in their definitions. Actually, in every traditional dictionary, the existence of cycles in the definitions is unavoidable, since the words are explained by cross references to another words reached in one or more steps. For example, we can define *treaty* as *pact*, *pact* as *agreement* and *agreement* as *treaty*, thus, returning to the first word in a two-step cycle. The idea behind dictionaries for humans is that their cycles should be as large as possible, then, it is probable that the person knows at least one of the words in the cycle. In this sense, the longer the paths, the better for humans. On the other hand, the dictionaries for computers cannot have cycles, because computers are not able to process them. So, when designing dictionaries for computers the main problem faced is how to break every cycle.

A dictionary can be represented as a directed graph. That is, for a determined entry the out arrows correspond to words in its definition. We will discuss formal representation of this idea in Section 2.

It is obvious that definitions contained in any dictionary are created using other words, but not all words are considered as the same category, actually there are special sets, i. e., words are considered either defining vocabulary or semantic primitives.

A defining vocabulary is a set of words with which the definitions are created in a dictionary. For example, the Longman vocabulary [9] (that was created and revised by human lexicographers), has about 3,000 words. If we consider a representation of the dictionary as a directed graph, then the words of the

Longman defining vocabulary would lie one step of distance from the dictionary entries, i.e., they are the words used in the definitions.

Conversely, the semantic primitives, named by Wierzbicka (1980, 1996) [15], [16] is the set of words characterized by lack of definition, i.e., the out arrows were removed from the graph, then, they guarantee that the graph has no cycles. Obviously, from each entry we reach only a small set of semantic primitives, not all of them.

This work aims to automatically extract a set of words considered semantic primitives, to create a dictionary without cycles for various CL tasks. In general, we consider that the smallest set of semantic primitives is the best one.

In the development of this work, we based on the following approaches: the hypothesis of the existence of a natural semantic meta-language [15]; the representation of the dictionary as a directed graph [11]; and the usage of an evolutionary algorithm for detecting semantic primitives [10]. We complement our previous works [10, 17] with the design of a multi-objective function for the algorithm Differential Evolution, and the usage of weights assigned by the PageRank algorithm [8] for semantic primitives identification.

The paper is organized as follows. Related works are discussed in Section 2. The proposed method is explained in Section 3. The validation of the experiments with the multi-objective function and the PageRank algorithm are shown in Section 4. Conclusions and future work are presented at the end of the paper.

2 Theoretical Framework

2.1 Related Work

The hypothesis proposed by Anna Wierzbicka [15], [16] claims the existence of a natural semantic meta-language (NSM), which is a vocabulary used to complete the lexicon of any language. Wierzbicka proposed a number of 60 words to be considered as primitives, they represent an irreducible semantic nucleus and that are used (with an additional set of rules) to generate new definitions. The core of this meta-language (the 60 words) is considered universal. Accordingly, the meaning of any expression can be specified through a reductive paraphrase. That is, any complex definition can be described using simpler terms than the original.

Apresjan (1995) [4] supports the idea of using restricted vocabularies for the development of lexicon, but he states that it cannot be as small as mentioned by Wierzbicka.

The Longman dictionary of contemporary English (LDOCE) [9] takes up the concept proposed by Apresjan and uses what is called a defining vocabulary.

In this dictionary, all the definitions are constructed using exclusively the restricted vocabulary. The size of this vocabulary is about 3,000 words in its latest version.

Kozima and Furugori (1993) [5] created a semantic network for LDOCE, in which each word of the dictionary is represented by a node, creating a closed system in which all words are defined by the same dictionary. The authors came to the following conclusion: "If there is a defining vocabulary, it corresponds to the dense part of the network, whereas words that are not defining are not linked to each other, therefore they are found on the periphery. This experiment was the first attempt for automatic vocabulary construction.

Rivera-Loza et al. (2003) [11] and Pichardo-Lagunas (2012) [10] returned to this problem using the Anaya dictionary and RAE dictionary for Spanish as cases of study. In both cases, the dictionary was represented as a directed graph, where each node represents a word. The graph was created by inserting word by word avoiding the existence of cycles in the system of definitions. For each iteration, if a word closed a cycle, then, it was considered as semantic primitive. Note that the order in which the words are added to the graph is important, i.e., different input permutations will generate different output sets. Since we look for the smallest set of primitives, our final goal is to find the input permutation that reduces the number of words considered as such.

In Rivera-Loza et al. [11], the entry words order was given by two methods: randomly and frequencies by random voting. The method of random frequencies obtained the best result with a total of 2,246 semantic primitives.

Pichardo-Lagunas et al. [10] proposed the use of heuristic methods, specifically the algorithm differential evolution (DE) adapted for handling permutations. The DE algorithm generated different permutations for constructing the graph and obtaining a total of 2,169 primitives.

There are some theoretical works related to semantic primitives for the English language, however, for the best of our knowledge, no other work for the automatic semantic primitives' detection is known to date.

2.2 Graph Theory Concepts

A directed graph G is a tuple $G = (V, F)$, where $V \neq \emptyset$, whose elements are called vertices, $F \subseteq V \times V$. The elements of F are called directed edges, see Figure 1(a).

A directed path in *G* is a finite sequence of vertices of *G* denoting:

$$V1, V2, \ldots, Vn.$$

Definitions:

- A directed path T is closed if and only if $V1 = Vn$,
- A closed directed path is a directed cycle or cycle,
- A semantic primitive is the vertex V that closes the directed path.

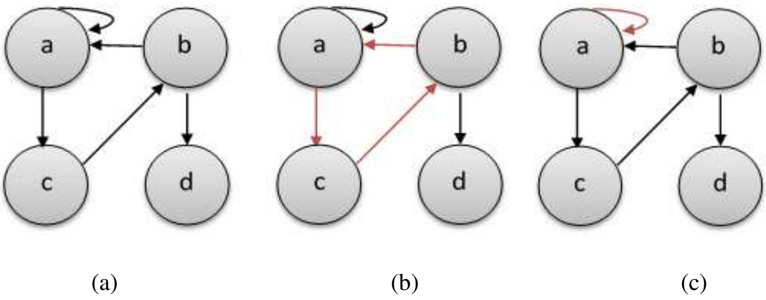


Figure 1
(a) Directed graph, (b) Cycle in a directed graph, (c) Loop in a directed graph

The above cases are shown in Figure 1(a), (b) and (c). Figure 1(a) represents an example of a directed graph. Figure 1(b) contains a cycle "*a→c→b→a*". Figure 1(c) contains another cycle "*a→a*" (the loop).

Let *G = (V, F)* be a directed graph, then *G '= (V', F')* is a subgraph of *G* if *V'* $\neq \emptyset$ and *F'*⊂*F*, where ∀ edge of *F'* is incident to the vertices of *V'*.

2.3 Multi-Objective Optimization

Multi-objective optimization attempts to find a solution vector that simultaneously optimizes more than one objective function. These functions typically are in conflict to each other, which means that improvement in one function makes worse the performance of the other ones. Multi-objective optimization can be mathematically defined as:

Find the vector $\vec{x}^*$ that optimizes the target function vector

$$f_{1(\vec{x})}, f_{2(\vec{x})}, \cdots f_{k(\vec{x})}$$

subjected to *m* inequality constraints

$$g_i = (\vec{x}) \leq 0; i = 1, \ldots, m,$$

and *p* equality constraints

$$h_i(\vec{x}) = 0; i = 1, \ldots, p.$$

2.3.1 Pareto Optimum

The Pareto Optimum is a set of solutions that reaches a compromise among the different objective functions. A formal definition is as follows:

A decision vector of variables $\vec{\mathrm{x}}^* \in \mathrm{F}$ (where F is the feasible area) is Pareto optimal if there is no other $\vec{x} \in F$ such that: $f_i(\vec{x}) \leq f_i(\vec{x}^*)$ for all i = 1,.., k and $f_j(\vec{x}) \leq f_j(\vec{x}^*)$ for at least one *j*.

In other words, "Pareto optimum is that vector of variables, in which the solutions of the problem cannot be improved in one objective function without worsening any of the others" (Abbass, 2002) [2].

The Pareto optimum provides a set of solutions called Pareto Optimal Set.

2.3.2 Pareto Dominance

The term Pareto Dominance can be defined as follows:

$A\ vector\ \vec{u} = (u_1, \ldots, u_k)\ dominates\ another\ vector\ \vec{v} = (v_1, \ldots, v_k)$
$if\ and\ only\ if\ it\ dominates\ anoter\ \vec{u}\ which\ is\ partially\ smaller\ than\ \vec{v}.$

For example, when comparing two different solutions A and B, there are three possible situations:

- A dominates B,
- A is dominated by B,
- A and B are not dominated to each other.

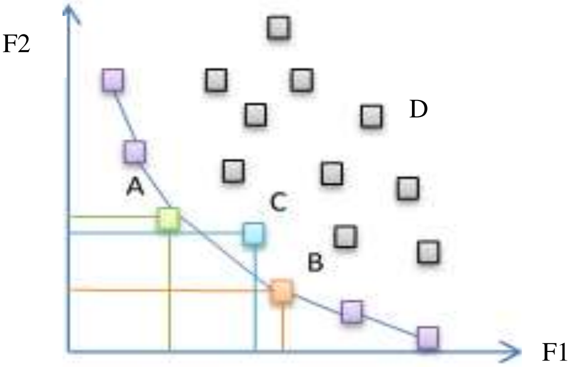


Figure 2

Objective function space illustration for two objectives F1 and F2

See for example in Figure 2, an illustration of two-objective functions F1 and F2. The solution B dominates to all the solutions represented by grey squares because B has smaller values for F1 and F2 than all of them. Further, A and B are not dominated to each other because B has smaller value for F2 but A has a smaller value in F1. The point D is dominated by A because is smaller in F1 and F2 than D.

2.3.3 Pareto Front

The solutions whose vectors are not dominated and are also in the Pareto optimal set are called the Pareto front. The formal definition is as follows:

For a given multi-objective problem $\vec{f}(x)$ and a set of Pareto optimal P^*, the Pareto front (FP^*) is:

$$FP^* := \{\vec{f} = [f_1(x), \ldots, f_k(x)] \mid x \in P^*\}.$$

2.4 Differential Evolution Algorithm

The differential evolution (DE) is a population-based evolutionary algorithm, developed for optimization in continuous spaces [13].

The general DE idea is as follows: The initial population is a set of real numbers randomly generated and stored in a vector. Then, three individuals are selected to play the role of parents. One of the candidates is the main father and is altered with information taken from the other two parents. If the resulting value (solution) from the previous operation is better than the current individual, then it is replaced. Otherwise, the parent is retained. The process is repeated until a determined criterion is reached.

As mentioned earlier, the DE algorithm was designed to work with potential solutions represented by real numbers. In the problem that is being addressed, we look for solutions with representation of permutations, so we used an adaptation that allowed us to convert the representation of permutations into real numbers [14]. Next, there is an example:

The vector solutions are a permutation of the integers from 1 to 5. Given two vectors X_{r1} y X_{r2} as follows:

$$X_{r1} = \begin{bmatrix} 1 \\ 3 \\ 4 \\ 5 \\ 2 \end{bmatrix}, \qquad X_{r2} = \begin{bmatrix} 1 \\ 4 \\ 3 \\ 5 \\ 2 \end{bmatrix}.$$

Then we transform them into $X_{r1,f}$ y $X_{r2,f}$. The subscript f denotes a floating point representation vector. This way now the vectors are real numbers and the algorithm DE can be directly applied as in its original version.

$$X_{r1,f} = \frac{X_{r1}}{5} = \begin{bmatrix} 0.2 \\ 0.6 \\ 0.8 \\ 1 \\ 0.4 \end{bmatrix}, \qquad X_{r2,f} = \frac{X_{r2}}{5} = \begin{bmatrix} 0.2 \\ 0.8 \\ 0.6 \\ 1 \\ 0.4 \end{bmatrix}.$$

Continuing with the general DE process, a third vector is randomly selected:

$$X_{r3} = \begin{bmatrix} 5 \\ 2 \\ 1 \\ 4 \\ 3 \end{bmatrix} \rightarrow X_{r3,f} \begin{bmatrix} 1 \\ 0.4 \\ 0.2 \\ 0.8 \\ 0.6 \end{bmatrix}.$$

Then the mutation can be as:

$$v_f = X_{r3,f} + F\left(X_{r1,f} - X_{r2,f}\right)^{F=0.85} = \begin{bmatrix} 1 \\ 0.4 \\ 0.2 \\ 0.8 \\ 0.6 \end{bmatrix} + 0.85 \begin{bmatrix} 0 \\ -0.2 \\ 0.2 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.23 \\ 0.37 \\ 0.8 \\ 0.6 \end{bmatrix}.$$

The resulting vector must be transformed back into integers:

$$v_f = \begin{bmatrix} 1 \\ 0.23 \\ 0.37 \\ 0.8 \\ 0.6 \end{bmatrix} \rightarrow v = \begin{bmatrix} 5 \\ 1 \\ 2 \\ 4 \\ 3 \end{bmatrix},$$

which is an adequate representation of our problem.

Due to the characteristics of the problem to be solved, a multi-objective ED algorithm was implemented, specifically, the Pareto Differential Evolution (PDE) [1], which is a modification of the original ED and whose algorithm is presented below.

```
$Let\ G\ denotes\ a\ generation, P\ a\ population\ of\ size\ M,$
    $and\ \vec{x}^{j}_{G=k}\ the\ j^{th}\ the\ individual\ of$
        $dimensions\ N\ in\ population\ P\ in\ generation\ k,$
        $and\ CR\ denotes\ the\ croosover\ probability$
$\boldsymbol{input}\ N, M \geq 4, F \in (0,1+), CR \in [0,1], and\ initial:$
        $bounds: lower\ (x_i), upper(x_i), i = 1, \dots N$
$initialize\ P_{G=0} = \{\vec{x}^{1}_{G=0}, \dots, \vec{x}^{M}_{G=0}\}\ as$
$\boldsymbol{For\ each}\ individual\ j \in P_{G=0}$
    $x^{j}_{i,G=0} = Gaussian\ (0.5, 0.15), i = 1, \dots N$
    $Repair\ \vec{x}^{j}_{i,G=k}\ if\ any\ variable\ is\ outside\ its\ boundaries$
$\boldsymbol{end\ for\ each}$
```

```
$evaluate\ P_{G=0}$
$\boldsymbol{k = 1}$
$\boldsymbol{while}$ $the\ stopping\ criterion\ is\ not\ satisfied$ $\boldsymbol{do}$
    $remove\ all\ dominated\ solutions\ in\ P_{G=k-1}$
    $\boldsymbol{if}$ $the\ number\ of\ non\ dominated\ solutions\ in\ P_{G=k-1} > \alpha$
        $\boldsymbol{then}$ $apply\ the\ rule\ of\ neighbord\ rule$
    $\boldsymbol{end\ if}$
    $\boldsymbol{for\ j = 0}$ $to\ the\ number\ of\ non\ dominated\ solutions\ in\ P_{G=k-1}$
        $\vec{\boldsymbol{x}}^{\boldsymbol{j}}_{\boldsymbol{G=k}} \leftarrow \vec{\boldsymbol{x}}^{\boldsymbol{j}}_{\boldsymbol{G=k-1}}$
    $\boldsymbol{end\ for}$
$\boldsymbol{while\ j \leq M}$
    $randomly\ select$ $\boldsymbol{r_1, r_2, r_3} \in \boldsymbol{(1, \dots, \alpha)}, of\ the$
          $non\ denominated\ solutions\ of$ $\boldsymbol{P_{G=k-1}}, where$ $\boldsymbol{r_1 \neq r_2 \neq r_3}$
    $randomly\ select$ $\boldsymbol{i_{rand}} \in \boldsymbol{(1, \dots, N)}$
    $\boldsymbol{for\ all\ i \leq N, \vec{x}'_{i,G=k} =}$

        $\begin{cases} \boldsymbol{x^{r_3}_{i,G=k-1} + Gaussian\ (0, 1)} \times \boldsymbol{(x^{r_1}_{i,G=k-1} - x^{r_2}_{G=k-1}} \\ \qquad\qquad \boldsymbol{x^{j}_{i,G=k-1}} \\ \qquad\qquad \boldsymbol{otherwise} \end{cases}$
     $\boldsymbol{end\ forall}$
     $Repair$ $\vec{\boldsymbol{x}}^{\boldsymbol{j}}_{\boldsymbol{G=k-1}}$ $if\ each\ variable\ is\ outside\ its\ boundaries$
     $\boldsymbol{if\ \vec{x}' = dominates\ \vec{x}^{r_3}_{G=k-1} then}$
         $\vec{\boldsymbol{x}}^{\boldsymbol{j}}_{\boldsymbol{G=k}} \leftarrow \vec{\boldsymbol{x}}'$
         $\boldsymbol{j = j + 1}$
     $\boldsymbol{end\ if}$
     $\boldsymbol{k = k - 1}$
     $\boldsymbol{end\ while}$
$\boldsymbol{return\ non\ dominated\ solutions}$
```

The next considerations were applied to the multi-objective algorithm:

1. The initial population is generated with a Gaussian distribution N (0.5, 0.15).
2. The parameter F is generated with a Gaussian distribution N (0, 1).
3. Reproduction is performed only with non-dominated solutions at each generation.
4. Limits on the variables are preserved by changing its sign, if it is less than 0, or subtracting 1 if it is greater than 1, until the variable is within the allowed limits.
5. A generated individual is placed in the population if he dominates his father.

The multi-objective DE algorithm is summarized as follows. An initial population is generated, all dominated solutions are removed from the population and the rest are used for reproduction. Three parents are randomly selected to generate a child. The offspring is placed in the population if it dominates the main father, otherwise he is forgotten. This process is repeated until the population is complete [12].

2.5 PageRank Algorithm

The PageRank algorithm was proposed by Larry Page and Sergey Brin (1998) [8] and is used to assign a numerical value that corresponds to the relevance of the different web pages that can be indexed by the search engines.

The PageRank algorithm is based on a democratic system that uses the link system as an indicator of the relevance of a particular webpage. Google interprets the links between pages as votes considering also the relevance of the page that contains the league. That is, the votes of a relevant page are more important than those of a page with less relevance. The algorithm at the beginning assigns random values and then iterates until no changes are produced.

The PageRank algorithm is described as follows:

$$PR(A) = (1-d) + d\sum_{i=1}^{n}\frac{PR(i)}{C(i)},$$

where PR(A) is the PageRank of page A, *d* is damping factor having a value between 0 and 1 (usually, 0.85), PR(i) are the PageRank values that have each page *i* that has links to A (incoming links), C(i) is the total number of outgoing links of the page *i* (whether or not to A).

3 Proposed Method for Automatic Detection of Semantic Primitives

The approach of this research is divided in three stages: (1) preprocessing, which consists of the dictionary debugging and the construction of the graph, (2) execution of the PageRank algorithm, that serves to weight the nodes that belong to the graph and (3) application of the algorithm of Differential Evolution, that determines different input permutations to obtain the set of semantic primitives and the construction of the dictionary without cycles.

3.1 Preprocessing and PageRank Algorithm

For the experiments, the dictionary of the Royal Spanish Academy (RAE for its acronym in Spanish), edition of 2007 was used. The RAE has a total of 152,370 entries. As it is common in computational linguistics, we ignore stopwords (like prepositions, conjunctions, etc.), i.e., we only used the content words: verbs, adverbs, nouns and adjectives. To identify the content words, the dictionary was tagged using the Freeling tool [7].

The description of words that have more than one meaning were grouped into a bag of words, that is, even if a word has more than one meaning, in the graph it was represented only once. We plan to take into account word senses in our future work.

Words that were contained in their own definition (that is, those that make loops or cycles) were detected and were considered as semantic primitives.

Words not used in other definitions were not added to the graph because they would not have the possibility of closing some cycle and therefore have no opportunity to be considered as semantic primitives. This process was done iteratively because each time a set of words were deleted some other words were no longer used in the set of definitions.

Once the list of dictionary entries was generated, a number was assigned to each of them, which functions as an index. The index identifies the word and allows the evolutionary algorithm to work only with numbers and not with the string of characters. With the indexes already assigned, the adjacency list representing the dictionary was generated as a graph.

Preprocessing was the same as that used in Rivera-Loza et al. [11] and Pichardo-Lagunas et al. [10]. Summarizing, it includes the following steps:

1. Delete additional information (like the origin, for example, "from Latin *Ab*").
2. Remove prefixes and suffixes from dictionary entries.
3. Delete entries contained in your own definition.
4. Remove entries that are not used in the rest of the definitions.
5. Tag dictionary words using Freeling.
6. Remove stopwords from entries and definitions.
7. Remove entries that are not used in definitions.

The adjacency list generated after the preprocessing was used as the input in the PageRank algorithm. The algorithm calculated the weighting of each node according to the relations that it maintains with the rest of the nodes. The information provided by the PageRank algorithm serves to evaluate the second objective of the PDE function that seeks to maximize the sum of the weighting associated to each of the nodes of the extracted set.

3.2 Pareto Differential Evolution (PDE)

In the context of the problem, it is necessary to construct the graph *G'*, which is a sub graph of *G,* where *G* is the preprocessed dictionary. The sub graph *G'* is constructed by inserting node after node verifying that it keeps without cycles. Thus, we try different input permutations.

The Pareto Differential Evolution looks for a permutation σ that is an input permutation for the graph G', which is constructed by inserting node by node (according to σ). With each insertion in *G'* it is verified that no cycle is generated between the definitions of the graph, if so, the vertex is not inserted and it is considered a semantic primitive.

To apply the algorithm of differential evolution it is required:

- Representation of possible solutions (permutations),
- Creation of an initial population of possible solutions (random values),
- Definition of the evaluation function (fitness function),
- Other parameters, such as:
    1. Population size,
    2. Probability of crossover,
    3. Probability of mutation,
    4. Maximum number of generations.

The PDE algorithm requires as input a list of indices of words:

$$x_{i=0}^{m} = \{(I1, j_1), (I2, j_2) \dots (In, j_n)\},$$

where *n* is the total of entries in the dictionary, *m* is the total of vectors in the population, *j* is the weight associated to the node, and the vector x^m is a permutation.

3.3 Fitness Function

The fitness of an individual is measured according to the Pareto dominance criterion (see 2.3.1), which is determined by evaluating the objective function in each set. The objective function for our problem is defined as follows:

$$\begin{cases} Minimize |P|, where\ p = \{x | x \in V \land x \notin G'\}, \\ Maximize\ S, where\ s = \left\{ \sum_{i=1}^{n} PR(p_i) \right\}. \end{cases}$$

The first objective function seeks to minimize *P* where *p* is a set of nodes belonging to *V* (which is the set of words of *G*) and which do not belong to *G'*. Where *G* is the complete graph and *G'* is the graph constructed without cycles. At the same time, the second objective function seeks to maximize *S,* where *s* is the sum of the weights obtained by PageRank associated with the p_i that represents the individual.

4 Experiments and Results

We conducted our experiments using the most influential RAE dictionary of Spanish. The RAE dictionary has 152,370 words with definitions. After the preprocessing tasks, this number is reduced to 77,300. The generated list was used as input for the execution of PageRank algorithm with three different parameters. In the first case, we used 70 iterations and the damping factor of 0.75. The second run used 50 iterations and 0.8 as the damping factor. The third case applied 100 iterations and the damping factor of 0.85, which are the parameters specified by Page and Brin (1996) [8].

Although the values generated by the algorithm varied in each case, the average difference remained within the range of ±3.0%, so we used the list generated by the third configuration.

The vector with the indices of identification and the weighting associated to each node served as the input for the algorithm of Pareto Differential Evolution. The proposed algorithm obtains given a directed graph *G*, a defining subset *P*, where $P \subseteq V$ and each *p* from *P* is considered semantic primitive, since any cycle in the graph *G* contains a vertex to *P*.

The first purpose of the objective function is to minimize the set of semantic primitives, seeking to maintain the *G'* graph with as few words as possible. The other objective is to maximize the PageRank value of *P*.

The PDE algorithm was executed 30 times [6]. In each of the executions different configuration parameters were used for the algorithm.

The configuration of parameters that obtained the best results was:

- 500 individuals,
- 300 generations,
- Probability of crossover: 0.2.

In each iteration, a set of non-dominated solutions of different sizes was obtained, but as proposed in Santana-Quintero (2004) [12] the final size of the set was reduced to 50 using the neighborhood distance function.

It was obtained a set of 50 non-dominated solutions. The one that is identified as the best solution is that obtained the least number of semantic primitives.

Table 1

Runs with best results

	Number of individuals	Number of generations	Probability of crossover	Number of primitives
Run 17	300	500	0.16	2,234
Run 23	500	500	0.1	2,252
Run 24	300	500	0.2	2,228
Run 30	500	300	0.2	**2,148**

The iteration with the best results obtained a total of 2,148 primitives with a sum PageRank value of 1,776.52. The smallest set was selected because the essential objective of this research is to find the set with the least number of words. The system also generated sets of words that obtained higher PageRank values, but for sets with greater size. These sets should be subjected to analysis in later works.

Table 2

Values of Page Rank for some runs

	Number of primitives	Sum of PageRank values
Run 17	2,234	1,762.179
Run 23	2,252	1,770.031
Run 24	2,228	1,785.185
Run 30	**2,148**	1,776.522

To carry out the validation between the obtained set and the complete vocabulary, we used an automatic translation of Longman vocabulary from LDOCE. A coincidence of the word from our set with this vocabulary means that at least one of the meanings of the translations of the English word coincides with at least one of the meanings of a word of the generated set in Spanish. The absolute coincidences with LDOCE are calculated by dividing the number of primitives that are ate same time present in LDOCE by the size of LDOCE. This measure shows which part of LDOCE is covered by the obtained set of primitives. The relative coincidences are calculated by dividing the number of primitives in LDOCE by the size of the obtained set of primitives. This measure shows which part of the set of primitive belongs to LDOCE.

Table 3
Comparison of matches with LDOCE vocabulary

	Number of primitives	Relative coincidences with LDOCE	Absolute coincidences with LDOCE
Pichardo-Lagunas *et al.*	2,169	1,594 (56.05%)	73.87%
Rivera-Loza *et al.*	2,246	1,487 (52.15%)	66.20%
ED Pareto	2,148	1,719 (80.02%)	72.95%

Considering the work done by Rivera-Loza et al. [11] and Pichardo-Lagunas et al. [10], a comparison was made between them and the results obtained by the multi-objective function presented in this work. Pichardo-Lagunas et al. obtained a 73.87% coincidence with the vocabulary Longman and the set obtained by PDE reached 72.95%, with a difference of 1%. As compared to Rivera-Loza et al.'s work, an improvement of 6.75% was obtained. It should be noticed that the relative coincidences with LDOCE of the proposed method augmented about 25%.

Conclusions

For the experiments carried out in other works, such as, Rivera-Loza et al. and Pichardo-Lagunas, the number of obtained primitives shows a certain level of stability.

The Pareto Differential Evolution algorithm (PDE) was applied, which improved the results obtained by previous works by 1.02% with respect to the size of the obtained set and the relative coincidences with LDOCE augmented to about 25%. Thus, including the importance of words (nodes) as an evaluation parameter (application of the PageRank algorithm), the sets of the primitives tend to decrease their size and the nodes of the obtained sets have relations of major importance within the graph.

Acknowledgements

This work was partially supported by the Mexican Government (CONACYT project 240844, SNI, COFAA-IPN, SIP-IPN 20161947, 20161958, 20162204, 20162064).

References

[1] Abbass, H. & Sarker, R. (2002). The Pareto Differential Evolution Algorithm. International Journal on Artificial Intelligence Tools, 11(4):531−552

[2] Abbass, H. (2002). The Self-Adaptive Pareto Differential Evolution Algorithm. Congress on Evolutionary Computation CEC'2002. Volume 1, 831−836, Piscataway, New Jersey.

[3] Abbass, H., Sarker, R. & Newton, C. (2001). PDE: A Pareto-frontier Differential Evolution Approach for Multi-objective Optimization Problems. Proceedings of the Congress on Evolutionary Computation, Vol. 2, New Jersey, 971−978.

[4] Apresjan, J. (1995). Selected works (in Russian). Moscow.

[5] Kozima, H. & Furogori, T. (1993). Similarity between words computed by spreading activation on an English dictionary. Proceedings of the 6th conference of the European chapter of ACL, 232−239.

[6] Levine, D., Berenson, M. & Krehbiel, T. (2006). Estadística para administración. Pearson Education, México.

[7] Padró, L., Collado, M., Reese, S., Lloberes, M. & Castellón, I. (2010). Freeling 2.1: Five years of open-source language processing tools. Proceedings of the 7th Language Resources and Evaluation Conference, La Valleta, Malta.

[8] Page, L. & Brin, S. (1998). The anatomy of large-scale hypertextual Web search engine. Computer Networks and ISDN Systems, 30(1-7), 107−117.

[9] Pearson Education (1991). Longman Dictionary of Contemporary English. London.

[10] Pichardo-Lagunas, O. (2012). Detección automática de primitivas semánticas con algoritmos bioinspirados. Tesis de doctorado, CIC-IPN, México.

[11] Rivera-Loza, G., Gelbukh, A. & Sidorov, G. (2003). Selección automática de primitivas semánticas para un diccionario explicativo del idioma español. Tesis de maestría, CIC-IPN, México.

[12] Santana-Quintero, L. (2004). Un algoritmo basado en evolución diferencial para resolver problemas multiobjetivo. Tesis de maestría, CINVESTAV-IPN, México.

[13] Storn, R. & Price, K. (1995). Differential evolution – a simple and efficient adaptative scheme for global optimization over continuous spaces. Technical Report TR-95-12, International Computer Science, Berkeley, California.

[14] Storn, R., Price, K. & Lampinen, J. (2005). Differential Evolution. A practical Approach to Global Optimization. Springer.

[15] Wierzbicka, A. (1980). Lingua Mentalis: The semantics of natural language. New York: Academic Press. xi, 368.

[16] Wierzbicka, A. (1996). Semantics: Primes and Universals. Oxford University. Oxford.

[17] Pichardo-Lagunas, O., Sidorov, G., Cruz-Cortés, N. & Gelbukh, A. (2014). Detección automática de primitivas semánticas en diccionarios explicativos con algoritmos bioinspirados [Automatic detection of semantic primitives in dictionaries using bio-inspired algorithms]. Onomazein, 29:104−117.

Semantic Approach for Discovery and Visualization of Academic Information Structured with OAI-PMH

Joanna Alvarado-Uribe[1], Arianna Becerril García[2], Miguel Gonzalez-Mendoza[1], Rafael Lozano Espinosa[1], José Martín Molina Espinosa[1]

[1] Tecnologico de Monterrey, School of Engineering and Sciences, Av. Eugenio Garza Sada No. 2501 Sur, Col. Tecnológico, 64849, Monterrey, N.L., México, {A00987514, mgonza, ralozano, jose.molina}@itesm.mx

[2] Universidad Autónoma del Estado de México, Instituto Literario Ote. No. 100, Col. Centro, 50000, Toluca, Estado de México, México, abecerrilg@uaemex.mx

Abstract: There are different channels to communicate the results of a scientific research; however, several research communities state that the Open Access (OA) is the future of academic publishing. These Open Access Platforms have adopted OAI-PMH (Open Archives Initiative - the Protocol for Metadata Harvesting) as a standard for communication and interoperability. Nevertheless, it is significant to highlight that the open source knowledge discovery services based on an index of OA have not been developed. Therefore, it is necessary to address Knowledge Discovery (KD) within these platforms aiming at students, teachers and/or researchers, to recover both, the resources requested and the resources that are not explicitly requested – which are also appropriate. This objective represents an important issue for structured resources under OAI-PMH. This fact is caused because interoperability with other developments carried out outside their implementation environment is generally not a priority (Level 1 "Shared term definitions"). It is here, where the Semantic Web (SW) becomes a cornerstone of this work. Consequently, we propose OntoOAIV, a semantic approach for the selective knowledge discovery and visualization into structured information with OAI-PMH, focused on supporting the activities of scientific or academic research for a specific user. Because of the academic nature of the structured resources with OAI-PMH, the field of application chosen is the context information of a student. Finally, in order to validate the proposed approach, we use the RUDAR (Roskilde University Digital Archive) and REDALYC (Red de Revistas Científicas de América Latina y el Caribe, España y Portugal) repositories, which implement the OAI-PMH protocol, as well as one student profile for carrying out KD.

Keywords: the Semantic web; knowledge discovery; user profile ontology; ontology merging; OAI-PMH; visualization

1 Introduction

There are different channels to communicate the results of a scientific research; however, the Open Access (OA) is the future of academic publishing [2]. These Open Access Platforms have adopted OAI-PMH (Open Archives Initiative - the Protocol for Metadata Harvesting) as a standard for communication and interoperability. For instance, according to ROAR (Registry of Open Access Repositories), there are more than 4,000 repositories in the World that implement OAI-PMH [19]. Such figures show the consolidation of repositories as well as the large amount of scientific-academic resources following the philosophy of OA, which are available online for query. Nevertheless, it is significant to highlight that the open source knowledge discovery services based on an index of OA have not been developed [8]. Therefore, it is necessary to address the Knowledge Discovery (KD) within these platforms aiming students, teachers, or researchers to recover both useful resources requested and resources not explicitly requested by them – which are also appropriate –. This objective represents an important issue for structured resources under the OAI-PMH protocol. This fact is caused because interoperability with other developments carried out outside their implementation environment is generally not a priority (Level 1 "Shared term definitions") [15]. It is here, where the Semantic Web (SW) becomes very important for this work.

Accordingly, it is noteworthy that our research work arises in the context of the original vision of SW. That is, the vision of providing more meaning to Web information through the logical connection of terms in order to establish interoperability between systems [47]. On the one hand, we consider SW as part of the Open World Assumption (OWA) – also known as the Classical Paradigm [40] – stipulating that there may be unspecified information (considered as unknown) that can be inferred [40]. On the other hand, we take into account that SW is based on the idea of adding more machine-readable semantics to Web information through annotations written in Resource Description Framework (RDF) [26]. Such that, the incorporation of the RDF model, in this work, will bring about two key features: to gain the W3C design principles and some main features of SW, such as interoperability, extensibility, evolution, and decentralization; and to allow anyone can make statements about any resource [26].

Thus, our work arises from these facts, with the aim of designing and developing a solution to allow a user KD on structured resources with OAI-PMH, by applying the SW technologies. Likewise, it is to both to improve the information retrieval and also the visualization of outcomes within this approach. Therefore, it is relevant to develop technologies that support the discovery of interesting resources within the structured repositories with OAI-PMH for any user, taking into account his/her context information [5]. Analogously, due to a large amount of information gathered and integrated, it is necessary to incorporate a layer for data visualization allowing visual interpretation of the retrieved information in a quick, simple, and easy to understand manner [3]. Consequently, we propose OntoOAIV, a semantic approach for the selective knowledge discovery into structured information with OAI-PMH, focused on supporting the activities of scientific or academic research for a specific user.

Because of the academic nature of the structured resources with OAI-PMH, the field of application chosen is the context information of a student. The idea of building a user profile model is supported in the envisioning of some researchers, belonging to the user modeling community, that propose to use and to share the user models' information among applications. By using and sharing this information, we could integrate the preferences, interests, and characteristics of the user into the context of applications in order to enhance the service provided [35]. Hence, a student profile model is developed. This modeling is carried out using an ontology, because it provides common conceptualizations for data integration [47] and represents one of the two major approaches used to address the lack of interoperability in the user modeling [35]. Furthermore, keeping in mind these arguments and that Dublin Core (DC) (Level 1) is part of the OAI-PMH [15] protocol, we use some initiatives implemented in SW, such as DC (Level 2) [26, 13, 14, 15], Friend Of A Friend (FOAF) [26, 9, 1], and the DBPedia's PersonData [12], in order to provide more user context information to the proposed approach. According to the foregoing, we verify that the conceptualization and the SW's technologies are useful for dealing with the protocol's gaps.

Thereby, the main contribution of our work is the designed and implemented semantic approach, which allows KD within academic contents structured with OAI-PMH considering the user's context. Highlighting three specific contributions: the usage of the algorithm for merging ontologies proposed by Ameen et al. [4], the adaptation of the students' representation model based on the ontological approach presented by Panagiotopoulos et al. [39], and the incorporation of the tool for visualizing information stored in triples developed by Alvarado-Uribe et al. [3]. Accordingly, this work also contributes to the four defined rules within the Linked Data area "for publishing, sharing, and interlinking structured data on the Web" [51]. Finally, in order to validate the proposed approach, we use the RUDAR (Roskilde University Digital Archive) [45] and REDALYC (Red de Revistas Científicas de América Latina y el Caribe, España y Portugal) [44] repositories, which implement the OAI-PMH protocol, as well as one student profile for carrying out KD [5].

The paper is organized as follows. In Section 2, a review of the related work is provided. In Section 3, the semantic approach proposed is explained. In Section 4, the experiments of our approach are described. Then, in Section 5, the results of these tests are reported and discussed. The conclusions and future work are given in the last section.

2 Related Work

Scientific collaboration has long promoted the reuse and sharing of knowledge and data widely [29]; therefore, regarding non-commercial solutions, since 2001, a movement has been consolidating. Such a movement promotes free and unrestricted access to scientific content, especially when this content has been publicly funded. This movement is called OA and was formalized by means of three declarations: Budapest [11], Berlin [36], and Bethesda [10]. In this sense, it is important to mention that the beginning of the Directory of Open Access Journals (DOAJ) in 2003, developed by the Lund University in Sweden, ushered in the formalization and organization of OA for the case of scientific journals. Currently, DOAJ contains more than 9,000 open access journals from 128 countries [16].

Repositories, portals, and journals that are integrated into OA adopt – as a good practice – an interoperability protocol to exchange information in order to have communication rules and standards for structuring data. For instance, OAI-PMH is a low-barrier mechanism for the interoperability among repositories [31], which provides a framework for the independent interoperability of the application based on metadata harvesting. It is noteworthy that metadata to be transmitted via OAI-PMH should be coded in the Dublin Core (DC) format within an XML file, which usually includes several DC records depending on the configuration of each data provider and harvesters. The OAI-PMH interoperability protocol is established as a standard for publication in OA since various solutions of free and open software include the implementation of this protocol to build repositories and to manage scientific publications, such as DSpace [17] and Open Journal Systems (OJS) [41]. As of 2015, more than 32,000 OJS installations have been identified, of which, 8,286 contain at least 10 articles published, for a total of 2.8 million articles available through OAI-PMH [41].

Inasmuch as the main objective of this article is knowledge discovery; it is relevant to clarify this concept. Due to their essential goal, libraries have a strategic interest in the tools and technologies that facilitate the discovery and access to resources for the communities they serve. In this field, the resource discovery systems represent the next generation of OPAC (Online Public Access Catalog) within the Integrated Library Systems (ILS), which are commonly known as web-scale discovery services [8]. In this regard, Schonfeld defines discovery as "the process and infrastructure required for a user to find an appropriate item" [46]. However, that recovery ability is focused on user-generated searches, but not in the discovery based on semantic recovery or through inference, which would give the user a more smart or enriched information retrieval. Thus, KD is approached from the computational point of view. KD is the most desirable end product of computing, Frawley et al. [23] define KD as non-trivial extraction of implicit, previously unknown, and potentially useful information, from data. Therefore, considering the previous definitions, the OntoOAIV model follows the KD approach. In accordance with the foregoing, we have divided this section into these two main sub-areas in order to compare and to position our work.

2.1 Resource Discovery Systems for Libraries

Regarding commercial solutions named as "discoverers", in the libraries context, there are EBSCO Discovery Service [18], WorldCat Discovery Service [38], Summon [43], Ex Libris Primo [21], among others. These discoverers do not perform KD as we proposed for the OntoOAIV model. Hence, in Table 1, we present a review of some libraries resource discovery products and services against the OntoOAIV's aims in order to make clear the differences with our approach. This comparison is divided into the following five criteria: Does the approach use the Semantic Web technologies? (SW column); Is the approach specialized on indexing structured resources with OAI-PMH or considering the meta-data of this protocol to gain an aggregate value? (OP column); Does the approach enable the knowledge discovery? (KD column); Does the approach consider the user context through a user profile? (UP column); Does the approach allow visualizing the outcomes of a query? (V column).

Table 1

Comparison of the resource discovery-oriented approaches for libraries and the OntoOAIV's aims

Approach	Description	Status	SW	OP	KD	UP	V
Ex Libris Primo [21]	Product (software) that allows libraries to access their collections	Active (commercial)		X			
WorldCat Discovery Service [38]	Search platform of libraries resources and external databases	Active (commercial)		X			
BLUEcloud PAC [49]	Platform for libraries services	Active (commercial)		X			
BiblioCore [6]	New generation of OPAC for libraries	Active (commercial)					
AquaBrowser [42]	Product (software) that allows libraries to access their collections	Active (commercial)					
Summon Service [43]	Web-scale discovery service for libraries and other resources of an institution	Active (commercial)		X			
Encore [27]	Resource discovery solution for libraries	Active (commercial)		X			
EBSCO Discovery Service [18]	Search platform of libraries resources and external databases	Active (commercial)		X			
Blacklight [7]	Discovery platform framework for libraries	Active (open code)		X			
VuFind [54]	Library resource portal for	Active		X			

	searching and for retrieving library's resources	(open code)					
EXtensible Catalog [22]	Next generation software for libraries	Active (open code)		X			
Franklin [53]	Discovery tool that provides access to multiple collections	Active (open code)		X			

2.2 Semantic Approaches to Information Retrieval and Discovery

In this section, some tools focused on semantic searches are introduced. A recent approach, called intelliSearch, is presented by Mehta et al. [37], introducing an implementation of a semantic web search engine based on semantic relatedness. Similarly, other projects that follow the approach of using semantic search techniques are Evi [20], AquaLog [33], and Yummly Recipe Search [55, 30]. In the same way, SemSearch [32] is based on a semantic search engine that supports complex queries in terms of multiple keywords. In addition, an application called Semantic Search, proposed by Guha and McCool [25], uses the SW data to improve a web search. It is important to mention that some SW's engines were born free but after a few years they have become commercial solutions or have been acquired by large companies, such is the case of Sindice [52] and Freebase [24]. The development of "Google Knowledge Graph" is partly based on Freebase. The "Knowledge Graph" [48], a Google search feature, was considered as a first step in building next-generation search engines.

In Table 2, we present a review of some semantic approaches to information retrieval and discovery against the OntoOAIV's aims in order to evaluate each approach and to highlight the areas of opportunities of our work. This comparison is divided into the same five criteria defined in Section 2.1.

Table 2

Comparison of approaches oriented to the Knowledge Discovery and the OntoOAIV's aims

Approach	Description	Status	SW	OP	KD	UP	V
IntelliSearch [37]	Implementation of a semantic web search engine based on semantic relatedness	Active (commercial)	X		X		
Evi [20]	Mobile application based on SW that incorporates a question-answering engine	Active (non commercial)	X		X		
AquaLog [33]	Portable question-answering system for SW	Active (non commercial)	X		X		
Yummly [55, 30]	Semantic web search engine for food, cooking, and recipes	Active (open code)	X		X	X	
Semantic Search [25]	Application that uses SW to augment the search results based	Active (private)	X		X	X	

	on traditional information retrieval						
SemSearch [32]	Semantic search engine that supports complex queries in terms of multiple keywords	Active (open code)	X	X	X		
Sindice [52]	A semantic search engine of labeled resources with RDF, microformats, microdata, and RDFa	Inactive (it is part of a commercial solution)	X		X		
Freebase [24]	A knowledge base with a platform and an API to access it	Inactive (it was acquired by Google)	X		X		
Knowledge Graph [48]	A knowledge base used by Google to improve its search engine	Active	X		X		

In summary, Table 1 and Table 2 present 21 projects: 12 for libraries and 9 for computational knowledge discovery. Regarding the library approaches, 10 of them address OAI-PMH, a key aspect of our work; however, none uses the Semantic Web technologies and provides the desired Knowledge Discovery in the computational field and in our approach. Likewise, the user's context considered as an input to the KD process and the visualization of graphs generated as an output for the user's query are not included in these projects. Conversely, accounting for the various computational approaches, all contemplate the use of semantic technologies and therefore, yield KD. However, only 2 applications encompass the user's context and even more, 1 handles structured information with OAI-PMH. Again, the graphics-based visualization is not present. As a consequence, OntoOAIV arises precisely with the idea of allowing the Knowledge Discovery and the visual analysis in structured resources under the OAI-PMH specification through the incorporation of semantic technologies and graphs.

3 Description of the Proposal

For the explanation of our proposal, we divided this section into three subsections. The first part describes the methodology and implementation of the proposed semantic approach, emphasizing the usage of the algorithm for merging ontologies. The second subsection explains the adaptation of the student's representation model. While the third part presents the incorporation of the tool for exploiting the OntoOAIV's knowledge base through a visual representation.

3.1 Semantic Approach

This semantic approach is based on the work proposed by Becerril et al. [5]. Thus, we will present a brief description of this extended proposal, named OntoOAIV, a semantic approach to context-aware resource discovery and visualization over scholarly content structured with OAI-PMH. The extended methodology of this proposal is shown in Figure 1.

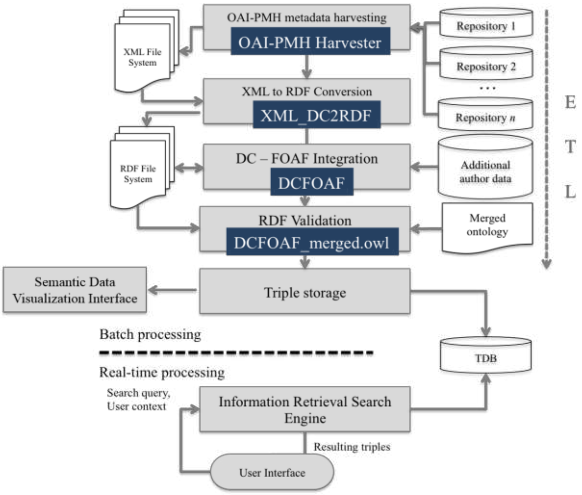


Figure 1
OntoOAIV's methodology

The first process involved in this approach is metadata harvesting of information resources available through repositories that implement the OAI-PMH protocol. OAI-PMH specifies the output information in XML file serialization format with DC; therefore, the output in XML is submitted to a transformation into RDF format. After that, those RDF files enter in the authorship information enrichment process, which is performed by following the specification of FOAF and by relating data retrieved from DBPedia, particularly from the Person-Data dataset [12]. As a result, for each harvested item, the creator tag – corresponding to the author or co-authors of the resource – is enriched. The RDF type (dc:creator) is specified as http://xmlns.com/foaf/0.1/Person, as well as the name (foaf:name), surname (foaf:givenName, foaf:surname), related person or co-author (foaf:knows), interest (foaf:topic interest), description of the creator (occupation, degree, among others) (dc:description), and birth date (onto:birthDate).

Subsequently, OntoOAIV validates the resulting knowledge using an ontology obtained from the merging of the DC and FOAF ontologies; such process is explained in Section 3.1.1. Later, the knowledge base gathered and validated is located as a repository, which can be exploited through a visualization interface (see Section 3.3). Finally, an information retrieval based on inference is performed, being this knowledge discovery the service provided to the end-user. This discovery process requires two inputs: a search query and the contextual user information. Such context is modeled through an ontological representation described in Section 3.2.

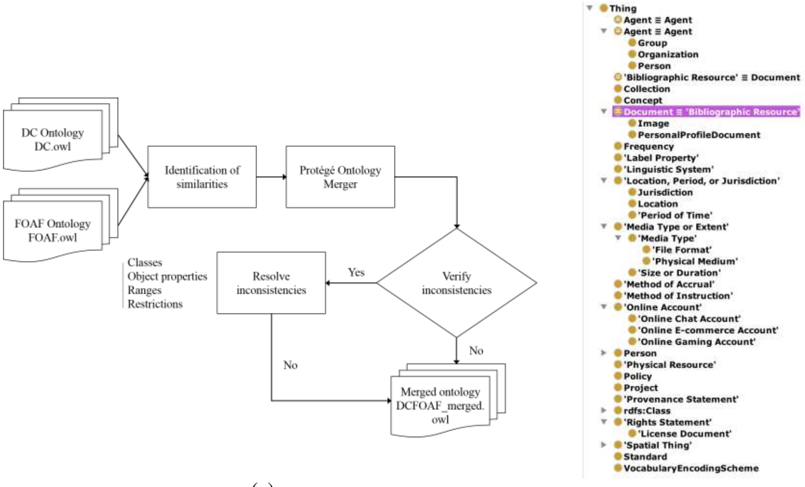
In summary, this approach is framed in the logic of the ETL (Extraction, Transformation, and Load) process, since the information is collected and organized, and then transformed into RDF, enriched and validated using ontologies, and finally loaded as triplets to a triple-store. Furthermore, this methodology is designed to add from 1 to *n* repositories, provided they fulfill the OAI-PMH specification. The technological contributions that OntoOAIV provides are a DCFOAF integrator, a DCFOAF_merged.owl ontology, an OntoOAIEstudiante.owl ontology and a visualization tool.

3.1.1 The Validation: Algorithm for Merging Ontologies

The purpose of the merging process is to generate a knowledge representation of the integration of data from the OAI-PMH repositories (DC) and other information sources, that allow enriching the data of the authors (FOAF), aiming to build a model to verify the consistency of such integration. Hence, the proposed ontological model integrates the namespace maintained by the Dublin Core Metadata Initiative (DCMI) [13], DC ontology constituted of 25 classes, and the FOAF namespace [9], ontology composed of 19 classes. Accordingly, the merged ontology can be used to model the properties of an information resource, such as a book, an article, among others, with its author or authors. For example, a researcher is an author of a publication (dc:creator) and at the same time is a person (foaf:person). Likewise, this ontology represents the authorship relation between one publication and one researcher as well as the co-authorship of one researcher with another, who shares a publication in common.

A merging of ontologies cannot be completely solved automatically due to a variety of factors. For instance, an insufficient specification of an ontology, which obstructs to find similarities with another ontology, thus, a merging is carried out manually or semi-automatically; where, a tool helps to find possible relations between items of different ontologies and an expert confirms these relations based on the ontology components' natural language description and his/her common sense. For this work, the semi-automatic approach is used since through Protégé [50] and its "Refactor>Merge Ontologies" function is carried out the merging of the two ontologies: DC and FOAF. Nevertheless, as Ameen et al. [4] mention, this automatic integration does not solve the inconsistencies generated after the process. In consequence, an adjustment is applied to the resulting ontology using the merging algorithm proposed by Ameen et al. [4]. This algorithm is illustrated in Figure 2.

Therefore, the automatic process in Protégé [50] identified (in both ontologies) and merged the Agent and rdfs:Class classes, as a result of their identical names. However, for the BibliographicResource (DC ontology) and Document (FOAF ontology) classes – equivalent class of CreativeWork –, the merging was performed manually, because there was not a coincidence in their names, even though their definitions are equivalent. In this way, the resulting ontology presents the class hierarchy shown in Figure 2.



(a)

(b)

Figure 2

(a) Algorithm to merge ontologies based on the work of Ameen et al. [4] and (b) Class hierarchy of the merged ontology

Subsequently, the object properties were analyzed in order to identify semantic coincidences, as well as to verify and to match their ranges and domains. Regarding this analysis, the Creator class from DC and Maker class from FOAF were declared as similar. On the other hand, the knows and topic interest properties from FOAF also were studied due to its relevance in the description of the resources in OntoOAIV – foaf:knows represents the people with whom a relation is given, for example, a co-authorship; and foaf:topic interest describes the people's interests (authors/co-authors) –. Afterward, the data properties were addressed, where the title property was manually modified to establish a relation between Title (DC) and title (FOAF) since they present a variation in their names (capital letter). At the end, the merged ontology, called DCFOAF_merged.owl, is verified using a reasoner. This verification proves that the resulting ontology is consistent, considering that the asserted model and the inferred model are similar. Therefore, this ontology can be used to check the consistency of the RDF/XML files obtained in the harvesting, transformation, and integration phases of OntoOAIV. The metrics of DCFOAF_merged.owl are presented below: 42 Classes, 66 Object properties and 40 Data properties.

3.2 The User's Context: Student Profile Ontology

For this proposal, the user's context is conceptualized as the ability to perceive information about the user's environment. This definition arises in order to infer non-explicit facts about the user for later providing results more consistent according to such information. Therefore, it is necessary to carry out a modeling of the user's profile. According to the foregoing, one of our goals is to build, to populate, and to use an ontology related to the scientific-academic nature of the information addressed in this approach, such as publications' data, in order to deal with the user context. Since this context can include profiles of researchers, teachers, and students, the scope of this work is limited to modeling one of the profiles mentioned above: the student's profile.

An approach to modeling a student using an ontology was proposed by Panagiotopoulos et al. [39]. This ontology specifies four main classes, named “Student”, “StudentCourseInformation”, “StudentCurrentActivity”, and “StudentPersonalInformation”. A relevant aspect of this ontology is the inclusion of the student's personal information in addition to the student's academic information since this personal information provides mostly static and permanent student information. However, the ontology proposed by Panagiotopoulos et al. [39] was not available on the Web and was designed with some different features to our approach. Therefore, a new ontology is built based on the student ontology proposed by Panagiotopoulos et al. [39]. As a consequence, an ontology called OntoOAIEstudiante was developed, which represents the user's context from the information that is previously known and is defined as a static entry. It is noteworthy that our student representation does not contemplate information about interaction with the application.

Thereby, the proposed student ontology for our approach considers four classes: "Estudiante" represents any student; "Cursos" describes the subjects, the school, the program, and the program level (BA, MA, Ph.D.) at which the student is enrolled; "ActividadActual" provides information about the student current enrolled period, previous experience, course goals, modules, and period; and "InformaciónPersonal" gives information about the student's accessibility, demographics, and motivation.

The user profile represented in the "OntoOAIEstudiante" ontology is the input that provides the user's context to the OntoOAIV's inference engine.

3.3 The Visualization: Semantic Data Visualization Tool

In OntoOAIV, the incorporation of a layer for data visualization is considered because a large amount of information was gathered and integrated for our experiments, in order to provide any user a quick, simple, and easy interpretation of this information. This fact is justified by the idea expressed by Machová et al. [34]: "Information in an ontology is usually too extensive to be visualized globally in its whole complexity".

The integrated visualization approach into OntoOAIV was developed by Alvarado-Uribe et al. [3]. This work was proposed with the aim of exploiting semantic information through an individual and collective visual representation of resources, using an SPARQL endpoint. The main features of the tool are addressed below: a based keyword search engine (these terms denote the resources that constitute the knowledge base specified by the SPARQL endpoint), an SPARQL endpoint (the only and main entry to this tool), and a visual representation (a set of graphs, such as bar charts, a heat map, and a location map, as well as text).

Due to the location information (e.g. latitude and longitude) is not included in the OntoOAIV's knowledge base (described in Table 3), we modified the tool in order to obtain graphs more relevant for our goals, such as to define on the homepage as main properties to dc:type and dc:title.

4 Experiments

In order to test our approach, we have carried out three experimental phases. The first phase is for validating our knowledge base for then, exploiting it in the second and third phases through a query formulated by a user. Therefore, in Section 4.2, the knowledge discovery process is described while in Section 4.3, the visualization process is explained.

4.1 Data Collection and Knowledge Base

For validating the proposed model, we used the RUDAR [45] and Redalyc [44] repositories, which implement OAI-PMH, as our two data providers. In addition, we merged the FOAF and DC ontologies, as well as the "PersonData" dataset from DBPedia [12], in order to enrich and to validate our knowledge base. To finally obtain an enriched, validated, and stored knowledge base (in triples).

4.2 Selective Knowledge Discovery Process: Use Case

The user's profile defined and used in this experiment belongs to the student "Margret Fintz", identified by "univ:DL9510078". This student is enrolled in a "Master’s" degree program in "Pedagogy Educational Studies" at "Deutsche Universität", taking the "Humanism and Pedagogy" subject. Margret is interested in "Social apprenticeship" and "Johan Friedrich Herbart". In the same way, she specifies that "German", "English", and "Spanish" are her preferred languages, "article" and "thesis" are her preferred types of information resources, and "DE" is her demographic data. Regarding the application example is described that

Margret searches for "non-violence". Consequently, our approach provides the "Education for Active Non-Violence" thesis identified by "oai:rudar.ruc.dk:1800/2990", proposed by "Uski, Juha Janne Olavi", and published on "2008-01-17", based on the search term, as well as on Margret's profile. Since this resource was also located by coincidence with the program in which Margret is enrolled (Pedagogy Educational Studies), this result is considered as relevant for the student, which leads to the deductive reasoning process. The reasoning process takes as input the resource identifier (oai:rudar.ruc.dk:1800/2990), providing as a first deduction that Juha Janne Olavi Uski and Stephen Carney (who is specified as a contributor of this resource) are authors of this thesis; therefore, they are defined as interesting for Margret. Thus, a second deduction arises by considering the works belonging to these authors as pertinent for her. Subsequently, if these works are relevant, then the co-authors of these resources are valuable to her. Finally, another deduction is given when the works of these co-authors are determined as significant to Margret. As a result, a dataset, composed of 129 resources, represents the output of Margret's search.

4.3 Implementation of the Visualization Tool: Use Case

Because the only entry for this tool is an SPARQL endpoint, we had to generate an endpoint of the OntoOAIV's knowledge base. Hence, we used Dydra [28] to produce this SPARQL endpoint for our knowledge base; however, such repository only counts on a significant amount of resources (100), for our purposes, since the size of the original knowledge base has almost half a terabyte of information. Afterward, the generated SPARQL endpoint is provided to the tool's interface, which provides a view of the resources contained in this repository. Once this connection is established, the "Thesis" term is chosen to perform a search. In consequence, the tool provides the visualization of 100 resources of this type in both graphics and text.

5 Results and Discussion

As a first result, OntoOAIV provides a knowledge base composed of 7,917,081 facts from the academic resources structured with OAI-PMH, described in Table 3. This base contains information about 968,903 authors, 60,354 out of which were enriched. It is relevant to examine the composition of the knowledge base in order to know the data that were finally stored in the triple-store since some of them were discarded for errors found. For example, only 60,354 authors, out of the 60,927 identified, were enriched. Regarding resources, the knowledge base is composed of the 395,419 resources result from the conversion process to RDF, where only 394,776 resources have an identifier, 379,966 have a source, and

394,775 have a publication date. A curious fact is that 1,600,540 dc:subject were found, i.e., an average of 4.04 keywords per resource representing the topics that address each of them. Unfortunately, in the subject description, there is a great diversity of nomenclatures, formats, and languages used in the repositories, which can prevent to identify relationships and to have a greater semantics. On the other hand, the lack of identifiers (URIs - Universal Resource Identifiers) for resources (books, articles, thesis, authors, among others) on the Web leads to ambiguity and homonymy issues, which can produce a difficult, incomplete, and inconsistent integration process.

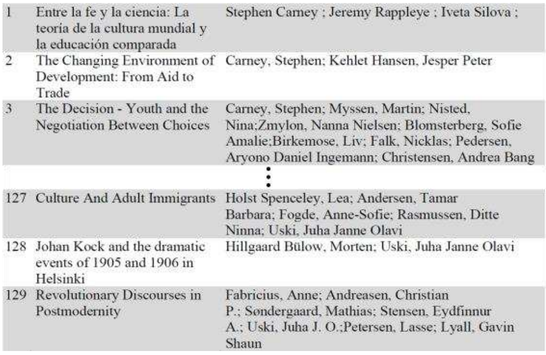
Table 3

OntoOAIV's knowledge base

Property	Total of Triples
http://xmlns.com/foaf/0.1/Person	60,354
http://purl.org/dc/elements/1.1/identifier	394,776
http://purl.org/dc/elements/1.1/source	379,966
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	60,354
http://purl.org/dc/elements/1.1/date	394,775
http://xmlns.com/foaf/0.1/name	60,354
http://purl.org/dc/elements/1.1/format	385,975
http://purl.org/dc/elements/1.1/description	361,577
http://purl.org/dc/terms/modified	394,772
http://dbpedia.org/ontology/birthDate	60,354
http://purl.org/dc/elements/1.1/type	405,463
http://purl.org/dc/elements/1.1/title	394,776
http://purl.org/dc/elements/1.1/publisher	382,412
http://purl.org/dc/terms/isPartOf	394,780
http://purl.org/dc/elements/1.1/subject	1,600,540
http://purl.org/dc/elements/1.1/rights	379,966
http://xmlns.com/foaf/0.1/surname	60,354
http://xmlns.com/foaf/0.1/givenName	60,354
http://purl.org/dc/elements/1.1/relation	381,980
http://purl.org/dc/elements/1.1/creator	968,903
http://purl.org/dc/elements/1.1/language	394,650

Regarding the results of the selective knowledge discovery process, an extract of this output in Figure 3 is provided since all of the results of this process cannot be presented for lack of space. It is noteworthy here that this process complies with our main goal: the knowledge discovery using information implemented with OAI-PMH and enriched with semantic technologies. This is verified when from 1 result is produced up to 129 outputs, all relevant to the user. One aspect that must be addressed is the update of the user's context because to in this work, this context is provided during the design phase and cannot be modified during execution.

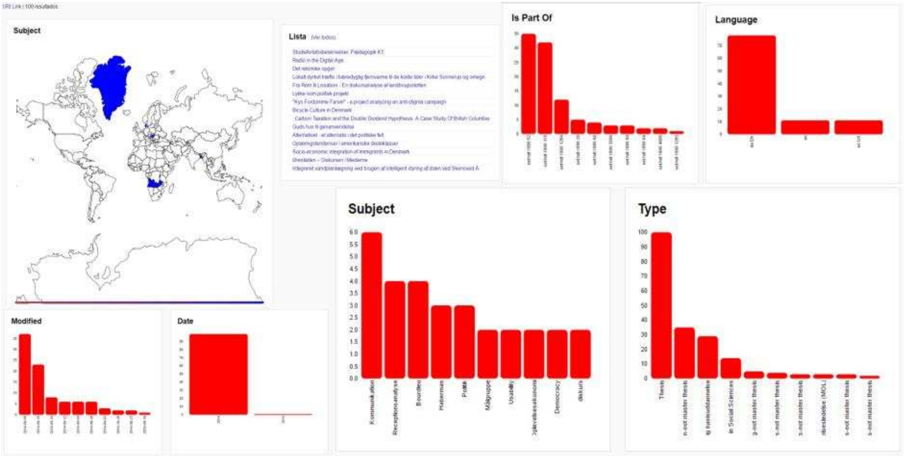
For the example of the visualization of the information contained in the OntoOAIV's knowledge base, Figure 4 is included. This visualization provides data clusters that allow a quick and easy exploration of the information. For example, it shows that all resources are Thesis published in 2014 and 2015, and written in "da-DK" (Danish (Denmark)), "en" (English), and "en US" (English (United States)). Although some results can be appreciated, the heat map reflects the lack of information as a result of only taking a sample of the original repository for this experiment. The endpoint used is https://dydra.com/joanna-au/oai-pmh-repository/sparql



1	Entre la fe y la ciencia: La teoría de la cultura mundial y la educación comparada	Stephen Carney ; Jeremy Rappleye ; Iveta Silova ;
2	The Changing Environment of Development: From Aid to Trade	Carney, Stephen; Kehlet Hansen, Jesper Peter
3	The Decision - Youth and the Negotiation Between Choices	Carney, Stephen; Myssen, Martin; Nisted, Nina;Zmylon, Nanna Nielsen; Blomsterberg, Sofie Amalie;Birkemose, Liv; Falk, Nicklas; Pedersen, Aryono Daniel Ingemann; Christensen, Andrea Bang
	⋮	
127	Culture And Adult Immigrants	Holst Spenceley, Lea; Andersen, Tamar Barbara; Fogde, Anne-Sofie; Rasmussen, Ditte Ninna; Uski, Juha Janne Olavi
128	Johan Kock and the dramatic events of 1905 and 1906 in Helsinki	Hillgaard Bülow, Morten; Uski, Juha Janne Olavi
129	Revolutionary Discourses in Postmodernity	Fabricius, Anne; Andreasen, Christian P.; Søndergaard, Mathias; Stensen, Eydfinnur A.; Uski, Juha J. O.;Petersen, Lasse; Lyall, Gavin Shaun

Figure 3

Example of the results found for the Margret's search



Studieforløbsbeskrivelser: Pædagogik KT
Radio in the Digital Age
Det retoriske opgør
Lokalt dyrket traffic i bæredygtig fremvarme til de kolde tider i Kirke Sonnerup og omegn
Fra Rom til Location - En diskursanalyse af [illegible]
Lykke som politisk projekt
"Kys Fordomme Farvel" - a project analyzing an anti-stigma campaign
Bicycle Culture in Denmark
Carbon Taxation and the Double Dividend Hypothesis: A Case Study Of British Columbia
Guds hus til genanvendelse
Alternativet - et alternativ i det politiske felt
Opdragelsestendenser i amerikanske skoleklasser
Socio-economic integration of immigrants in Denmark
[illegible] – Diskursen i Medierne
Integreret vandplantevækst ved brugen af intelligent styring af [illegible] ved Skærbæk A

Figure 4

Visualization of the search performed using the "Thesis" term

This visualization also confirms that it is important to count on properties related to the location, such as latitude and longitude, aiming to populate the heat map, one of the most representative charts of the tool; otherwise, this graph would not be useful. Alternatively, because the tool depends on the endpoint provided, restrictions such as availability, maintenance and format can avoid the correct functioning of this tool. Consequently, if the endpoint's information repository is not updated, the application will not return useful information to users.

Conclusions and Future Work

In this paper, OntoOAIV is introduced aiming to verify that the incorporation of the Semantic Web technologies provides the interoperability that the Open Access platforms structured with OAI-PMH need to address for the selective knowledge discovery and in consequence, to enable students the recovering of resources not explicitly requested by them but that result potentially useful. Hence, it is proved based on the results that the incorporation of semantic technologies (algorithm for merging ontologies and ontology for providing the student's context) allows the knowledge discovery in structured information with OAI-PMH. Therefore, OntoOAIV is an approach that allows dealing with the interoperability issue presented by OAI-PMH, achieving satisfactory results in the knowledge discovery despite the issues and limitations faced (such as the validation of the merged ontology). Regarding the visualization tool, the inclusion of graphs in the query's results is an important feature that differentiates our work from those reviewed in the literature. This incorporation is relevant because the graphs allow the user to analyze large amounts of information through the simplified and understandable presentation of the data, rather than just getting a text.

OntoOAIV could be extended to take advantage of the information sources of Linked Open Data as other input for our approach. In addition, the proposal can be enhanced with the use of controlled vocabularies and/or multilingual ontologies to retrieve information in several languages. Regarding the user profile, the incorporation of a mechanism capable of learning and updating the information contained in this is established as an improvement, i.e., the inclusion of a dynamic user profile. Concerning the visualization approach, it is recommended to improve the clustering algorithm, resulting in more enriched graphs. For example, finding matches in different languages.

Acknowledgement

Joanna Alvarado-Uribe is currently supported by a CONACYT studentship.

References

[1] Allemang, D., Hendler, J.: Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL. Morgan Kaufmann, 2nd edn. (May 2011)

[2] Alok Jha: Open access is the future of academic publishing, says Finch report. [Online]. Available: https://www.theguardian.com/science/2012/jun/19/open-access-academic-publishing-finch-report (current November 2016)

[3] Alvarado-Uribe, J., González-Mendoza, M., Hernández-Gress, N., Escobar-Ruiz, C.E., Hernández-Camacho, M.U.: Una herramienta visual para la búsqueda semántica rdf. Research in Computing Science 95, 9-22 (2015)

[4] Ameen, A., Rahman Khan, K.U., Rani, B.P.: Semi-automatic merging of ontologies using protégé. International Journal of Computer Applications 85(12), 35-42 (January 2014)

[5] Becerril, G.A., Lozano, E.R., Molina, E.J.M.: Enfoque semántico para el descubrimiento de recursos sensible al contexto sobre contenidos académicos estructurados con oai-pmh. Computación y Sistemas 20(1), 127-142 (2016)

[6] BiblioCommons: Bibliocore. [Online]. Available: http://www.bibliocommons.com/products/bibliocore (current November 2016)

[7] Blacklight: Blacklight. [Online]. Available: http://projectblacklight.org/ (current November 2016)

[8] Breeding, M.: The future of library resource discovery: A white paper commissioned by the niso discovery to delivery (d2d) topic committee. White paper, National Information Standards Organization (NISO), 3600 Clipper Mill Road, Suite 302. Baltimore, MD 21211 (February 2015)

[9] Brickley, D., Miller, L.: Foaf vocabulary specification 0.99. [Online]. Available: http://xmlns.com/foaf/spec/ (current September 2016)

[10] Brown, P.O., Cabell, D., Chakravarti, A., Cohen, B., Delamothe, T., Eisen, M., Grivell, L., Guédon, J.C., Hawley, R.S., Johnson, R.K., Kirschner, M.W., Lipman, D., Lutzker, A.P., Marincola, E., Roberts, R.J., Rubin, G.M., Schloegl, R., Siegel, V., So, A.D., Suber, P., Varmus, H.E., Velterop, J., Walport, M.J., Watson, L.: Bethesda statement on open access publishing. [Online]. Available: http://legacy.earlham.edu/~peters/fos/bethesda.htm (current October 2016)

[11] Chan, L., Cuplinskas, D., Eisen, M., Friend, F., Genova, Y., Guédon, J.C., Hagemann, M., Harnad, S., Johnson, R., Kupryte, R., Manna, M.L., Rév, I., Segbert, M., de Souza, S., Suber, P., Velterop, J.: Budapest open access initiative. [Online]. Available: http://www.budapestopenaccessinitiative.org/ (current October 2016)

[12] DBpedia: The dbpedia data set (2014). [Online]. Available: http://wiki.dbpedia.org/services-resources/datasets/dbpedia-data-set-2014 (current September 2016)

[13] DCMI: Dcmi metadata terms. [Online]. Available: http://dublincore.org/documents/2012/06/14/dcmi-terms/ (current September 2016)

[14] DCMI: Dublin core metadata element set, version 1.1. [Online]. Available: http://dublincore.org/documents/dces/ (current September 2016)

[15] DCMI: Metadata basics. [Online]. Available: http://dublincore.org/metadata-basics/ (current September 2016)

[16] DOAJ: Directory of open access journals (doaj). [Online]. Available: https://doaj.org/ (current October 2016)

[17] DuraSpace: Dspace. [Online]. Available: http://www.duraspace.org/ (current October 2016)

[18] EBSCO: Ebsco discovery service. [Online]. Available: https://www.ebscohost.com/discovery (current November 2016)

[19] Eprints: Registry of open access repositories. [Online]. Available: http://roar.eprints.org/view/type/ (current February 2016)

[20] Evi: Evi an amazon company. [Online]. Available: https://www.evi.com (current May 2016)

[21] Ex Libris: Primo discovery and delivery. [Online]. Available: http://www.exlibrisgroup.com/category/PrimoOverview (current November 2016)

[22] eXtensible Catalog Organization: extensible catalog. [Online]. Available: https://www.extensiblecatalog.org/ (current November 2016)

[23] Frawley, W.J., Piatetsky-Shapiro, G., Matheus, C.J.: Knowledge discovery in databases: An overview. AI Magazine 13(3), 57-70 (September 1992), american Association for Artificial Intelligence

[24] Google: Freebase data dumps. [Online]. Available: https://developers.google.com/freebase/#freebase-wikidata-mappings (current May 2016)

[25] Guha, R., McCool, R.: Tap: A semantic web platform. Computer Networks 42(5), 557-577 (August 2003), Elsevier

[26] Gutierrez, C., Hurtado, C., Mendelzon, A.O.: Foundations of semantic web databases. In: Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. pp. 95-106. PODS '04, ACM, New York, NY, USA (2004)

[27] Innovative Interfaces: Encore discovery solution. [Online]. Available: https://www.iii.com/products/sierra/encore (current November 2016)

[28] James Anderson and Arto Bendiken: Dydra. [Online]. Available: https://dydra.com/ (current November 2016)

[29] Kessler, C., dAquin, M., Dietze, S.: Linked data for science and education. Semantic Web 4(1), 1-2 (2013)

[30] Kessler, W.: Semantic search. [Online]. Available: http://wiltrud.hwro.de/teaching/semweb15w/supplements/4S_SemanticSearch.handout.pdf (current May 2016)

[31] Lagoze, C., Van de Sompel, H.: The open archives initiative: Building a low-barrier interoperability framework. In: Proceedings of the 1st ACM/IEEE-

CS Joint Conference on Digital Libraries. pp. 54-62. JCDL '01, ACM, New York, NY, USA (2001)

[32] Lei, Y., Uren, V., Motta, E.: Managing Knowledge in a World of Networks: 15th International Conference, EKAW 2006, Poděbrady, Czech Republic, October 2-6, 2006. Proceedings, Lecture Notes in Computer Science, vol. 4248, chap. SemSearch: A Search Engine for the Semantic Web, pp. 238-245. Springer Berlin Heidelberg (2006)

[33] Lopez, V., Pasin, M., Motta, E.: The Semantic Web: Research and Applications: Second European Semantic Web Conference, ESWC 2005, Heraklion, Crete, Greece, May 29-June 1, 2005. Proceedings, Lecture Notes in Computer Science, vol. 3532, chap. AquaLog: An Ontology-Portable Question Answering System for the Semantic Web, pp. 546-562. Springer Berlin Heidelberg, Berlin, Heidelberg (May-June 2005)

[34] Machová, K., Vrana, J., Mach, M., Sinčák, P.: Ontology evaluation based on the visualization methods, context and summaries. Acta Polytechnica Hungarica 13(4), 53-76 (2016), BUDAPEST TECH BECSI UT 96-B, BUDAPEST, H-1034, HUNGARY

[35] Martinez-Villaseñor, M.d.L., Gonzalez-Mendoza, M., Hernandez-Gress, N.: Towards a ubiquitous user model for profile sharing and reuse. Sensors 12(10), 13249-13283 (2012)

[36] Max-Planck-Gesellschaft: Berlin declaration on open access to knowledge in the sciences and humanities. [Online]. Available: https://openaccess.mpg.de/Berlin-Declaration (current October 2016)

[37] Mehta, A., Makkar, P., Palande, S., Wankhede, S.B.: Semantic web search engine. International Journal of Engineering Research and Technology 4(4), 687-691 (April 2015)

[38] OCLC: Worldcat discovery. [Online]. Available: https://www.oclc.org/worldcat-discovery.en.html (current November 2016)

[39] Panagiotopoulos, I., Kalou, A., Pierrakeas, C., Kameas, A.: Artificial Intelligence Applications and Innovations: 8th IFIP WG 12.5 International Conference, AIAI 2012, Halkidiki, Greece, September 27-30, 2012, Proceedings, Part I, IFIP Advances in Information and Communication Technology, vol. 381, chap. An Ontology-Based Model for Student Representation in Intelligent Tutoring Systems for Distance Learning, pp. 296-305. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)

[40] Patel-Schneider, P.F., Horrocks, I.: Position paper: A comparison of two modelling paradigms in the semantic web. In: Proceedings of the 15th International Conference on World Wide Web. pp. 3-12. WWW '06, ACM, New York, NY, USA (2006)

[41] PKP: Open journal systems. [Online]. Available: https://pkp.sfu.ca/ojs/ (current October 2016)

[42] ProQuest: Aquabrowser. [Online]. Available: http://www.proquest.com/products-services/AquaBrowser.html (current November 2016)

[43] ProQuest: The summon service. [Online]. Available: http://www.proquest.com/products-services/The-Summon-Service.html (current November 2016)

[44] Redalyc: Sistema de información científica redalyc - red de revistas científicas de américa latina y el caribe, españa y portugal. [Online]. Available: http://www.redalyc.org (current May 2016)

[45] RUDAR: Rudar - roskilde university digital archive. [Online]. Available: http://rudar.ruc.dk/ (current May 2016)

[46] Schonfeld, R.C.: Does discovery still happen in the library? roles and strategies for a shifting reality. Report, Ithaka S+R, New York, NY (2014), http://www.sr.ithaka.org/wp-content/mig/files/SR_Briefing_Discovery_20140924_0.pdf

[47] Shadbolt, N., Hall, W., Berners-Lee, T.: The semantic web revisited. IEEE Intelligent Systems 21(3), 96-101 (May 2006), iEEE

[48] Singhal, A.: Introducing the knowledge graph: things, not strings. [Online]. Available: https://googleblog.blogspot.mx/2012/05/introducing-knowledge-graph-things-not.html (current May 2016)

[49] SirsiDynix: Bluecloud pac. [Online]. Available: http://www.sirsidynix.com/products/bluecloud-pac (current November 2016)

[50] Stanford University: Protégé. [Online]. Available: http://protege.stanford.edu/ (current November 2016)

[51] Subirats-Coll, I.: Seven things you should know about linked data. COAR Repository Observatory (2) (2014)

[52] Tummarello, G., Delbru, R., Oren, E.: The SemanticWeb: 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007. Proceedings, Lecture Notes in Computer Science, vol. 4825, chap. Sindice.com: Weaving the Open Linked Data, pp. 552-565. Springer Berlin Heidelberg (2007)

[53] University of Pennsylvania: Franklin. [Online]. Available: http://franklin.library.upenn.edu/index.html (current November 2016)

[54] Villanova University's Falvey Memorial Library: VuFind. [Online]. Available: http://vufind-org.github.io/vufind/ (current November 2016)

[55] Yummly: Yummly api documentation. [Online]. Available: https://developer.yummly.com/documentation (current May 2016)

An Edge Detection Method using a Fuzzy Ensemble Approach

Ernesto Moya-Albor, Hiram Ponce, Jorge Brieva[*]

Universidad Panamericana, Campus México, Facultad de Ingeniería, Augusto Rodin 498, México, Ciudad de México, 03920, México

emoya@up.edu.mx, hponce@up.edu.mx, jbrieva@up.edu.mx

[*] Corresponding author

Edge detection is one of the most important low level steps in image processing. In this work we propose a fuzzy ensemble based method for edge detection including a fuzzy c-means (FCM) approach to define the input membership functions of the fuzzy inference system (FIS). We tested the performance of the method using a public database with ground truth. Also, we compared our proposal with classical and other fuzzy based methods, using F-measure curves and the precision metric. We conducted experiments with different levels of salt & pepper noise to evaluate the performance of the edge detectors. The metrics illustrate the robustness of the choice of the threshold in the binarization step using this fuzzy ensemble method. In noisy conditions, the proposed method works better than other fuzzy approaches. Comparative results validated that our proposal overcomes traditional techniques.

Keywords: edge detection; fuzzy inference system; fuzzy clustering; noise, image processing

1 Introduction

Edge detection is one of the most important low level steps in image processing. It is used in several high level analysis as features identification, register and motion estimation [1, 2]. From a signal processing point of view, the process consists of the detection of abrupt local changes in intensity, texture or luminosity [3]. The expected final result is a binary image corresponding to the pixels labeled as point contours. This procedure includes two steps: an enhancement of the image contours and a decision step to determine if a pixel is a contour or not on the basis of some local or regional information. This second step remains important and it usually can be done by thresholding [4].

Many techniques have been presented in the literature to solve the edge detection problem. In this paper, we will present the most representatives. Well known traditional techniques use the convolution of derivative-based linear-time invariant (LTI) filters with the image, such as: Sobel [5], Roberts [6] and Prewitt [7] detectors. Ulipinar et al. [8] use zero crossings of the Laplacian of Gaussian operator. Canny uses derivative-based filters adding a restriction of gradient direction to eliminate the non-coherent contours [9]. These methods are not robust in noisy images [10] and they remain very dependent of the threshold choice in the binarization step. Also, mathematical morphology approaches are widely used in the segmentation and edge detection problems, since they work very well in the presence of high contrast structures. In the case of noisy images and low contrast structures, some works which attack this problem are presented in the literature. For example, Wang et al. [11] proposed a multi-scale and multi-form method. Jiang et al. [12] developed a mask based on noise filtering. A combination of morphological filtering and the Laplacian operator were proposed by [13], and a histogram based edge detector was carried out by Krishnamurthy et al. [14]. The main weakness of these approaches is the dependency of the choice on the form and scale of the structural element.

Edge detectors using intelligent approaches have been widely used, those using artificial neural networks [15], spatial clustering [16] and adaptive neuro-fuzzy systems [17]. It is well known the merits of these kinds of methods, but the learning step is very dependent of the application. In particular, fuzzy logic has been employed for edge detection [18, 19, 20]. It consists of the definition of a set of rules and membership functions to associate a possibility of true detection to the pixel. In [18], authors proposed an edge detection algorithm based on fuzzy rules to estimate the edge strength, and a threshold is estimated using an optimization algorithm. Setarehdan et al. proposed a fuzzy temporal and multi-scale method applied to a sequence of cardiac images [21]. Moreover, in the presence of noise, the edge detection task becomes more difficult due to the presence of abrupt changes in intensity not corresponding to edges belonging to objects in the image; but fuzzy based techniques can deal with the presence of noise due to its uncertainty property. For instance, Haq et al. [22] proposed a fuzzy logic based edge detection in smooth and noisy clinical images. They employed a $3\ x\ 3$ mask guided by fuzzy rules set. The mask was formed by the differences of gray level between the center pixel in the mask and each one of the neighbors.

In this work, we propose an edge detection algorithm for gray scale images based on a fuzzy ensemble of both fuzzy inference system and fuzzy c-means clustering. The aim of this method is to exploit global information of an image and use it locally, by a mask technique, to detect edges. The input membership functions are defined by a variable region of uncertainty and an automatic adaptation method to characterize their shapes. Moreover, this method allows to control the mask definition. In addition, an evaluation methodology is proposed and tested for a set of possible thresholds. Different levels of salt & pepper noise are used to establish the robustness of our proposal.

The remaining article is organized as follows. First, the proposed edge detection based on a fuzzy ensemble is presented. Then, an evaluation methodology and the metrics are described. Comparative results with other well-known methods are carried out, as well as the analysis and discussion of the work. Finally, conclusions and future work are discussed.

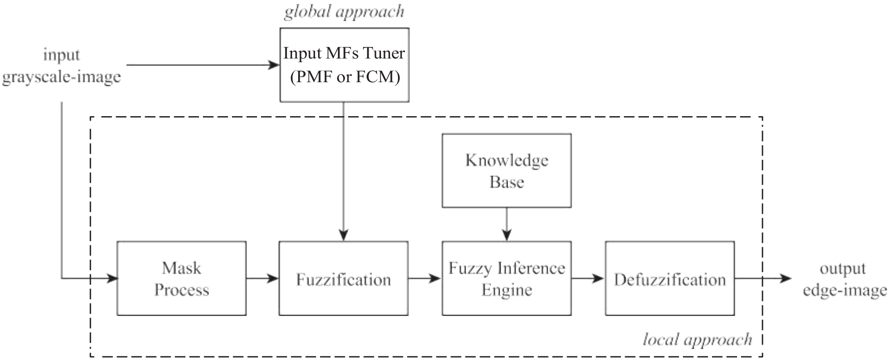


Figure 1
Block diagram of the proposed edge detection algorithm.

2 Methodology

The proposed edge detection algorithm for grayscale images is based on a fuzzy ensemble of both fuzzy inference system (FIS) and fuzzy c-means (FCM) clustering. The aim of this method is to exploit global information of an image and use it locally, by a mask technique, to detect edges. As a result, the proposed edge detector can be adapted to images in terms of the intensity values and the present noise.

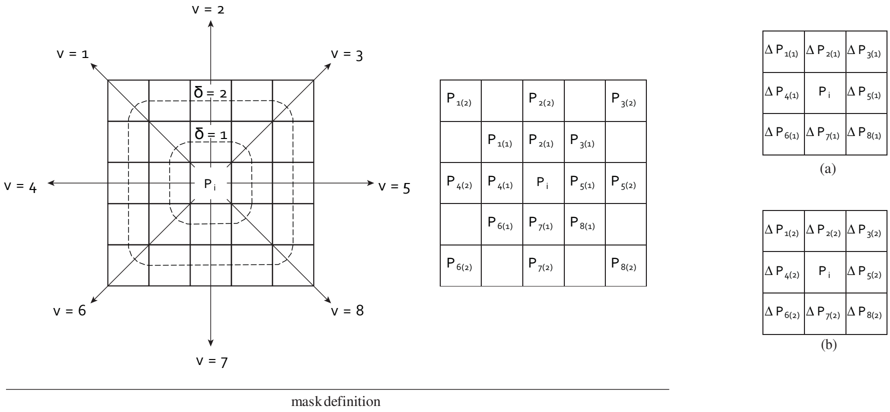
Figure 1 shows the components of our proposal. As shown, the grayscale image inputs to the input membership functions tuner (global approach) aiming to represent the whole image in terms of the space of difference values by using the FCM method. Then, the input image enters to the mask process (local approach) that uses a $3\ x\ 3$ window. At each window, the FIS with the tuned input membership functions computes if the central pixel of the window is an edge or not. At the end of the fuzzy-based mask process, an output edge map is obtained. This proposal is described in detail below.

2.1 Local Approach

Locally, the proposed edge detection algorithm implements a mask technique consisting on a $3\ x\ 3$ window, as shown in Figure 2. Considering that P_i is the central pixel and $P_{v(\delta)}$ for all $v = 1, \ldots, 8$ are the eight neighborhood pixels with radius δ from P_i, then the mask computes the difference values between the central pixel and its neighbors with radius δ as expressed in (1).

$$\Delta P_{v(\delta)} = \left| P_{v(\delta)} - P_i \right| \tag{1}$$

Once the difference values from P_i are calculated, the edge detection algorithm employs a fuzzy inference system to classify if P_i is an edge pixel or not.



$P_{1(2)}$		$P_{2(2)}$		$P_{3(2)}$
	$P_{1(1)}$	$P_{2(1)}$	$P_{3(1)}$	
$P_{4(2)}$	$P_{4(1)}$	P_i	$P_{5(1)}$	$P_{5(2)}$
	$P_{6(1)}$	$P_{7(1)}$	$P_{8(1)}$	
$P_{6(2)}$		$P_{7(2)}$		$P_{8(2)}$

$\Delta P_{1(1)}$	$\Delta P_{2(1)}$	$\Delta P_{3(1)}$
$\Delta P_{4(1)}$	P_i	$\Delta P_{5(1)}$
$\Delta P_{6(1)}$	$\Delta P_{7(1)}$	$\Delta P_{8(1)}$

(a)

$\Delta P_{1(2)}$	$\Delta P_{2(2)}$	$\Delta P_{3(2)}$
$\Delta P_{4(2)}$	P_i	$\Delta P_{5(2)}$
$\Delta P_{6(2)}$	$\Delta P_{7(2)}$	$\Delta P_{8(2)}$

(b)

Figure 2
Mask definition in the proposed edge detection algorithm, where P_i is the central pixel and $P_{v(\delta)}$ are its neighbors v with radius δ. Example of a 3 x 3 window mask: (a) with $\delta = 1$ and (b) with $\delta = 2$.

2.1.1 Fuzzification

The difference values $\Delta P_{v(\delta)}$ for all $v = 1, \ldots, 8$ are used as inputs in the fuzzification step of the FIS. Based on literature [22], each input is partitioned into two fuzzy sets: *small* ($F_{v,\, small}$) and *large* ($F_{v,\, large}$) difference values. Then, both fuzzy sets are represented by membership functions $\mu_{small}(\Delta P_{v(\delta)})$ and $\mu_{large}(\Delta P_{v(\delta)})$.

Actually, the definition of the input membership functions is highly important because they directly influence in the behavior of the FIS [23]. Moreover, these input membership functions are typically tuned in terms of experts in the field [24] or using a priori knowledge of the problem domain [23, 25]. In this proposal, we explore the latter by implementing a global approach that recognizes the whole image and represents it in the input membership functions, as described below.

2.1.2 Fuzzy Inference Engine and Knowledge Base

The next step in the FIS is the fuzzy inference engine. It receives the input membership values computed in the step before and performs an inference operation in the fuzzy space in order to obtain a fuzzy consequence value p_s that describes the belonging of the central pixel P_i to be an edge pixel or not. If the inference operation is described as a fuzzy rule, the s-th fuzzy rule R_s can be expressed as (2); where, ∧-operator represents the T-norm in the fuzzy inference. In this work, the min-operator is selected as the T-norm. To this end, the set of fuzzy rules is based on literature [22] and in additional information extracted from a prior analysis to different configurations of masks detecting edges as summarized in Table 1.

$$R_s\colon if\ \ \wedge_{v=1}^{8} \Delta P_{v(\delta)} \in F_v, then\ p_s \in P_i \tag{2}$$

Table 1
Summary of fuzzy rules detecting an edge pixel

rule	$\Delta P_{1(\delta)}$	$\Delta P_{2(\delta)}$	$\Delta P_{3(\delta)}$	$\Delta P_{4(\delta)}$	$\Delta P_{5(\delta)}$	$\Delta P_{6(\delta)}$	$\Delta P_{7(\delta)}$	$\Delta P_{8(\delta)}$	P_i
1	*large*	*large*	-	-	-	-	-	*small*	*edge*
2	*large*	-	-	*large*	-	-	-	*small*	*edge*
3	-	*large*	*large*	-	-	-	-	*small*	*edge*
4	-	-	-	*large*	-	*large*	-	*small*	*edge*
5	*large*	*large*	-	-	-	-	*small*	-	*edge*
6	*large*	-	-	*large*	-	-	*small*	-	*edge*
7	-	*large*	*large*	-	-	-	*small*	-	*edge*
8	-	-	-	*large*	-	*large*	*small*	-	*edge*
9	*large*	*large*	-	-	*small*	-	-	-	*edge*
10	*large*	-	-	*large*	*small*	-	-	-	*edge*
11	-	*large*	*large*	-	*small*	-	-	-	*edge*
12	-	-	-	*large*	*small*	*large*	-	-	*edge*
13	-	-	*small*	-	-	*large*	*large*	-	*edge*
14	-	-	*small*	-	-	-	*large*	*large*	*edge*
15	-	*small*	-	-	-	*large*	*large*	-	*edge*
16	-	-	*large*	-	*large*	*small*	-	-	*edge*
17	-	-	-	-	*large*	*small*	-	*large*	*edge*
18	-	-	*large*	-	*large*	-	*small*	-	*edge*
19	-	*small*	-	-	-	-	*large*	*large*	*edge*
20	-	-	-	-	*large*	-	*small*	*large*	*edge*
21	-	-	-	-	*small*	*large*	*large*	-	*edge*
22	-	-	*large*	*small*	*large*	-	-	-	*edge*
23	-	-	-	-	*small*	-	*large*	*large*	*edge*
24	-	-	-	*small*	*large*	-	-	*large*	*edge*

2.1.3 Defuzzification

The last step of the FIS considers the defuzzification process. It calculates the crisp output value P_i representing if the central pixel is an edge pixel or not. In this work, we use the center of gravity approach [26] expressed as (3); where $\mu_s(p_t)$ represents the membership value of p_t, an intensity value in the space of edges.

$$P_i = \frac{\sum_t \mu_s(p_t) \cdot p_t}{\sum_t \mu_s(p_t)} \tag{3}$$

At last, the output of the proposed edge detector is partitioned into two fuzzy sets: *non-edge* (P_{ne}) and *edge* (P_e), represented as Gaussian membership functions with means 10 and 245 and with standard deviation 3.5.

2.2 Global Approach

As already said, the global approach of the proposed edge detection algorithm corresponds to define the input membership functions of the FIS aiming to represent the whole image in terms of the space of difference values. This enables to the algorithm getting an overall perspective of the image and the frequency of difference values across it. To accomplish it, we propose two methods: one based on the probability distribution of difference values in the image, and the other based on a fuzzy clustering of the difference values in the image.

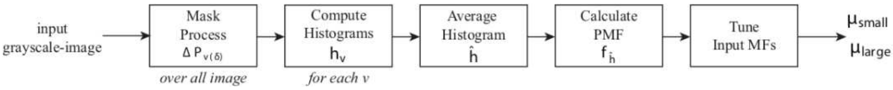


Figure 3
Block diagram of the global approach based on the probability mass function of the difference values

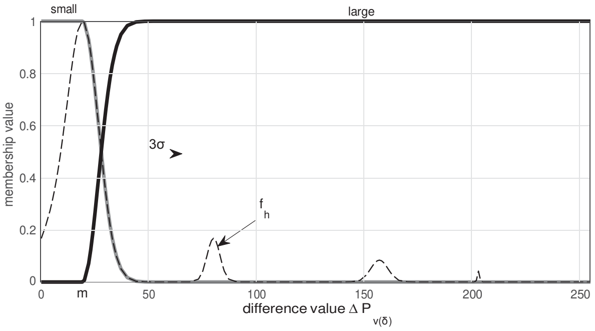


Figure 4
Input membership functions derived from the probability distribution of difference values in the image

2.2.1 Method Based on Probability Distribution

The first method consists on determine the probability mass function (PMF) of the difference values in the whole image. Figure 3 shows the block diagram of this method.

At first, the 3 x 3 mask is used to calculate the difference values $\Delta P_{v(\delta)}$, $v = 1, \dots, 8$, with radius δ from the central pixel P_i. Then, a histogram h_v is computed for each direction v in the neighborhood. The histogram contains the number of bins as the number of integer values has the input $\Delta P_{v(\delta)}$ in the universe of discourse in the FIS. For instance, if the universe of discourse of the difference values ranges from 0 to 255, then the histogram will have 256 bins. Notice that all the histograms will have the same number of bins since the difference values in any direction are computed over the same grayscale values. To this end, we propose to use the number of gray levels in the image as the universe of discourse, and also in the number of bins in histograms. Secondly, an average histogram $\hat{h}$ is computed using all the histograms h_v from $v = 1, \dots, 8$. Then, the probability mass function $f_{\hat{h}}(\Delta P_{v(\delta)})$ of the average of difference values in every direction is expressed as (4).

$$f_{\hat{h}}(\Delta P_{v(\delta)}) = \frac{\hat{h}}{\max(\hat{h})} \quad (4)$$

Considering that homogeneous regions dominate in the image, then the probability mass function will shape at least one peak, i.e. the largest, in the left side representing the most frequent difference values in the image. In fact, this peak represents *small* difference values, and it is close to the mean value m of $f_{\hat{h}}$. Thus, the left side of the peak in $f_{\hat{h}}$ is proposed to be used as part of the input membership function *small*, as denoted in (5); where, σ represents the standard deviation of $f_{\hat{h}}$, and $\Delta P_{max} = \arg\max_{\Delta P_{v(\delta)}} (f_{\hat{h}})$. To this end, the input membership function *large* is the complement of the *small* one, as expressed in (6). Figure 4 shows the input membership functions defined by this method.

$$\mu_{small}\left(\Delta P_{v(\delta)}\right) = \begin{cases} 1 & \Delta P_{v(\delta)} < \Delta P_{max} \\ f_{\hat{h}}(\Delta P_{v(\delta)}) & \Delta P_{max} \leq \Delta P_{v(\delta)} < \Delta P_{max} + 3\sigma \\ 0 & \Delta P_{v(\delta)} \geq \Delta P_{max} + 3\sigma \end{cases} \tag{5}$$

$$\mu_{large}(\Delta P_{v(\delta)}) = 1 - \mu_{small}(\Delta P_{v(\delta)}) \tag{6}$$

2.2.2 Method Based on Fuzzy C-Means

The second method consists on divide the difference values in the whole image using fuzzy c-means clustering technique. Figure 5 shows the block diagram of this method. Fuzzy c-means is a clustering method that groups data using a similarity metric, but also it computes the degree of membership of each data point with respect to each class [27].

In this proposal, FCM clusters the bins (i.e. the difference values $\Delta P_{v(\delta)}$) of the average histogram $\hat{h}$, or alternatively of the probability mass function $f_{\hat{h}}$, by minimizing the modified objective function in (7); where, p_i is the ith difference value in the set $\Omega = \cup_v \Delta P_{v(\delta)}$ of all difference values, c_j is the center of the jth cluster, d is the number of difference values in Ω, k is the number of clusters, μ_{ij} is the degree of membership of p_i in the jth cluster, and $s > 1$ is the exponent of the fuzzy partition matrix for controlling the degree of fuzzy overlap (in this work, we use $s = 2$). It is remarkable to say that FCM constrains the degree of membership for a given data point p_i such that their membership values sum one.

$$J_m = \sum_{i=1}^{d} \sum_{j=1}^{k} {\mu_{ij}}^2 \|p_i - c_i\|^2 \quad , \quad \sum_{j=1}^{k} \mu_{ij} = 1 \tag{7}$$

Using the above method, all the difference values $p_i \in \Omega$ can be clustered in two groups (i.e. $k = 2$): *small* and *large* values. In that sense, we propose to define the input membership functions of the FIS to be the membership values μ_{ij} obtained so far from the FCM algorithm. Thus, the input membership function *small* is proposed to be as (8); where, $\Delta P_{max} = \arg\max_{\Delta P_{v(\delta)}}(\mu_{i1})$ and $\Delta P_{min} = \arg\min_{\Delta P_{v(\delta)}}(\mu_{i1})$ such that $\Delta P_{min} > \Delta P_{max}$ if $c_1 < c_2$ (i.e. c_1 represents the left-class). Again, the input membership function *large* is proposed to be as (6). To this end, a look-up table or a hash function should be defined in order to use the μ_{ij} values as part of the input

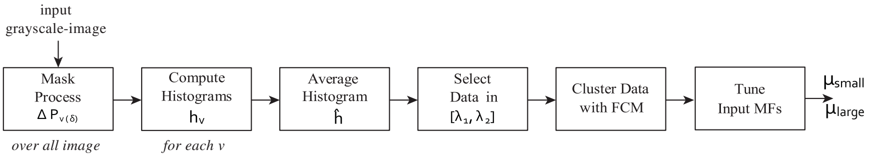


Figure 5
Block diagram of the global approach based on the FCM clustering of the difference values.

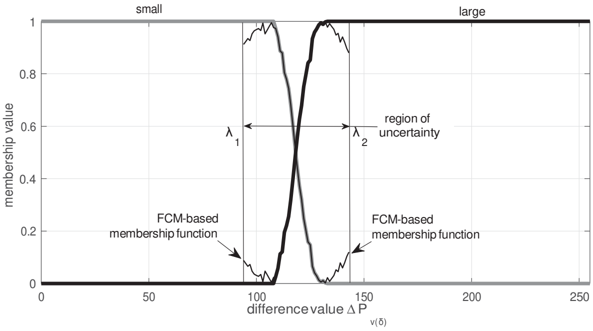


Figure 6
Input membership functions derived from the FCM clustering of difference values in the image

membership functions. Figure 6 shows the input membership functions defined by this method.

$$\mu_{small}\left(\Delta P_{v(\delta)}\right) = \begin{cases} 1 & \Delta P_{v(\delta)} < \Delta P_{max} \\ \mu_{i1}(\Delta P_{v(\delta)}) & \Delta P_{max} \leq \Delta P_{v(\delta)} < \Delta P_{min} \\ 0 & \Delta P_{v(\delta)} \geq \Delta P_{min} \end{cases} \tag{8}$$

It is important to notice that clustering every difference value p_i in the set Ω of all difference values in the image, does not represent the uncertainty region between *small* and *large* difference values in a suitable form. Then, the FCM method should be applied only in a region of interest inside the average histogram $\hat{h}$, or alternatively inside the probability mass function $f_{\hat{h}}$. If we denote the region of interest to be the interval $\Lambda = [\lambda_1, \lambda_2] \subset \Omega$, then we propose to use the FCM method only with data points in the subset Λ such that $p_i \in \Lambda$, as denoted in (9). Finally, the overall FCM-based algorithm as the global approach of the edge detector is summarized in Algorithm 1.

$$J_m = \sum_i \sum_{j=1}^{k=2} \mu_{ij}{}^2 \|p_i - c_i\|^2 \quad , \quad \forall p_i \in [\lambda_1, \lambda_2] \tag{9}$$

Algorithm 1 Global approach based on FCM method.

Input: A matrix of difference values dp from each direction v, the region of interest from λ_1 to λ_2, and the number of bins $nbins$ in the histogram.
Output: The membership functions μ_{small} and μ_{large}.

```
procedure AUTOMATIC-TUNING USING FCM($dp$,$\lambda_1$,$\lambda_2$,$nbins$)
    $h_v \leftarrow$ matrix of histograms of size $8 \times nbins$
    for $v = 1$ to $8$ do
        $h_v(v, :)$ = histogram of $dp$ associated to direction $v$ with $nbins$
    end for
    $\hat{h}$ = mean of $h_v$,  $\forall v = 1, \ldots, 8$
    $p_i$ = all values of $\hat{h} \in [\lambda_1, \lambda_2]$
    $c, \mu$ = fcm($p_i$, 2)                              ◁ FCM-method with $k = 2$ and $s = 2$
                                                          ◁ $\mu$ is a matrix such that $\mu = [\mu_{i1}, \mu_{i2}]$
    if $c_1 < c_2$ then
        $\mu_{tmp} = \mu_{i1}$
    else
        $\mu_{tmp} = \mu_{i2}$
    end if
    $\Delta P_{max}$ = argmax($\mu_{tmp}$)
    $\Delta P_{min}$ = argmin($\mu_{tmp}$)
    $\mu_{small}(1 : \Delta P_{max}) = 1$
    $\mu_{small}(\Delta P_{max} : \Delta P_{min}) = \mu_{tmp}(\Delta P_{max} : \Delta P_{min})$
    $\mu_{small}(\Delta P_{min} : nbins) = 0$
    $\mu_{large} = 1 - \mu_{small}$
    $\mu_{small}$ and $\mu_{large}$ converted as look-up table or hash function
    return $\mu_{small}$, $\mu_{large}$
end procedure
```

3 Results and Discussion

In this section, we evaluate the performance of our proposed fuzzy ensemble based edge detection method using a public database and compare it with well-known classical methods (Sobel, Prewitt, LoG, Roberts, Canny) and the fuzzy based method proposed by [22].

3.1 Materials

To evaluate our proposal, we used the "Berkeley Segmentation Dataset and Benchmark BSD300" [28] available from [29], which is formed by natural images manually segmented by some subjects (between 5 and 10). The ground truth (GT) is formed by the superposition of weighted manually segmented annotations. In Figure 7 we show the selected images, and its corresponding ID in the database, in which we applied the edge detection methods.

3.2 Evaluation Method for Edge Detection

We conducted several experiments to test the performance of our method on natural and noisy images. The evaluation method consisted on the following steps: (i) application of the edge detection algorithm to an image, (ii) normalization of the edges map (for these experiments, between 0 to 255), (iii) binarization of the edge map by a given set of N thresholds, and (iv) comparison of the N binary images against to the ground truth. In the last step, we carried out a matching edges process as in [29] between each binary image and the ground truth using the libraries for graph

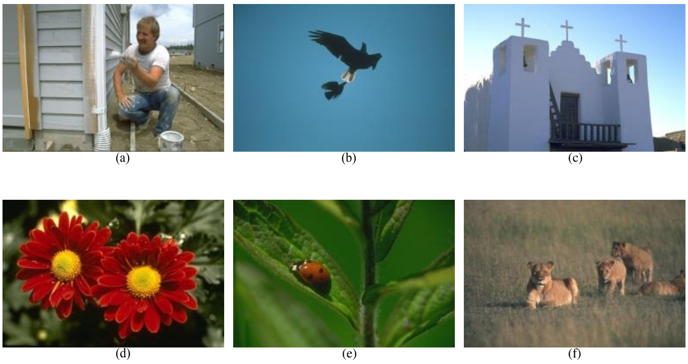


Figure 7
Selected images for experimentation extracted from the BSD300 dataset. IDs: (a) 23080, (b) 135069, (c) 24063, (d) 124084, (e) 35058 and (f) 105053.

assignment problem of Andrew Goldberg's CSA package [30]. Then, we applied different metrics (precision, recall and F-measure) and we obtained F-measure vs threshold curves for each edge detection method.

For the boundary detection problem, some metrics were used to measure the performance of the method tested, the most used are precision (*P*), recall (*R*) and F- measure (*F*). Precision is defined as the ratio of edges that are true positives rather than false positives, or the probability that the edge detector is valid; whereas re-call is the ratio of edges that are true positives detected rather than missed, or the probability that the reference data was detected [31]. Another metric that allows to measure the effectiveness of the algorithm equally in terms of precision and recall is F-measure, that captures the trade-off between precision and recall, as the weighted harmonic mean.

One important issue in the parametric methods for the image processing is the sensitivity analysis of the parameters. To this end, we tried different instances of our proposed method in order to find a suitable set of parameters. These instances are proposed in terms of mask definition (δ) and the region of uncertainty ($[\lambda_1, \lambda_2]$). Table 2 shows these variations (F_1–F_7) as well as the parameters used in classical methods of edge detection.

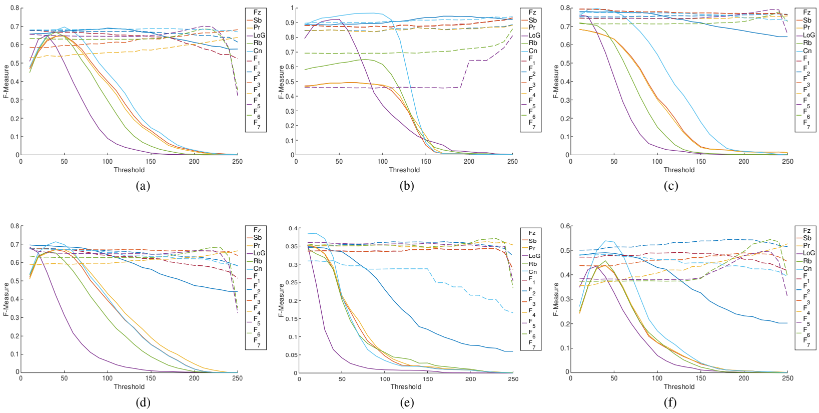
3.2.1 Computation of Suitable Parameters

Using the evaluation method described above and given a set of thresholds (between 10 and 250 with steps of 10), we show F-measure vs threshold curves for the selected images in Figure 8 for selected images. From these results, we may observe that our proposals (F_1–F_7) are competitive with the classical methods found in the literature (*Sb*, *Pr*, *LoG*, *Rb* and *Cn*) and comparable with the method of [22] (*Fz*).

By inspection in Figure 8, it can be seen clearly two different clusters: traditional and fuzzy based methods. The classical ones are less robust with respect to thresh-

Table 2
Labels and parameters for edge detection methods

Label	Method	Parameters
Fz	Fuzzy based Haq et al. [22]	$\delta = 1$ and $[\lambda_1 = 25, \lambda_2 = 75]$
Sb	Sobel	3 × 3 kernel
Pr	Prewitt	3 × 3 kernel
LoG	LoG	3 × 3 kernel
Rb	Roberts	3 × 3 kernel
Cn	Canny	3 × 3 kernel
F_1	Our proposal 1	$\delta = 1$ and $[\lambda_1, \lambda_2]$ were estimated automatically using PMF and FCM based methods
F_2	Our proposal 2	$\delta = 2$ and $[\lambda_1, \lambda_2]$ were estimated automatically using PMF and FCM based methods
F_3	Our proposal 3	$\delta = 1$ and $[\lambda_1 = 10, \lambda_2 = 60]$
F_4	Our proposal 4	$\delta = 2$, and $[\lambda_1 = 10, \lambda_2 = 60]$
F_5	Our proposal 5	$\delta = 1$ and $[\lambda_1, \lambda_2]$ were estimated by PMF based method
F_6	Our proposal 6	$\delta = 2$ and $[\lambda_1, \lambda_2]$ were estimated by PMF based method
F_7	Our proposal 7	$\delta = 1$ and $[\lambda_1 = 20, \lambda_2 = 50]$ using FCM based method



(a) (b) (c)

(d) (e) (f)

Figure 8
F-measure vs threshold curves for the selected images. IDs: (a) 23080, (b) 135069, (c) 24063, (d) 124084, (e) 35058 and (f) 105053.

old in contrast to the fuzzy based methods described in this work. Quantitatively, the robustness dependency can be measured by computing the F-measure standard deviation for all methods over selected images, as shown in Figure 9. An important evidence from the latter is that the fuzzy methods have standard deviation values close to 0.1, observing that the instances of our proposal are less than this value. Thus, fuzzy based methods are less dependent to threshold. Therefore, we fixed the threshold at the middle range of edges maps (128) for the following experiments, allowing to define an automatic binarization method with a fixed threshold.

In order to determine the suitable variation of our method, we computed the F-measure and precision at the fixed threshold for our proposals (F_1–F_7) as summarized in Tables 3 and 4, respectively. In Table 3, we may observe that the best set of parameters corresponds to the tests F_2, F_3 and F_7, and the precision measure shown

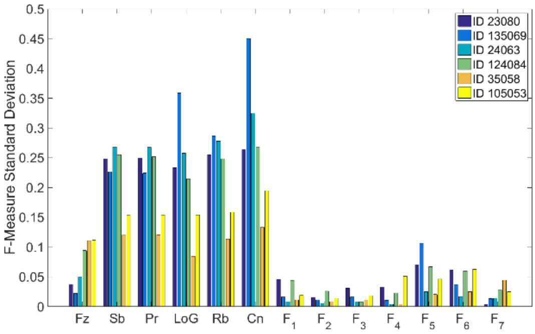


Figure 9
F-measure standard deviation for all methods over selected images.

Table 3
F-measure for fixed threshold

ID	F_1	F_2	F_3	F_4	F_5	F_6	F_7
23080	0.653	0.672	0.623	0.561	0.645	0.628	**0.684**
135069	0.884	0.857	0.884	0.854	0.456	0.691	**0.908**
24063	0.773	0.755	**0.773**	0.742	0.742	0.713	0.771
124084	0.633	0.655	**0.671**	0.611	0.648	0.627	0.636
35058	0.341	**0.361**	0.340	0.348	0.355	0.353	0.288
105053	0.490	**0.532**	0.462	0.418	0.381	0.376	0.448

Table 4
Precision measure for fixed threshold

ID	F_1	F_2	F_3	F_4	F_5	F_6	F_7
23080	**0.730**	0.692	0.478	0.402	0.511	0.490	0.618
135069	0.803	0.752	0.804	0.748	0.296	0.528	**0.852**
24063	0.746	0.699	0.747	0.642	0.620	0.577	**0.820**
124084	0.640	**0.657**	0.569	0.466	0.524	0.491	0.628
35058	0.343	0.326	0.342	0.269	0.246	0.241	**0.541**
105053	0.414	0.439	0.334	0.272	0.241	0.234	**0.469**

in Table 4 reports that the best results correspond to the test F_7. According to previous results, we selected F_7 to be the method with suitable parameters ($\delta = 1$ and $[\lambda_1 = 20, \lambda_2 = 50]$). As an example of the results obtained, in Figure 10 we report the edge detection results using our F_7 method without binarization. For contrasting with the other methods, Figure 11 shows an image example (ID image 24063) of the edge detection results for the fixed threshold. It can be seen that the best edge detection corresponds to *Fz*, *Cn* and F_7 methods. However, *Fz* and F_7 approaches detect more details on the image. Moreover, F_7 preserves more edges than *Fz*.

3.2.2 Experiments on Noisy Images

In order to prove the robustness of our method in presence of noise, we tested it using images corrupted with 35, 30 and 24 dB of peak signal to noise ratio (PSNR) levels of salt & pepper noise. As concluded previously, the following experiments were carried out using the set of parameters corresponding to F_7 instance.

The same evaluation method was used to compare the performance of our proposed

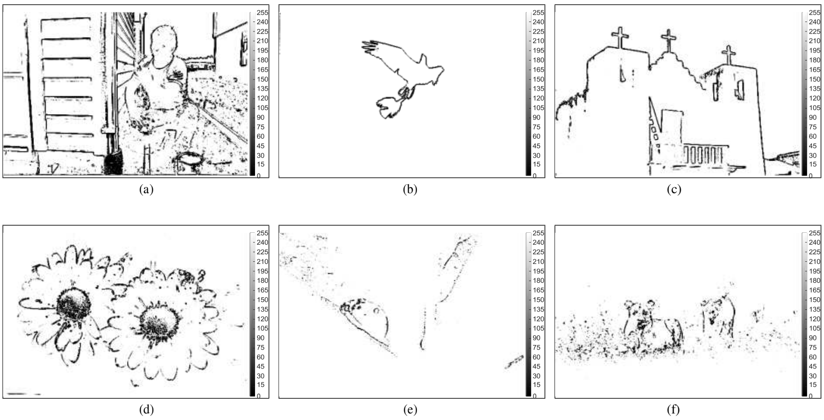


Figure 10
The edge maps resulting from F_7 method for images: (a) 23080, (b) 135069, (c) 24063, (d) 124084, (e) 35058 and (f) 105053.

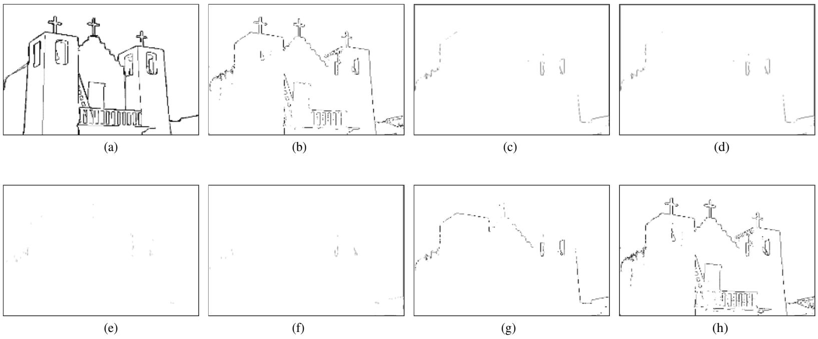
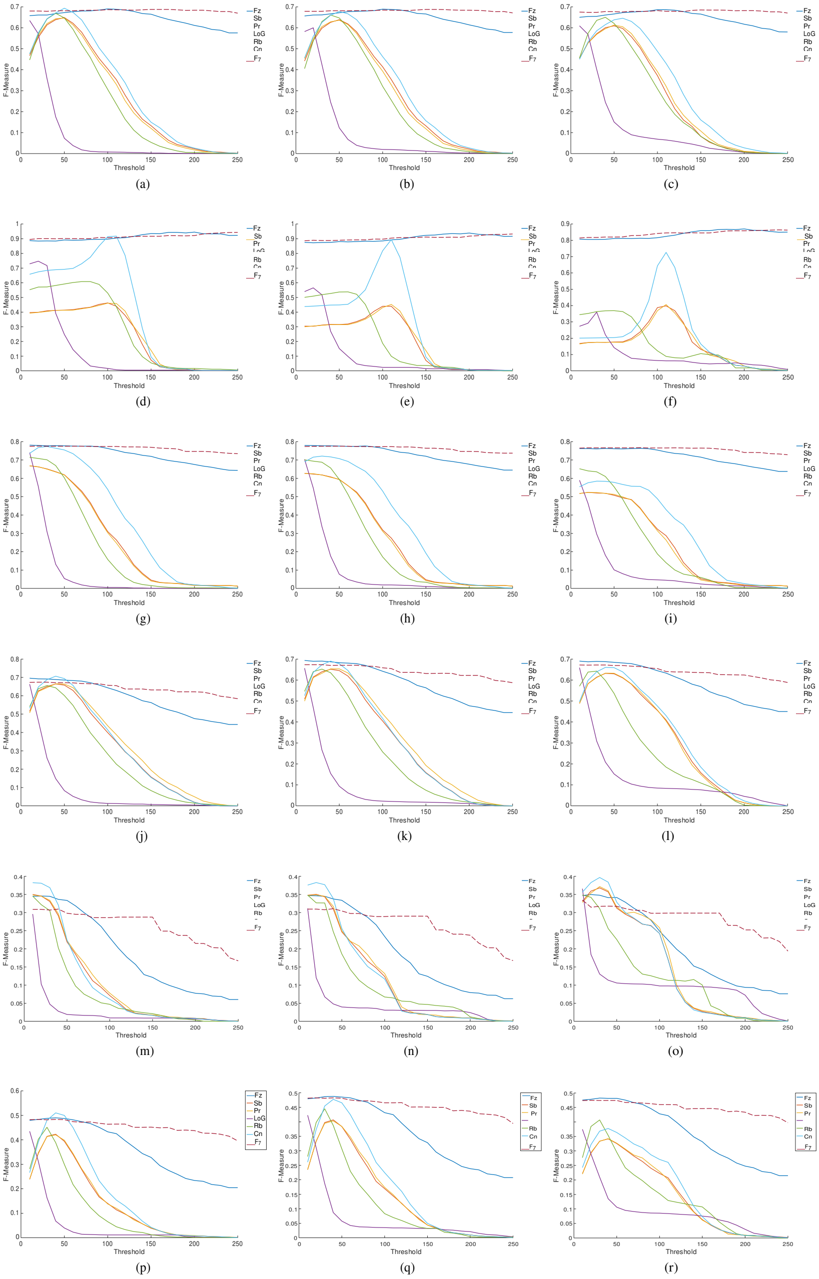


Figure 11
The edge maps resulting using a fixed threshold for all methods (ID Image 24063). Methods: (a) GT, (b)*Fz*, (c) *Sb*, (d) *Pr*, (e) *LoG*, (f) *Rb*, (g) *Cn* and (h) F_7.

fuzzy ensemble based edge detection approach. Figure 12 shows the F-measure vs threshold curves for each edge detection method. Columns represent the noise levels (35, 30 and 24 dB of PSRN) and rows correspond to the ID images. Analyzing them, we observed that traditional methods are less robust against to noise levels than fuzzy based methods. For instance, in Figure 12 (g-i), we see that in the best traditional method (Canny) the F-measure decreases 25% in the different noise levels values (35, 30, 24 dB of PSNR). In contrast, the fuzzy based methods remain almost constant independently to noise level (the F-measure decrease only 1.2% approximately). We may appreciate that the F_7 method is more robust with respect to threshold variation than *Fz* method. The same behavior can be seen in all the images.

Figures 13 and 14 report the binarization edge detection results using all the methods (rows) at different noise levels (columns) for the image 24063. In general, we see degradation of edges detection and addition of false edges because of the increasing



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l) (m) (n) (o) (p) (q) (r)

Figure 12
F-measure vs threshold curves for different levels of noise. ID/PSNR: (a) 23080/35dB, (b) 23080/30dB, (c) 23080/24dB, (d) 135069/35dB, (e) 135069/30dB, (f) 135069/24dB, (g) 24063/35dB, (h) 24063/30dB, (i) 24063/24dB, (j) 124084/35dB, (k) 124084/30dB, (l) 124084/24dB, (m) 35058/35dB, (n) 35058/30dB, (o) 35058/24dB, (p) 105053/35dB, (q) 105053/30dB and (r) 105053/24dB.

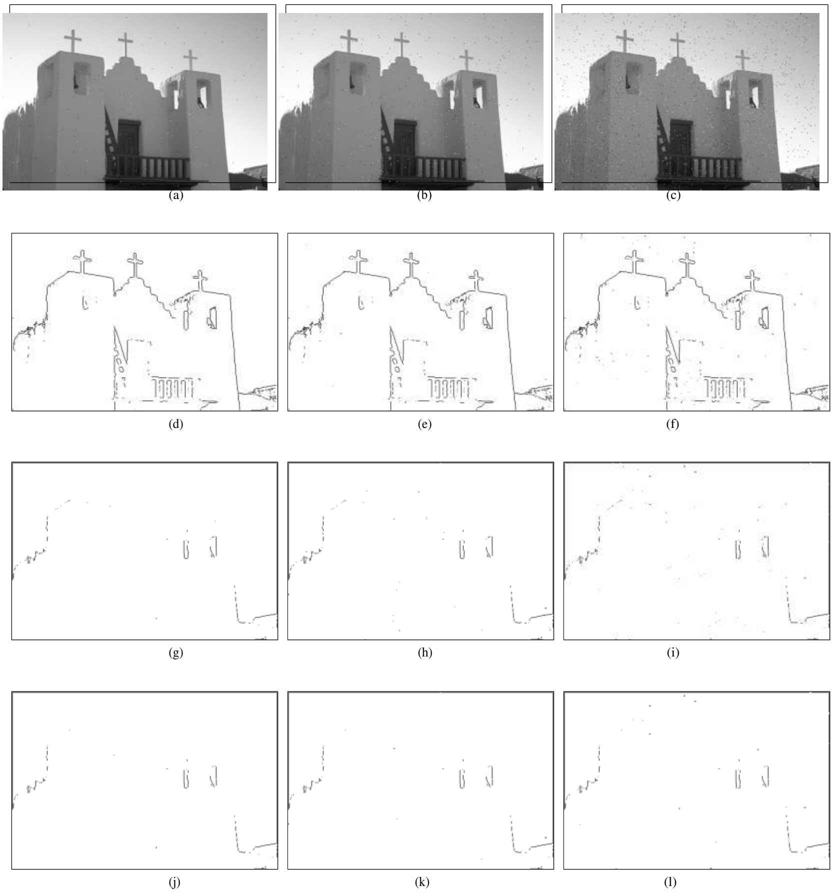


Figure 13
Binarization results in noisy image 24063 using a fixed threshold. (a) PSNR 35dB, (b) PSNR 30dB and (c) PSNR 24dB. Method/PSNR: (d) *Fz*/35dB, (e) *Fz*/30dB, (f) *Fz*/24dB, (g) *Sb*/35dB, (h) *Sb*/30dB, (i) *Sb*/24dB, (j) *Pr*/35dB, (k) *Pr*/30dB and (l) *Pr*/24dB.

of salt & pepper noise level. In addition, *Fz*, *Cn* and F_7 methods give the best responses independently to the noise level. At first, we can see that fuzzy based methods detect less noise than Canny method. In comparison to *Fz*, F_7 method preserves more edge details.

In Figure 15, we show all the edge detection images for the highest noise level (24 dB) using the F_7 method. In Figures 15(b and c), it can be seen well defined and proper edges maps corresponding to images with homogenous backgrounds, as noted in curves Figures 12(f and i). For the high contrast images, Figures 15(a and d), the edges remain well detected, as observed in Figures 12(c and l). However, F_7 method is not robust for low contrast images as shown in Figures 15(e and f). Nevertheless, the other methods used in this work also have the same problem for low contrast images as depicted in Figure 12(o and r).

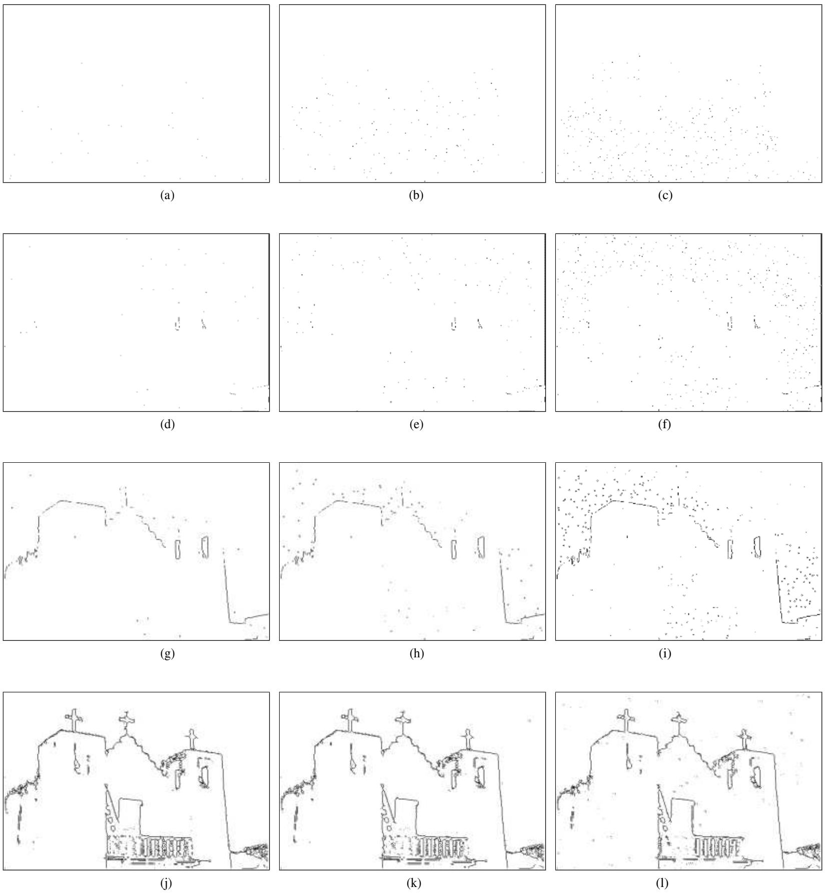


Figure 14
Cont. Binarization results in noisy images using a fixed threshold. Method/PSNR: (a) *LoG*/35dB, (b) *LoG*/30dB, (c) *LoG*/24dB, (d) *Rb*/35dB, (e) *Rb*/30dB, (f) *Rb*/24dB, (g) *Cn*/35dB, (h) *Cn*/30dB, (i) *Cn*/24dB, (j) F_7/35dB, (k) F_7/30dB and (l) F_7/24dB.

3.3 Discussion

From the experiments, above, the advantages of our methodology are listed following. First, our methodology use a fuzzy technique that is very robust to the noise because of its uncertainty characteristic [23]. In addition, in our proposal the membership functions are designed from a global estimation of statistics parameters on the image in contrast with the Fz approach [22]. In fact, the input membership functions are defined from both the region of uncertainty controlled by λ_1 and λ_2 and its shape characterized using the FCM algorithm. It allows less variability with respect to threshold. In addition, this method allows to choose the δ value that controls the mask definition.

To validate the latter, we chose a straightforward protocol experiment to determine the suitable set of parameters in our edge detection method. We adopted the F-

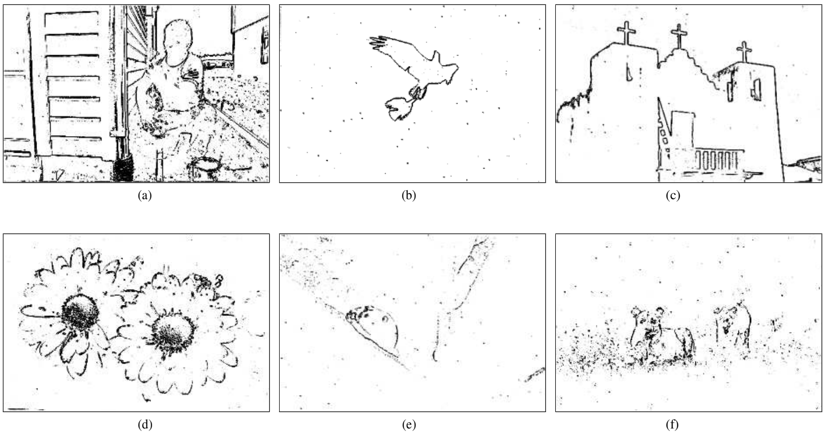


Figure 15
The edge maps resulting from the F_7 method applied to the 24 dB noisy images using a fixed threshold. IDs: (a) 23080, (b) 135069, (c) 24063, (d) 124084, (e) 35058 and (f) 105053.

measure metric to test sensibility and precision of the method. We also used it to confirm the adoption of a fixed threshold, handling the threshold choosing problem [33].

From the results, in both cases with and without noise, our methodology overcomes the traditional methods and compete with the fuzzy method of [22]. In particular, our method performs better in presence of noise and is also robust regardless of the threshold. In that way, we can argue the choosing of a fixed value. In addition, the performance of our method is invariant to different noise levels.

On the other hand, our methodology presents some limitations. For low contrast images our approach decreases its performance. However, it can be improved by changing the definition of the mask. Also, we noticed that the region of uncertainty has direct impact on the performance of the edge detection method. In this work, it has not been detected automatically.

Conclusions

In this work, we have proposed a fuzzy ensemble based method for edge detection. It includes a FCM strategy to define the input membership functions of the fuzzy inference system from a global estimation of statistics parameters on the image, and the characterization of the region of uncertainty.

We tested the performance of the method using a public database and we compared it with classical methods and the fuzzy based method presented in [22]. In order to find a suitable set of parameters in our method we tested it with different instances. The F-measure was used to evaluate the performance of the edge detectors. We also measured the standard deviation for the F-measure to confirm the invariability of our method with respect to the threshold in comparison to classical methods.

The method was also tested with three levels of salt & pepper noise. Results validated that fuzzy ensemble based method overcomes the traditional methods and compete with the fuzzy method of [22]. In general, the metric values obtained from our proposal in the highest noise condition were better than the other fuzzy approach.

The advantages of our methodology include: the design of membership functions from a global estimation of statistics parameters on the image using FCM for shape characterization, the less variability with respect to threshold and the robustness to noise.

In comparison to the fuzzy method of [22], our methodology competes and extends its fixed parameters (window size and shape of membership functions) by including some tuning parameters. In addition, our approach allows the use of a constant threshold. Furthermore, the current work has some advantages with respect to other image fuzzy enhancement methods as [20] like the definition of the membership functions are not preset, an optimization algorithm is not required for threshold selection and an extensive quantitative analysis to compare to other methods was performed.

For future work we are considering to define another kind of mask to handle low contrast images and to design an automatic strategy to estimate the region of uncertainty. Also, other strategies to add high order information (e.g. texture features) to the inputs of the fuzzy system could be used to improve the performance of the edge detection method.

References

[1] Mathew SP, Balas VE, Zachariah KP, Samuel P. A content-based image retrieval system based on polar raster edge sampling signature. Acta Polytechnica Hungarica. 2014;11(3):25-36

[2] Mathew SP, Balas VE, Zachariah KP. A content-based image retrieval system based on convex hull geometry. Acta Polytechnica Hungarica. 2015;12(1):103-116

[3] Torre V, Poggio TA. On Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1986;PAMI-8(2):147-163

[4] Arbelaez P, Maire M, Fowlkes C, Malik J. Contour Detection and Hierarchical Image Segmentation. IEEE Trans Pattern Anal Mach Intell. 2011;33(5):898-916

[5] Jin-Yu Z, Yan C, Xian-Xiang H. Edge detection of images based on improved Sobel operator and genetic algorithms. In: 2009 International Conference on Image Analysis and Signal Processing; 2009. pp. 31-35

[6] Rosenfeld A. The Max Roberts Operator is a Hueckel-Type Edge Detector. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1981;PAMI-3(1):101-103

[7] Yang L, Wu X, Zhao D, Li H, Zhai J. An improved Prewitt algorithm for edge detection based on noised image. In: Image and Signal Processing (CISP), 2011 4th International Congress on. Vol. 3; 2011. pp. 1197-1200

[8] Ulupinar F, Medioni G. Refining edges detected by a LoG operator. Computer Vision, Graphics, and Image Processing. 1990;51(3):275-298

[9] Canny J. A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1986;PAMI-8(6):679-698

[10] Genming C, Baozong YAA. A new edge detector with thinning and noise resisting abilities. Journal of Electronics (China). 1989;6(4):314-319

[11] Wang X, Zhang X, Gao R. An adaptive edge detection algorithm based on gray-scale morphology. In: Measurement, Information and Control (ICMIC), 2013 International Conference on. Vol. 02; 2013. pp. 1251-1254

[12] Jiang J, Chuang C, Lu Y, Fahn C. Mathematical-morphology-based edge detectors for detection of thin edges in low-contrast regions. IET Image Processing. 2007;1(3):269-277

[13] Guo X, Xu Z, Lu Y, Pang Y. An Adaptive Edge Detector Using Soft Mathematical Morphology. In: The Fifth International Conference on Computer and Information Technology (CIT'05); 2005. pp. 608-607

[14] Krishnamurthy S, Iyengar SS, Holyer RJ, Lybanon M. Histogram-based morphological edge detector. IEEE Transactions on Geoscience and Remote Sensing. 1994;32(4):759-767

[15] Singh H, Kaur G, Gupta N. Robust edge detector using back propagation neural network with multi-thresholding. In: Computational Intelligence and Computing Research (ICCIC), 2014 IEEE International Conference on; 2014. pp. 1-6

[16] Li N, Huo H, m Zhao Y, Chen X, Fang T. A Spatial Clustering Method With Edge Weighting for Image Segmentation. IEEE Geoscience and Remote Sensing Letters. 2013;10(5):1124-1128

[17] Boskovitz V, Guterman H. An adaptive neuro-fuzzy system for automatic image segmentation and edge detection. IEEE Transactions on Fuzzy Systems. 2002;10(2):247-262

[18] Khunteta A, Ghosh D. Edge detection via fuzzy rule-based edge strength estimation and optimal threshold selection using PSO. In: 2013 IEEE 8th International Conference on Industrial and Information Systems; 2013. pp. 560-565

[19] Talai Z, Talai A. A fast edge detection using fuzzy rules. In: Communications, Computing and Control Applications (CCCA), 2011 International Conference on; 2011. pp. 1-5

[20] Zhang D, Zhan B, Yang G, Hu X. An improved edge detection algorithm

based on image fuzzy enhancement. In: 2009 4th IEEE Conference on Industrial Electronics and Applications; 2009. pp. 2412-2415

[21] Setarehdan SK, Soraghan JJ. Automatic cardiac LV boundary detection and tracking using hybrid fuzzy temporal and fuzzy multiscale edge detection. IEEE Transactions on Biomedical Engineering. 1999;46(11):1364-1378

[22] Haq I, Anwar S, Shah K, Khan MT, Shah SA. Fuzzy Logic Based Edge Detection in Smooth and Noisy Clinical Images. PLoS ONE. 2015;10(9):1-17

[23] Guillaume S. Designing Fuzzy Inference Systems from Data: An Interpretability-Oriented Review. IEEE Transactions on Fuzzy Systems. 2001;9(3):426-443

[24] Kim S, Lee M, Lee J. A Study of Fuzzy Membership Functions for Dependence Decision-Making in Security Robot System. Neural Computing and Applications. 2015; pp. 1-10

[25] Shtovba SD. Fuzzy model Tuning Based on a Training Set With Fuzzy Model Output Values. Cybernetics and Systems Analysis. 2007;43(3):334-340

[26] Iancu I. 16. In: Dadios E, editor. A Mamdani Type Fuzzy Logic Controller. InTech; 2012. pp. 325-350

[27] Bezdek JC. Pattern Recognition With Fuzzy Ocjective Function Algorithms. Norwell, MA: Kluwer Academic Publishers; 1981

[28] Martin D, Fowlkes C, Tal D, Malik J. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In: Proc. 8th Int'l Conf. Computer Vision. Vol. 2; 2001. pp. 416-423

[29] Martin D, Fowlkes C, Tal D, Malik J. The Berkeley Segmentation Dataset and Benchmark; 2007. Online. Available from: http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/

[30] Goldberg A. Andrew Goldberg's Network Optimization Library; 1995. Online. Available from: http://www.avglab.com/andrew/soft.html

[31] Martin DR, Fowlkes CC, Malik J. Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2004;26(5):530-549

[32] Rijsbergen CJv. Information Retrieval. 2nd ed. Newton, MA, USA: Butterworth-Heinemann; 1979. Available from: http://www.dcs.gla.ac.uk/Keith/pdf/Chapter7.pdf

[33] Kim DS, Lee WH, Kweon IS. Automatic edge detection using 3x3 ideal binary pixel patterns and fuzzy-based edge thresholding. Pattern Recognition Letters. 2004;25(1):101-106

An Intelligent System for the Diagnosis of Skin Cancer on Digital Images taken with Dermoscopy

Heydy Castillejos-Fernández[a], Omar López-Ortega[a], Félix Castro-Espinoza[a] and Volodymyr Ponomaryov[b]

[a]Universidad Autónoma del Estado de Hidalgo, Área Académica de Computación y Electrónica, Carretera Pachuca – Tulancingo km. 4.5, Mineral de la Reforma, Hidalgo, México. C. P. 42083
heydy_castillejos@uaeh.edu.mx, lopezo@uaeh.edu.mx, fcastro@uaeh.edu.mx

[b]Instituto Politécnico Nacional, ESIME Unidad Profesional Culhuacan
Avenida Santa Ana 1000, Coyoacan, San Francisco Culhuacan,
Ciudad de México, México. C.P. 04430
vponomar@ipn.mx

Abstract: Skin cancer is a major health issue affecting a vast segment of the population regardless the skin color. This affectation can be detected using dermoscopy to determine whether the visible spots on skin are either benign or malignant tumors. In spite of the specialists' experience, skin lesions are difficult to classify, reason for which computer systems are developed to increase the effectiveness of cancer detection. Systems assisting in the detection of skin cancer process digital images to determine the occurrence of tumors by interpreting clinical parameters, relying, firstly, upon an accurate segmentation process to extract relevant features. Two of the well-known methods to analyze lesions are ABCD (Asymmetry, Border, Color, Differential structures) and the 7-point check list. After clinically-relevant features are extracted, they are used to classify the presence or absence of a tumor. However, irregular and disperse lesion borders, low contrast, artifacts in images and the presence of various colors within the region of interest complicate the processing of images. In this article, we propose an intelligent system running the following method. The feature extraction stage begins with the segmentation of an image, for which we apply the Wavelet – Fuzzy C-Means algorithm. Next, specific features should be determined, among others the area and the asymmetry of the lesion. An ensemble of clusterers extracts the Red-Green-Blue values that correspond to one or more of the colors defined in the ABCD guide. The feature extraction stage includes the discovery of structures that appear in the lesion according to the method known as Grey Level Co-Occurrence Matrix (GLCM). Then, during the detection phase, an ensemble of classifiers determines the occurrence of a malignant tumor. Our experiments are performed on images taken from the ISIC repository. The proposed system provides a skin cancer detection performance above 88 percent, as measured by the accuracy. Details of how this performance fares when compared with other systems are also given.

Keywords: segmentation; fuzzy logic; color detection; classification

1. Introduction

Skin cancer is a major health issue affecting vast segments of the population regardless the skin color. Data indicate that the incidence of melanoma, which is a type of cancer that metastasizes rapidly, has increased alarmingly. It begins by modifying melanocytes (epidermal cell that produces melanin) of normal skin or moles, resulting as a dark area on the skin. This damaging process changes the normal concentration of melanin (dark-brown, black or reddish-brown substance that is natural of people's skin, hair and eyes). Because this affectation is apparent on skin, it is possible to use a non-invasive technique called dermoscopy (derma - scope) to determine whether the visible spots on skin are either benign or malignant tumors.

Numerous techniques have been proposed in order to characterize and define patterns and structures of pigmented and non pigmented skin lesions. Nonetheless, skin lesions are difficult to classify, reason for which computer-based systems are developed to improve the detection of skin cancer through the extraction and interpretation of several clinical parameters. Generally, the following stages must be completed by any computerized diagnostic system:

- Pre-processing. In this stage, filters for removal artifacts are applied.
- Image segmentation. A specific region of the lesion is separated from the rest of the original digital image.
- Feature extraction. Clinically-relevant features that are defined in various guides, among others the ABCD (Asymmetry, Border, Color, Differential structures), must be extracted correctly in order to interpret the lesion. Another guideline that could be implemented is a checklist of 7 criteria that define a malignant tumor.
- Learning and diagnosis. This stage is facilitated by employing machine learning techniques, i.e. classifiers.

Thus, intelligent systems must implement an accurate image segmentation process to analyze borders, colors, and structures of a lesion. This requirement is compulsory to extract clinically relevant features of dermoscopy images. However, irregular and disperse lesion borders, low contrast, artifacts in images and variety of colors within the interest region pose a tremendous challenge in the segmentation step. After the segmentation and feature extraction processes are complete, the set of relevant features must be classified accurately to determine the presence of a malignant tumor or discard its occurrence.

To solve these two major problems (feature extraction and classification) we propose an intelligent system for detecting whether a lesion is a benign or malignant tumor. The proposed intelligent system executes the following method. First, feature extraction is achieved by segmenting the image of the skin lesion with the Wavelet Fuzzy C Means (W-FCM) algorithm [1]. When the lesion segmentation is done, the following features are obtained: asymmetry, all the features considered in the Grey-Level Co-Occurrence Matrix (GLCM) and, as novel proposals, our method includes the extraction of the eccentricity value and the color content of the lesion. The color extraction is performed by an ensemble of clusterers that estimates the presence of one or more colors following the ABCD guide.

After the feature extraction phase is terminated, the learning phase of the intelligent system commences. We propose an ensemble of classifiers as a means to elevate the accuracy of classifying the lesion as either benign or malignant. We measure the effectiveness of the classification task by calculating values for sensibility, specificity, accuracy and the area under the ROC curve.

Our experiments were done on images taken from the ISIC repository. With the method proposed, our system provides a skin cancer detection performance ranking at the top tier, as contrasted with other systems that have been reported. The paper is organized as follows. Section 2 presents the method that covers the lesion segmentation, color and clinically-relevant features extraction, and learning. Section 3 contains detailed experimental results. A comparison of our system performance with other systems is given in Section 4. Finally, conclusions and future work are delineated.

2. The Proposed Method

The method that we propose is illustrated in Figure 1, where each of the stages is represented with a dash-lined rectangle. The relevant stages of the method are: Lesion segmentation and feature extraction based on W-FCM; color extraction based on an ensemble of clusterers; creation of the features vector and, finally, learning and prediction based on an ensemble of classifiers.

Each phase is explained next.

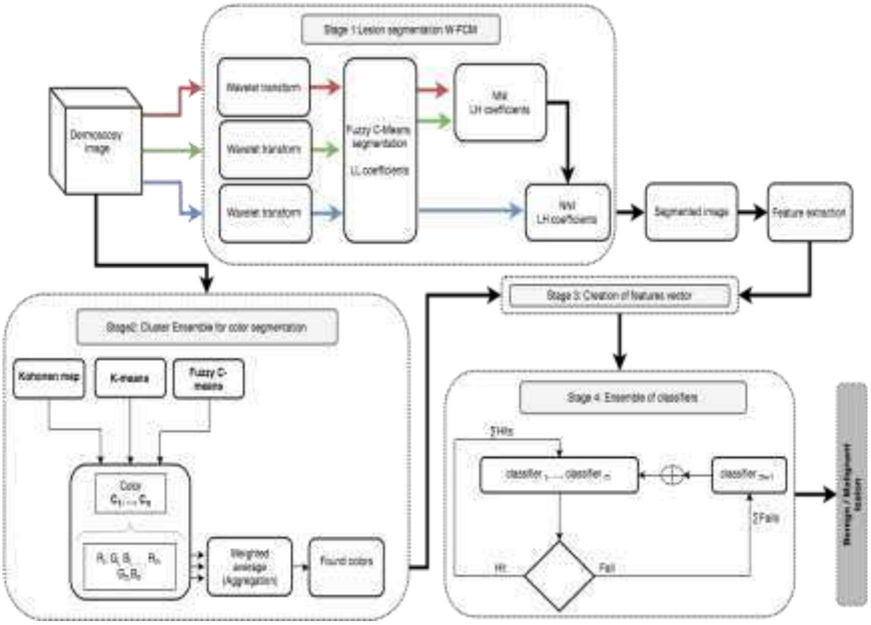

Stage 1:Lesion segmentation W-FCM
Dermoscopy image
Wavelet transform
Wavelet transform
Wavelet transform
Fuzzy C-Means segmentation
LL coefficients
NNI LH coefficients
NNI LH coefficients
Segmented image
Feature extraction
Stage 3: Creation of features vector
Stage2: Cluster Ensemble for color segmentation
Kohonen map
K-means
Fuzzy C-means
Color
$C_1, ..., C_n$
$R_i, G_i, B_i, ... R_n, G_n, B_n$
Weighted average (Aggregation)
Found colors
Stage 4: Ensemble of classifiers
∑Hits
$\text{classifier}_1, ..., \text{classifier}_m$
classifier_{m+1}
∑Fails
Hit
Fail
Benign / Malignant lesion

Figure 1

Block diagram of the proposed method to determine whether a lesion is a benign or malignant tumor

2.1 Lesion Segmentation

Before the segmentation process, we employ a framework that employs the feature extraction in Wavelet Transform (WT) space. This operation is paramount because it is possible to obtain data from the Red, Blue and Green channels of a digital image [1, 2]. The acquisition of the three channels is performed by a nearest neighbor interpolation (NNI).

The segmentation process occurs as follows: a digital color image *I[n,m]* is separated in Red, Green and Blue channels, where each color channel is decomposed calculating their wavelets coefficients using Mallat's pyramid algorithm [3]. Then, using the biorthogonal 6.8 wavelet family, the original image is decomposed into four sub-bands. Three of these sub-bands, named LH, HL and HH represent the finest scale wavelet coefficient (detail images), while the sub-band LL corresponds to coarse level coefficients (approximation image), noted below as $D_h(2^i)$, $D_v(2^i)$, $D_d(2^i)$, and $A(2^i)$, respectively at given scale 2^j, for $j=1,2, \ldots J$, where J is the number of scales used in the Discrete Wavelet Transform (DWT) [4].

The DWT is represented as follows:

$$W_i = |W_i| exp(j * \theta_i), \tag{1}$$

$$|W_i| = \sqrt{\left|D_{h,i}\right|^2 + \left|D_{v,i}\right|^2 + \left|D_{d,i}\right|^2}, \tag{2}$$

where $|W_i|$ is the wavelet modulus on a chosen decomposition level *i;* $D_{h,i}$, $D_{v,i}$, $D_{d,i}$ are the horizontal, vertical and diagonal detail components on a level *i*, and the phase Θ_i, is defined as follows:

$$\theta_i = \begin{cases} \alpha_i & if \quad D_{h,i} > 0 \\ \pi - \alpha_i & if \quad D_{h,i} < 0 \end{cases}, \tag{3}$$

$$\alpha_i = tan^{-1}\left(D_{v,i}/D_{h,i}\right). \tag{4}$$

Consequently, W_i is considered as a new image for each color channel. The next step is the Fuzzy C-Means segmentation, where the segmented image corresponding to the red channel is interpolated with the segmented image corresponding to the green channel. This new image is obtained by applying a *NNI*. The *NNI* is repeated, taking the segmented image corresponding to the blue channel. The image that is obtained at the end of these interpolations is considered the output of the segmentation step.

By using the three color channels in the segmentation process, the extraction of clinically-relevant features is improved, thus making the classification more accurate, as compared when the original image is used to extract relevant features.

The color segmentation can be executed while the lesion segmentation is taking place. The color segmentation process is explained next.

2.2 Color Segmentation

Another variable that is used to diagnose skin cancer is the color content of the lesion. To detect what colors are present in the image, color segmentation is done by an Ensemble of Clusterers (EoCls). We decided to use EoCls because they are thought to overcome the limitations of single clustering algorithms by exploiting diversity in data processing. An EoCls can be obtained by using clustering algorithms on the same data or by using different values to the parameters of a single algorithm [5]. The EoCls employed to detect the RGB values of the colors that are present in a lesion is formed by three different algorithms: K-Means, Fuzzy C-Means and a Kohonen map. These algorithms run in parallel, each on its own thread, making the color extraction process faster. Each of the clusterers extracts the representative values of the partitions detected in the image being analyzed. Then, by averaging each channel representative, a global RGB value is obtained for each color.

2.3 Creation of Features Vector

Texture analysis is one of the most important stages for a better classification because texture features provide special characteristics present in the image. Several authors have proposed methods to extract features of dermoscopy images [6, 7, 8, 9, 10]. However, those methods extract statistical properties and do not consider both local and global spatially correlated relationships among pixels. As opposed to the mentioned reports, we calculate the feature extraction using the GLCM method. The features vector includes: assymetry, area of the lesion, eccentricity, all the features in GLMC, and the color content of the lesion.

2.4 Learning and Predicting by an Ensemble of Classifiers

Classification is the task of learning a target function *f* that maps the description of a certain set of instances to the values of a predefined attribute known as class. The input data for solving a problem of this kind is a collection of *N* instances, which are characterized by a tuple *(X,y)*, where *X* is a set of attributes and *y* is the attribute that indicates the class label [11]. Classification has two main purposes: (i) descriptive modeling that explains the behavior between objects of different classes, and (ii) predictive modeling used for assigning a class label of an unknown instance.

For the problem of skin cancer detection, the classification task objective is to assign an input object x_{input} to one of the binary outputs *malignant tumor* or *benign tumor*. Input x_{input} possesses the set of features extracted during the lesion segmentation and color segmentation stages (see Section 3 for details).

Nonetheless, single classification algorithms do not always provide the most accurate predictions. To overcome this limitation, an ensemble of classifiers is proposed. Ensembles of classifiers are thought to outperform individual classifiers because they allow to filter out hypothesis that are not accurate due to a small training set; ensembles of classifiers help overcoming problem of local optima; different classifiers expand the universe of available target functions *f* [12].

The ensemble of classifiers that we developed (named MAEoC since it is developed following the Multi-Agent paradigm) acts on two premises: (i) the performance of base classifiers and (ii) the communication of hits (H) and failures (F) obtained by base classifiers. The design of MAEoC can be consulted in [13]. The MAEoC works according to the following algorithm:

Iteration t = 0

> *m* classifiers, $m > 2$ are recruited and *m* classifier agents are started.
>
> Dataset *D* containing features is broadcasted to classifier agent$_i$, for all *i*, $i = 1, \ldots m$.

Classifier$_i$ performs a ten fold cross-validation. F-Measure$_i$ is calculated.

Classifier$_i$, for all i, $i = 1...m$, constructs two subsets. Subset H_i contains objects correctly classified; subset F_i contains objects incorrectly classified.

Iteration t = 1

Aggregated sets *AH* and *AF* are formed. $AH = \cup_i H_i$; $AF = \cup_i F_i$.

classifier$_{m+1}$, is started, based on the highest F-Measure$_i$ obtained at $t=0$.

Classifier$_{m+1}$ is trained with set *AF*. F-Measure C_{m+1} is obtained by ten fold cross validation on *AF*.

Classifiers$_{1,\ldots,\ m}$ are trained with set *AH*. F-Measures$_{1,\ \ldots,\ m}$ are obtained by ten fold cross-validation on *AH*.

Iteration t = 2

Classifiers$_{1,\ldots,.m+1}$ are given weights according to their updated F-Measure at $t=1$. Weighted voting is used to reach a final conclusion.

The algorithms that form the ensemble of classifiers are: a Multilayer Perceptron (MLP) [14], a Naive Bayes classifier [15], a decision tree C4.5 [16], a K nearest-neighbor [17], and a support vector machine [18].

Classification metrics are obtained to measure the performance of the ensemble. We compare the performance of the MAEoC with those of the individual classifiers that make it up. The MAEoC is also contrasted with classical aggregation methods such as Bagging [19], Boosting [20], and Stacking [21].

2.4.1 Classification Metrics

The following metrics are employed to evaluate the performance of classifiers: sensitivity, specificity, precision, recall, true positive rate, false positive rate, and the area under the ROC curve (AUC) and F-Measure.

Firstly, to determine how well the segmentation algorithm performs, it requires a ground truth (GT) image, which is determined by drawing manually the border around the lesion. Using a GT image, the exclusive disjunction (XOR) operation is calculated [22]. For dermoscopy images, *sensitivity* measures the proportion of actual lesion pixels that are correctly identified as such. *Specificity* measures the proportion of background skin pixels that are correctly identified. Generalizing:

- TP (true positive). Objects that are correctly classified as the object of interest.
- FP (false positive). Objects that are incorrectly identified as the object of interest.

- TN (true negative). Objects that are correctly identified as not being the object of interest.
- FN (false negative). Objects that are incorrectly identified as not being the object of interest.

Sensitivity and specificity are given by:

$$sensitivity = TP/(TP + TN) \tag{5}$$

$$specificity = TN/(FP + TN) \tag{6}$$

The *precision* of a classifier is the fraction of tuples that were correctly classified as positive from all the tuples that are actually positve. Precision is defined as follows:

$$P = precision = (TP)/(TP + FP). \tag{7}$$

Recall is the fraction of positive tuples that were correctly classified as positive:

$$R = recall = (TP)/(TP + FN). \tag{8}$$

We also apply the *Receiver Operating Characteristic* (ROC) analysis. Points of the ROC curve are obtained by sweeping the classification threshold from the most positive classification value to the most negative. A quantitative summary of the ROC curve is called the area under the ROC curve (AUC).

Classification is also quantified by the *F-measure*, defined as the weighted harmonic mean of its precision and recall:

$$F = 2PR/(P + R). \tag{9}$$

The F-measure assumes values in the interval *[0,1]*. It is *0* when no relevant instances have been retrieved, and is *1* if all retrieved instances are relevant and all relevant instances have been retrieved. Experimental results are given in the following section.

3. Experimental Results

This section provides the results of determining the occurrence or not of skin cancer on 147 images of the ISIC repository. All of the images are stored as 24-bit color image in JPEG format. They have already been characterized with both, Ground Truth and the diagnosis given by the expert. Even though we do not contemplate the pre-processing of the images as part of the proposed method, occlusions and artifacts were removed in all the images by applying the DullRazon algorithm [23].

Figures 2 and 3 show the lesion segmentation process. In them, Figure *(c)* illustrates the result of the segmentation after applying the W-FCM algorithm on figures *(a)*. When comparing the final result *(c)* with figure *(b)* (Ground Truth of lesion), the W-FCM displays higher precision and accuracy.

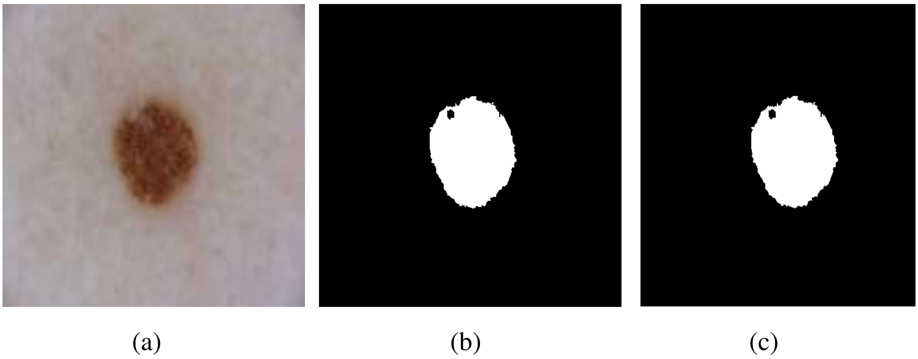


Figure 2
(a) Image ISIC_0000261 skin lesion benign accord to data set (b) Ground Truth as delineated by an expert (c) Segmentation with the W-FCM method. The following metrics are obtained: Precision = 0.99317, Sensitivity = 0.9998, Specificity = 0.92388, Accuracy = 0.99355

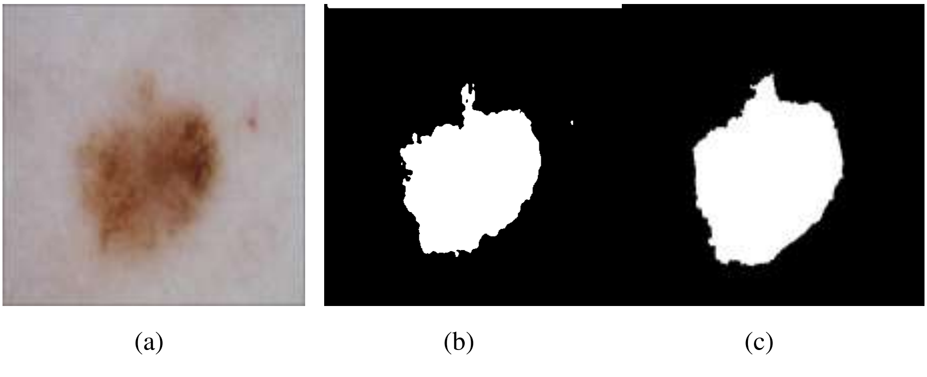
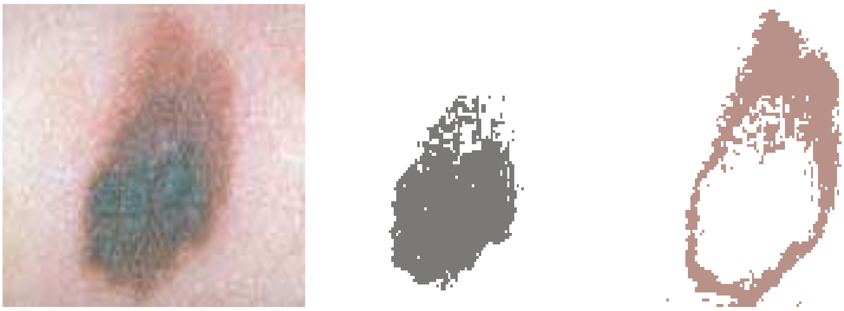


Figure 3
(a) Image ISIC_0000054 skin lesion malign according to data set (b) Ground Truth (c) Segmentation with the W-FCM method. The following metrics are obtained: Precision = 0.94664, Sensitivity = 0.98260, Specificity = 0.78957, Accuracy = 0.94239

As for the color segmentation, we exemplify this stage with the following digital images. Figure 4(a) shows a digital image of dermoscopy taken form the ISIC repository. After applying a Kohonen Map to discover the most representative values, Figures 4(b) and 4(c) are obtained. For this particular image only two colors were discovered. The result of this stage is the obtainment of the RGB values of each of the segmented images. Needless to say that such values are added to the final features vector.



(a) (b) (c)

Figure 4

Illustration of the color segmentation stage. (a) Original digital image taken from ISIC. (b) Example of color segmentation of Figure 4(a) by using a Kohonen Map. (c) Second color found in Figure 4 (a) by using a Kohonen Map

Altogether, the vector of extracted features contains the assymmetry of the lesion, the eccentricity value, the area of the lesion, all the features of GLCM (i. e. autocorrelation, energy, entropy, dissimilarity), and the RGB values of the found colors, according to the ABCD guide. This complete vector of features is the actual input to the MAEoC.

The performances of both, the MAEoC and its constituting classifiers are presented in the following three tables. Metrics were obtained after running a ten-fold cross validation.

Table 1 presents classification metrics when the vector consists in the following features: Colors quantity, texture features and morphology features. Table 2 presents the performance when, in addition to the features that were used to obtain Table 1, the RGB values of each found color are added to the vector of features. Finally, Table 3 presents the performance of the classifiers when the area of each found color is included in the features vector.

Table 1
Classification results using number of colors, texture features and morphology features

Classifier	Accuracy	ROC	Average Precision	F-Measure
Multi Layer Perceptron	0.631	0.513	0.628	0.630
Support Vector Machine	0.77	0.5	0.594	0.671
Decision Trees	0.708	0.408	0.624	0.657
Naive Bayes	0.604	0.465	0.646	0.622
KNN; k = 3	0.715	0.428	0.61	0.652
KNN; k = 5	0.77	0.9	0.715	0.694
AdaBoost	0.729	0.498	0.618	0.66
Bagging	0.729	0.52	0.618	0.66
Stacking	0.77	0.464	0.594	0.671
MAEoC	**0.888**	**0.789**	**0.903**	**0.875**

Table 2
Classification results using number of colors, texture features, morphology features, and RGB values obtained in the color segmentation phase

Classifier	Accuracy	ROC	Average Precision	F-Measure
Multi Layer Perceptron	0.666	0.492	0.652	0.659
Support Vector Machine	0.77	0.5	0.594	0.671
Decision Trees	0.673	0.473	0.619	0.643
Naive Bayes	0.611	0.444	0.634	0.622
KNN; k = 3	0.729	0.488	0.638	0.669
KNN; k = 5	0.77	0.441	0.712	0.683
AdaBoost	0.729	0.405	0.618	0.66
Bagging	0.75	0.508	0.639	0.672
Stacking	0.77	0.464	0.594	0.671
MAEoC	**0.84**	**0.716**	**0.868**	**0.805**

Table 3
Classification results using number of colors, texture features, morphology features, RGB values obtained in the color segmentation phase, and the area of the lesion

Classifier	Accuracy	ROC	Average Precision	F-Measure
Multi Layer Perceptron	0.68	0.484	0.681	0.681
Support Vector Machine	0.77	0.5	0.594	0.67
Decision Trees	0.631	0.428	0.611	0.621
Naive Bayes	0.611	0.425	0.649	0.628
KNN; k = 3	0.729	0.467	0.638	0.669

KNN; k = 5	0.756	0.471	0.674	0.686
AdaBoost	0.701	0.449	0.581	0.636
Bagging	0.736	0.517	0.588	0.654
Stacking	0.77	0464	0.594	0.671
MAEoC	**0.84**	**0.668**	**0.868**	**0.805**

It can be noticed that the best metrics correspond, in these three cases, to MAEoC. In the following section, we present a comparison of how the MAEoC fares when comparing its performance with the related work presented in the literature.

However, it is worth noticing that the performance of the MAEoC decreases slighty when the number of features of the input vector increases. This effect can be seen on the accuracy values reported in Table 1 (0.88) and Tables 2 and 3 (0.84). Also, the area under the ROC curve decreases from 0.789 in Table 1, to 0.716 in Table 2, and 0.668 in Table 3.

One possible explanation refers to the nature of the lesion under analysis. Since melanin is a substance determinant in the pigmentation of the skin, a malign melanoma changes the naturally occurring color of the skin, as well as its texture. In this sense, data such as the area of the lesion might as well be of no relevance. That is to say, the area of the lesion could be small or large and yet the effects on melanin are noticeable changes on color and texture. More experimentation is needed, though.

4. Comparison with other Methods

Computer Aided Diagnosis (CAD) systems for malignant melanoma have been developed rather recently. Although not all of the systems necessarily include the same processes, the following steps are common: image pre-processing, feature extraction, color interpretation, classification and lesion evaluation. A review of such systems is given in [24], and we selected five systems displaying the best performance. A summary is given in Table 4.

The classification methods that have been used in those five top-performers employ □-Nearest Neighbor, Decision Trees, Support Vector Machines, Artificial Neural Network (ANN), Neuro-Fuzzy, Fuzzy C-Means, and Naive Bayes. The best results are obtained when hybrid techniques are employed. Even though the ensemble of classifiers that we developed is not strictly a hybrid system, it does benefit from using multiple classifiers for the detection of malignant lesions.

Moreover, these hybrid systems are trained with a large set of features extracted from the digital images. In the system we present, the features vector also displays a high dimensionality, although the quantity of images we process is not large (147 images).

We consider that more details should have been given in those reports. For instance, the source of the images is not made explicit as opposed to the images we use, which are available to a broad community (the ISIC repository). Neither is it clear whether the images employed in those CAD systems were pre-processed in order to eliminate artifacts.

Table 4
Comparison of the proposed method with other approaches

Autor	Dataset	Pre processing	Feature extraction method	Classifier	Detection performance
(Sheha et al) [25]	102 dermoscopy Atlases	Resizing and Color space Transformation	GLCM	Multi-Layer Perceptron	Accuracy =%92
(KumarJain & Jain) [26]	From different sources	Image contour Tracing Algorithm	Discrete Wavelet Transform	Clustering & k-Nearest Neighbors	Accuracy = 92% Accuracy = 95%
(Elgamal) [27]	From a digital camera with dermoscope	Gaussian - Median Filter	Principal Component Analysis. Discrete Wavelet Transform	Artificial Neural Networks, k-Nearest Neighbors.	Accuracy = 95% Accuracy = 97.5%
(Mengistu) [28]	Dermquest/ Dermnet	Median Filtering	GLCM and color features	Self Organizing Maps and Radial Basis Functions	Accuracy = 96.15%
(Immagulate & Vijaya) [29]	Dermnet/ Dermofit	Image resizing	Color and Texture Features	Support Vector Machine	Accuracy = 86%
Proposed System	ISIC repository	Artifact removal with Razor algorithm	Fuzzy Discrete Wavelet Transform	Multi-Agent ensemble of Classifiers	Accuracy = 88%

5. Conclusions and Future Work

One of the main problems to obtain a good performance regarding segmentation and classification in dermoscopy images refers to the proper selection of the features that characterize a skin lesion. To solve this problem, we propose a method consisting in the following stages: lesion segmentation, feature extraction, color extraction, and learning. An intelligent system was developed mirroring the mentioned steps.

Particularly, we have proposed the extraction of the following features: asymmetry, eccentricity, features of the well-known method called Grey Level Co-occurrence Matrix, and the color content of the lesion, for which an Ensemble of Clusterers is used. Nevertheless, the feature extraction is only one step in the automatic detection of skin cancer. The other major task for an intelligent system is learning from the combination of feature values that represent either a malignant tumor or a benign lesion. The learning and classification stage is performed by an ensemble of classifiers called MAEoC. As it is mentioned in the literature, ensembles of classifiers take advantage of the combined results of different classifiers. MAEoC is formed by a Multi-Layer Peceptron, a decision tree, a K nearest-neighbor, a Naïve – Bayes and a Support Vector Machine. The performance metrics indicate that MAEoC displays a better performance than single classifiers. However, aggregation methods such as stacking and bagging fare at least as well as the MAEoC.

One of the limitations of the results we present refers to the number of images that were analyzed. We are embarked in using a larger database than the 147 images that were processed in order to obtain the results given along the present article. Also, we are experimenting with more segmentation techniques, and algorithms that make adaptable both of the ensembles.

Another improvement is the addition of more relevant information to the features vector such as the ratio of the color area to the lesion area once the lesion has been separated from the original image. Regarding color extraction, we also envision the discovery of colors in a different color space than the RGB to include data such as hue and brightness.

Acknowledgements

The authors thank Universidad Autónoma del Estado de Hidalgo and SEP - PRODEP for the financial support. The authors also want to thank Gonzalo Chávez Fragoso and Jaime Calderón for their time dedicated to program some modules of the system.

References

[1] H. Castillejos, V. Ponomaryov, L. Niño de Rivera and V. Golikov. Wavelet transform fuzzy algorithms for dermoscopic image segmentation.

Computational and Mathematical Methods in Medicine. Vol. 2012, pp. 11-21, 2012

[2] H. Castillejos, V. Ponomaryov and R. Peralta-Fabi. Image segmentation in Wavelet Transform Space implemented on a DSP. Proceedings of the SPIE 8437. Real – time image and video processing. Vol. 8437, April 2012

[3] S. Mallat. A theory for multi-resolution signal decomposition: The Waveltet representation. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 11, No. 7, pp. 338-353, 1989

[4] V. Kravchenko, H. Meana and V. Ponomaryov. Adaptive digital processing of multidimensional signals with applications. Editorial FizMatLit, Kiev. 2009

[5] S. Mimaroglu and E. Erdil. An efficient and scalable family of algorithms for combining clusterings. Engineering Applications of Artificial Intelligence. Vol. 26, pp. 2525-2539, 2013

[6] I. Maglogiannis and D. Kosmopoulos. Computational vision systems for the detection of malignant melanoma classifiers using ROC curves. Oncology Reports. Vol. 15, pp. 1027-1032, 2006

[7] D. Ruiz, V. Berenguer, A. Soriano et al. A decision support system for the diagnosis of melanoma: a comparative approach. Expert Systems with Applications. Vol. 38, pp. 15217-15223, 2011

[8] T. Tanaka, S. Torii, I. Kabuta et al. Pattern classification of nevus with texture analysis. EEJ Transactions on Electrical and Electronic Engineering. Vol. 3, pp. 143-150, 2008

[9] C. Serrano and B. Acha. Pattern analysis of dermoscopic images based on Markov random fields. Pattern Recognition. Vol. 42, pp. 1052-1057, 2009

[10] M. Sadegui, M. Razmara, T. Lee et al. A novel method for detection of pigment networks in dermoscopic images using graphs. Computerized Medical Images and Graphics. Vol. 35, pp. 137-143, 2011

[11] Pang-Ning Tang, Michael Steinbach and Vipin Kumar. Introduction to Data Mining. Addison – Wesley, 2006

[12] Michael Wozniak, Manuel Grana and Emilio Corchado. A survey of multiple classifier systems as hybrid systems. Information Fusion. Vol. 16, pp. 3-17, 2014

[13] Jaime Calderón, Omar López-Ortega and Félix Castro-Espinoza. A multi-agent ensemble of classifiers. Advances in Artificial Intelligence and Soft Computing: 14th Mexican International Conference on Artificial Intelligence, MICAI 2015, Cuernavaca, Morelos, Mexico, October 25-31, 2015, Proceedings, Part I, pp. 499-508, 2015

[14] Christopher M. Bishop. Neural networks for pattern recognition. Oxford University Press, 1995

[15] Nir Friedman, Dan Geiger and Moises Goldszmidt. Bayesian networks classifiers. Machine Learning. Vol. 29, No. 2-3, pp. 131-163, 1997

[16] J. Ross Quinlan. C4:5: Programs for machine learning. Elsevier, 2014

[17] T. Cover and P. Hart. Nearest neighbor pattern classification. IEEE Transactions on Information Theory. Vol. 13, No. 1, pp. 21-27, 1967

[18] Corrina Cortes and Vladimir Vapnik. Support-vector networks. Machine learning. Vol. 20, No. 3, pp. 273-297, 1995

[19] Leo Breiman. Bagging predictors. Machine Learning. Vol. 24, No. 2, pp. 123-140, 1996

[20] Yoav Freund and Robert E. Schapire. A decision – theoretic generalization of on-line learning and an application to Boosting. Journal of Computer and System Sciences. Vol. 55, pp. 119-139, 1997

[21] Leo Breiman. Stacked regressions. Machine Learning. Vol. 24, No. 1, pp. 49-64, 1996

[22] N. Lachiche and P. A. Flach. Improving accuracy and cost of two-class and multi-class probabilistic classifiers using ROC curves. In Proceedings of the ICLM, Vol. 2003, pp. 1027-1032, 2006

[23] T. Lee, V. Ng, R. Gallagher, A. Coldman and D. McLean. DullRazor: A software approach to hair removal from images. Computers in Biology and Medicine. Vol. 27, pp. 533-543, 1997

[24] M. A. Arasi, , E. S. A. El-Dahshan, E. S. M. El-Horbaty, and A. B. M. Salem. Malignant Melanoma Detection Based on Machine Learning Techniques: A Survey. Egyptian Computer Science Journal. Vol. 40, No. 3, pp. 1-10, September 2016

[25] M. Sheha, M. Mabrouk and A. Sharawy. Automatic detection of melanoma skin cancer using texture analysis. International Journal of Computer Applications. Vol. 42, No. 20, pp. 22-26, 2012

[26] Y. Kumar Jain and M. Jain. Comparison between different classification methods with application to skin cancer. International Journal of Computer Applications. Vol. 53, No. 11, pp. 18-24, 2012

[27] M. Elgamal. Automatic Skin Cancer Images Classification. International Journal of Advanced Computer Science and applications. Vol. 4, No. 3, pp. 287-294, 2013

[28] A. D. Mengistu. Computer Vision for Skin Cancer Diagnosis and Recognition using RBF and SOM. International Journal of Image Processing. Vol. 9, pp. 311-319, 2015

[29] I. Immagulate and M. S. Vijaya. Categorization of Non-Melanoma Skin Lesion Diseases Using Support Vector Machine and Its Variants. International Journal of Medical Imaging. Vol. 3, No. 2, pp. 34-40, 2015

Customer Analysis via Video Analytics: Customer Detection with Multiple Cues

Tatpong Katanyukul and Jiradej Ponsawat

Faculty of Engineering, Khon Kaen University,
123 Mitraparb road, Khon Kaen, Thailand, 40002
tatpong@kku.ac.th, jiradej@kku.ac.th

Abstract: In addtion to security purposes, closed circuit video camera usually installed in a business establishment can provide extra customer information, e.g., a frequently visited area. Such valuable information allows marketing analysis to better understand customer behavior and can provide a more satifying service. Underlying customer behavior analysis is customer detection that usually serves as an early step. This article discusses a complete automatic customer behavior pipeline in detail with a focus on customer detection. Conventional customer detection approach relies on one source of decision based on multiple small image areas. However, a human visual system also exploits many other cues, e.g., context, prior knowledge, sense of place, and even other sensory input, to interpret what one sees. Accounting for multiple cues may enable a more accurate detection system, but this requires a reliable integration mechanism. This article proposes a framework for integration of multiple cues for customer detection. The detection framework is evaluated on 609 image frames captured from a retailer video data. The detected locations are compared against ground truth provided by our personnel. Miss rate to false positive per window is used as a performance index. Performance of the detection framework shows at least 42% improvement over other control treatments. Our results support our hypothesis and show the potential of the framework.

Keywords: customer detection, human detection, video analytics, hot zone visualization, multiple-cue integration, global-local inference integration, ensemble framework

1 Introduction

The closed circuit video camera is becoming more common in many businesses and household establishments, mostly for security purposes. To gain an extra value out of a camera system, many studies investigate utilization of video data installed in business establishments, such as shopping malls and supermarkets, for customer behavior analysis. Customer behavior analysis via video analytics has automatic customer detection as its essential part.

Conventional customer detection relies solely on visual information contained within a limited area of the window. This approach simplifies a detection process and enables the use of a regular classifier, which takes input of a small and fixed size, in a detection problem involving images of larger and various sizes. However, this approach leads to limited inference capability, as discussed in Torralba [1] and Mottaghi et al. [2]. Contextual information and prior knowledge are natural cues. Along with focal-point visual information of an object itself, human visual system employs a collective sense of scene, other surrounding objects, prior knowledge of relation among different types of objects, dynamic nature of objects, and continuity of objects in the perceiving stream to interpret current visual perception. Accounting for auxiliary information can provide viable additional cues for a more accurate automatic customer detection, as well as benefiting object detection in general. Given various sources of information, a reliable integration mechanism is essential. Such a mechanism may also enable an ensemble of multiple models, which in turn provides a key to adjust a global inference system with a local sense to better fit a specific task. Our work proposes an integration framework that can accommodate various types of cues under customer detection settings. A general customer detection approach and hot zone customer analysis based on video information are also discussed.

Section 2 provides a review of previous studies on customer analysis via video analytics. Section 3 discusses an approach for customer detection and hot zone analysis. Section 4 discusses a framework for integration of multiple cues. Section 5 discusses our experiments and results and also provides discussion, conclusions, and potential directions.

2 Literature Review

Utilization of video data from closed circuit camera for customer behavior analysis is of great interest in business and academia [3][4][5]. Customer behavior analysis via video analytics employs and integrates techniques from various related fields, e.g., motion detection [6][7], pedestrian detection [8][9], object detection and recognition [10][11][12], object tracking [14][15], and activity recognition [16][17][18].

Popa et al. [3] studied and designed a system to detect, track, and analyse customers and their behavior in a large business establishment, e.g., a shopping mall, or a supermarket. They investigated a dedicated system designed specifically for customer behavior analysis, not a value-added security camera system. It employed various types of sensors, including high-angle cameras to locate customers, face-level cameras to read facial expression, microphones for verbal information, and dynamic Bayesian network for data fusion. To locate customers, Popa et al. [3] used background subtraction to detect customers in the entry points and then tracked them with mean shift algorithm [15].

Background subtraction was a widely-used method to detect motion or a moving object. To detect moving objects, a background model was subtracted from an image under question. The background model itself is an image similar to the underlying image, but without the objects. Therefore, the difference between the two images revealed the moving objects. There were several methods to derive a background model. Popa et al. [3] did not provide the details of how they derived the background model. Popa et al. [4] extended [3] by adding higher level analysis of customer's action and behavior. Both works [3][4] conformed to a general approach of customer behavior analysis via video analytics.

Ko [19] summarized that customer behavior analysis via video analytics consisted of video acquisition, object and motion detection, object classification, object tracking, behavior and activity analysis, person identification, data fusion, and control, alarm, and visualization. Ko identified background subtraction, temporal differencing, and optical flow methods as the main approaches for motion detection. Ko credited simplicity as the main reason for popular use of background subtraction. Object classification or object category recognition was extensively studied [20]. Semantic segmentation [2] and image description [13] were closely related fields. Object classification referred to an approach to identify pre-defined categories of objects in an image. Along with sliding window technique, object classification could be used to locate positions of objects in an image. Once the object of interest had been located, it could be tracked more efficiently with an object tracking method, e.g., a mean shift algorithm [15]. To get good tracking performance, Yilmaz et al. [14] recommended that a good selection of features to represent an object of interest, online selection of discriminative features, and exploitation of prior knowledge and contextual information were among the key factors.

In addition, pedestrian detection research also worked on many similar key challenging issues. Dollar et al. [21] followed a general approach for object detection. They proposed a scheme to perform less image scaling, while delivering a similar detection quality. They also emphasized that, rather than using pixel intensities directly, employing image features as an input for classification was a key factor for high quality detection. Dollar et al. [9] noted that histogram of oriented gradients [22] was a widely-used choice of image features for pedestrian detection. Dollar et al.[9] also noted that occlusion was still a major issue for automatic pedestrian detection. They speculated that motion features [23], inference of detection from consecutive frames, contextual information, and combination of various types of visual features could mitigate the issue.

Comparing customer detection to pedestrian detection, while pedestrian detection often involved a moving camera, changing background, and variably lighting conditions, customer detection usually involved a stationary camera and a relatively constant background. In addition, an area where customers entered or exited a store could commonly be identified. This prior information could be exploited. Regarding an issue of image distortion, pedestrian detection often involved a focal point viewing which delivered a lower image distortion than ones normally found in customer detection. Customer detection often involved a wide-angle view, from a camera installed at the corner on the ceiling. Image distortion and odd angle-view posed a unique challenging issue, specific to customer detection. Simple multi-scaling alone might not be adequate to handle the issue. Another difference was that customer detection was usually a preliminary part of a pipeline that ultimately delivered customer analysis at the end. In addition to high level customer behavior analysis [3], Connell et al. [5] commented people counting and hot zone were among the most common end results.

Our study discusses a pipeline of customer behavior analysis, from customer detection to hot zone map visualisation, as well as a close investigation on a framework for integrating multiple cues. Our study implements the pipeline based on common practice in video processing [19] and object detection [22].

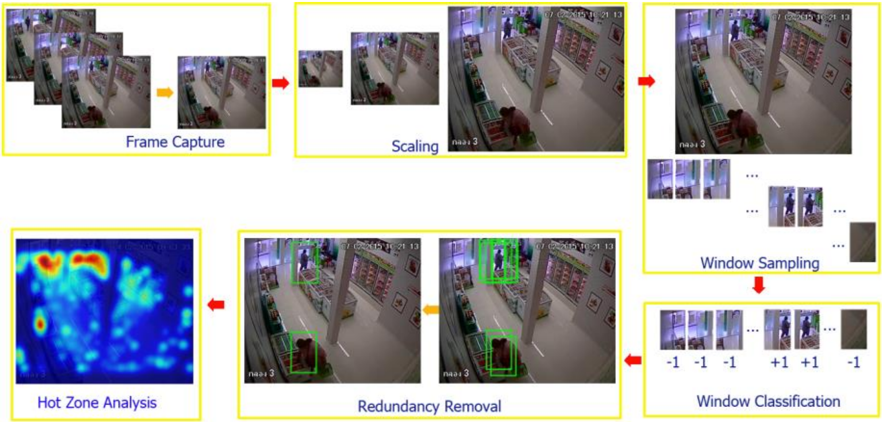


Figure 1

Customer analysis pipeline: (1) image frames are captured from video data, (2) each image frame is scaled to multiple sizes, (3) small cropped images, called “windows”, are sampled, (4) each window is classified to either +1 or −1, when a positive one indicates a window containing a customer and a negative one indicates no customer in a window, (5) multiple positive adjacent windows for the same customer are redundancies and most of them are removed, leaving only one detection for one customer, and (6) detection results of image frames in the video are summarized into a hot zone map.

3 Customer Analysis Pipeline

One of the most common customer behavior video analytics is hot zone analysis [5]. Hot zone map shows frequently visited spots in the area of interest. Fig. 1 shows a working pipeline, starting from video data and finishing as a heat map. Firstly, video data is turned into a series of image frames, so that the task can be simplified to multiple processing on each image frame (Frame Capture in Fig. 1). Then, customer detection is performed for each image frame (Collective process of Scaling, Window Sampling, Window Classification, and Redundancy Removal in Fig. 1). Each frame is scaled to multiple sizes, so that objects at different distances appearing in different sizes have fair chances to be detected (Scaling). Then, at each scale, windows—small fixed-size image patches— are sampled from the scaled image frame (Window Sampling). Each window is passed through a classifier to decide whether it contains visual cues indicating presence of a customer (Window Classification). Once a window is classified positive (indicating presence of a customer), a set of coordinates of the top-left and bottom-right corners of the window is recorded as a detected location. In practice, it is likely that presence of a customer may trigger multiple positive windows around the location of one's presence. Multiple positive windows indicating the same presence are redundant. Only one positive window is needed and the other redundant windows are discarded (Redundancy Removal). The result from the redundancy removal step is a collection of detected locations in an image frame. This is a detection result. Detection results from multiple image frames indicate frequencies of locations inside a retailer store that customers have visited. The visiting frequencies are mapped to colors to provide a hot zone map, marketing personnel can use for behavior, marketing, and store layout analyses.

Window Sampling. Our window sampling step is implemented by sliding window scheme [24]. A sliding window scheme starts by taking a sample from a top-left corner of an image frame. Then, it requires taking a sample a step size right from the previous one until reaching the right end, and then back to take the next sample from a position on the left end but a step size down from the previous one. It repeats the procedure until an entire image frame is exhausted. A sequence of sampled windows appear like a series of cropped images seen from a fixed size viewing area that slides through the image from top-left to bottom-right row by row, hence the scheme is named sliding window.

Denote a scaled image frame (in form of a matrix of pixel intensities) $F \in \boldsymbol{I}^{C \times R}$ and a window $W_{ij} \in \boldsymbol{I}^{A \times B}$, where pixel intensity $\boldsymbol{I} = \{0, 1, 2, ..., 255\}$, (C,R) and (A,B) are frame and window sizes, respectively. Given step size of (a,b), sliding window is a mapping function, $S\colon F \rightarrow \{W_{ij}\}$, for $i = 0, ..., \lfloor (C-A)/a \rfloor$ and $j = 0, ..., \lfloor (R-B)/b \rfloor$. Each window $W_{ij} = [w_{p,q}^{(i,j)}]$, $p = 1, ..., A$ and $q = 1, ..., B$, is a submatrix of $F = [f_{m,n}]$, $m = 1, ..., C$ and $n = 1, ..., R$, where $w_{p,q}^{(i,j)} = f_{a \cdot i+p, b \cdot j+q}$.

Although sliding window is simple to implement and it guarantees complete frame coverage, it requires considerable computational cost. In order to speed up the system, a detection proposal method can be used instead. Detection proposal method employs the idea of cascading. A weak but fast classifier is applied to initially decide if the window is a good candidate. A failed window is discarded. A passed window gets to the next round with a stronger but slower classifier. The mechanism is that a deserved candidate passes through a series of classifiers to reach a positive label, while other candidates are discarded along the way. Therefore, the highest quality classifier, which usually is very slow, only performs on a few worthy candidate windows. Hosang et al. [10] provided formal investigation on detection proposal methods.

Window Classification. Given a window W_{ij}, a classifier determines if the window contains visual features of a customer. Window classification is to map W_{ij} to one of the decisive labels, in our case, positive label (+1) indicating a detected customer or negative label (−1) indicating no detection. Regarding common practice, classification in object or pedestrian detection usually takes image features, rather than pixel intensities, as an input. The exception may be later development of deep learning [20]. Features represent an original input in a way that allows a task to be achieved easier than operating directly on the original input. Milestones of object detection development tie strongly with development of visual features: Haar features [24], Histogram of Oriented Gradient (HOG) [22], and a Bag of Visual Words [25]. Some later features are built on previously well-developed ones. A deformable model [26] employs HOGs as its building blocks. An ensemble approach of Malisiewicz et al. [27] predicts a class based on combined predictions from multiple linear Support Vector Machines (SVMs), each trained on only one example. Later development takes a deep learning approach [11][12][13]. Although, most deep learning object detections do not require image features and can directly take image intensities as input. Visual features are constructed internally during the learning process of deep networks. Despite great potential, a deep learning approach requires considerable resources, compared to an explicit feature-based approach.

Following common practice [22], our study implements window classification in two successive stages, (1) feature mapping and (2) classification. That is, (1) window W_{ij} is mapped to visual feature vector X_{ij} and then (2) X_{ij} is mapped to decisive label $y_{ij} \in \{-1,+1\}$. Histogram of Oriented Gradient (HOG) [22] is used for our feature mapping and Support Vector Machine (SVM) [28] is used for our feature classification.

It should be noted that the approach presented here is only to detect presence and a location of a human in an image. It does not distinguish a high-level concept that whether the detected human is actually a customer or a store staff. Distinction between a customer and a staff is not only crucial to accurate customer behavior analysis, it may also provide an insight bridging a low-level concept, e.g., an activity, to a high-level concept, e.g., a role. To distinguish between a customer and a staff, a pattern of a moving trajectory of the detected human can provide an essential cue. However, with its depth and implication, research on this high-level notion deserves a dedicated study on its own right and it is beyond our current scope of this investigation.

Histogram of Oriented Gradient (HOG). HOG is a mapping function, $H: W \rightarrow X$, when $W \in \boldsymbol{I}^{A \times B}$ is a matrix of pixel intensities and $X \in \boldsymbol{R}^{D}$ is a HOG-feature vector. Generally, a size of X is much smaller than that of W, i.e., $D << A \times B$. There are many types of features for visual input. Good features emphasize relevant information to the intended task and mumble noise or irrelevant information. HOG [22] is among the most widely-used feature families for object and pedestrian detection. The assumption underlying HOG is that distribution of image gradients provides a good cue to an object's shape and presumably identification of an object. Votes of image gradients are collected within a small area, called a "cell." Each cell has K votes. Each vote is for each of K pre-defined orientations. A vote can be defined as a sum of magnitudes of all gradients locating inside the cell and having the corresponding orientation. Then, to mitigate shadow and variant lighting, cell votes are normalized locally. That is, cell votes are spatially grouped into a block. Therefore, a block of N_c cells has $N_c \cdot K$ cell votes. All cell votes are normalized within a block. Blocks are defined in an overlapping manner to allow each cell to be normalized under multiple surroundings. Finally, all normalized cell votes are collected to make up a complete set of HOG features. Our investigation follows Dalal and Triggs [22], using 64x28-pixel detection window and HOG with a cell size of 8x8 pixels, 9 orientations (spacing evenly in 0^{o}–180^{o}), a block size of 2x2 cells, and block spacing of 8 pixels in either direction. It should be noted that combination of HOG and other types of features, e.g., color similarity score (CSS) [23], may lead to better discriminative performance.

Support Vector Machine (SVM). A classifier determines a class label y for a given feature vector X. Our study employs Support Vector Machine (SVM) [28], one of the most widely-used classifiers. SVM is a discriminant function, which directly maps $X \in \boldsymbol{R}^{D}$ to $y \in \{-1,+1\}$ and does not provide related probability estimation. As most supervised machine learning methods, SVM has two operating modes, training and prediction modes. In a training mode, SVM uses training data to lay on a projection space in order to find a decision hyperplane that best separates the training data based on corresponding labels. The decision hyperplane is then used in a prediction mode to decide a class label for a given input.

Specifically, the training stage is formulated as a constrained optimization problem, $min_{w,b,\xi}\ ½\ w^{T} \cdot w + C \cdot \Sigma_i\ \xi_i$, s.t. $y'_i \cdot (w^{T} \cdot \phi(X'_i) + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$, for all $i = 1, ..., N$. Vector w and scalar b are parameters characterizing a decision hyperplane. Variables ξ_i's are slack variables to allow SVM to compromise outlying datapoints. User-specified parameter C is to control a degree of relaxation of ξ_i's. A higher value of C penalizes training misclassification heavier, which results in forcing SVM to reduce its misclassified training examples. A proper value of C leads to a good performing SVM. Too large value of C may lead to overfitting to training data and loss of prediction generality. Vector X'_i and scalar y'_i represent respectively features and correct class label of the i^{th} datapoint in a training dataset of size N. Function $\phi(\cdot)$ is a projection function, intended to map input features onto a multi-dimensional space that eases the data separation. However, instead of directly solving the minimization problem in its original form, it is more efficient to solve its dual form, $min_a \Sigma_i a_i - ½ \Sigma_i \Sigma_j a_i a_j y'_i y'_j k(X'_i , X'_j)$, s.t. $\Sigma_i y'_i a_i = 0$ and $0 \leq a_i \leq C$, for all $i =1, ..., N$. Variables a_i's are dual variables and kernel function $k(u,v) = \phi(u)^T \phi(v)$. Our work uses a radial basis kernel, $k(u,v) = exp(-\gamma||u-v||^2)$, where γ is a user-specified parameter. Once SVM is trained (all a_i's are determined), SVM can be used in a prediction mode. Given input vector X, its class label y is predicted by $y = sign(s)$, when

$$s = \sum_i a_i y'_i k(X, X') + b. \tag{1}$$

Variable s, called "a decision score," indicates a degree of class likeliness. A decision score is, related but, nor a probability nor a distance to a decision boundary. Vectors X'_i's and scalars y'_i's are of training data points (or selected data points, called "support vectors" [29]). Parameter $b = |M|^{-1} \Sigma_{i \in M} \{ y'_i - \Sigma_{j \in S} a_j y'_j k(X'_i , X'_j)\}$ when $|M|$ is a size of set M, $M = \{i: 0 \leq a_i \leq C\}$ and $S = \{i: a_i > 0\}$, and $sign(u) = 1$ when $u > 0$, otherwise $sign(u) = -1$.

Fundamentally, SVM is a binary classifier. However, multiclass capability can be achieved by an extension, such as one-against-one approach [30] that builds a multiclass classifier from multiple binary classifiers. Our current problem settings require only a binary classifier. As a discriminant function, SVM in its original development does not provide estimated probability, however its popularity has attracted extensive studies to extend SVM capability enabling SVM to provide estimate probability [29].

Redundancy Removal. Once positive windows have been identified in Window Classification, all locations of positive windows are recorded as detected bounding boxes. Adjacent windows may be triggered positive for the same customer and that causes redundant detected bounding boxes. Out of all bounding boxes corresponding to the same customer, only one bounding box is reported and others are suppressed.

Non-maximum suppression [27] is a mechanism to remove redundant bounding boxes. Given a threshold, any pair of bounding boxes with an overlapping area larger than the threshold is considered redundant and one of them should be suppressed. A simple way to perform non-maximum suppression is to arbitrarily pick one bounding box from a pair and suppress the other. Arbitrary selection may be convenient, but this practice may lead to sub-quality results. Study of edge detection [31] has a similar issue. A non-local maxima suppression approach is used to remove edge redundancy. Each edge candidate has a fitting value, indicating how likely the candidate may be an edge. A fitting value of the candidate is compared to values of all its neighbors. The candidate is suppressed if there is at least one value of its neighbors larger than the one of the candidate. To employ such redundancy removal approach, it is required an extra information in addition to window location. Such extra information is a value quantifying a degree of detection confidence, e.g., posterior probability $P(y|X)$ or, in case of SVM, a decision score [29] (Eq. 1). Our redundancy removal is performed by (1) sorting all detected bounding boxes by their fitting values in descending order, (2) choosing the top bounding box on the sorted list and putting it in another list, called a reporting list, (3) then choosing the next bounding box on the sorted list to be a candidate bounding box, (4) comparing the candidate to every bounding box on the reporting list, if the candidate is redundant to any box on the reporting list, it is suppressed. Otherwise, it is put into the reporting list. Then, (5) repeat the process (steps 3 and 4) until the sorted list is exhausted.

Redundancy is checked by assuming that if an area of a candidate bounding box overlaps an area of a higher fitting-value bounding box (which is a bounding box on the reporting list) more than a specific threshold, then it is redundant. In practice, we found that using an overlapping ratio is more favourable. It is more intuitive and also insensitive to window size. A ratio of overlapping is defined as a proportion of an overlapping area between two bounding boxes to a larger area of the two. That is overlapping ratio, $R = (A_c \cap A_r)/max\{A_c, A_r\}$, where A_c and A_r are areas of the candidate and reporting bounding boxes, respectively. Rationale for using a larger area to be a denominator is drawn from a case of comparing bounding boxes of different sizes and, especially, when the sizes are too different. Firstly, this scheme gives a consistent result whether a candidate is a small bounding box compared to a large reporting bounding box or vice versa. Secondly, when the two sizes are too different, they are likely to indicate two different customers locating at different depth of view. Therefore, using a larger denominator allows a candidate a better chance to be retained.

Hot Zone Visualization. Hot zone map is a color-based presentation of spatial visiting frequencies. Visiting frequencies are inferred from detected locations on image frames corresponding to time duration of interest. Our study constructs a hot zone map based on Kernel Density Estimation (KDE) [32]. Given top-left and bottom-right coordinates (x_t,y_t)'s and (x_b,y_b)'s of the detected locations, representative points $(c^{(x)},c^{(y)})$'s are computed as centroid coordinates: $c^{(x)} = (x_t + x_b)/2$ and $c^{(y)} = (y_t + y_b)/2$. Given every detected centroid $c_i = (c_i^{(x)},c_i^{(y)})$ for $i = 1, ..., N_d$ and N_d is a number of detected coordinates, KDE estimates a probability density at location v by $(1/N_d) \cdot (2\pi\sigma^2)^{-1/2} \cdot z(v)$, where

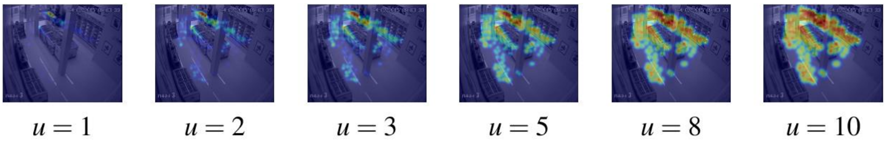
$$z(v) = \sum_{i=1}^{N_d} e^{-\frac{\|v-c_i\|^2}{2\sigma^2}} , \tag{2}$$

variable σ is a user-specified parameter to control smoothness of the function. Producing a hot zone map does not require a proper probability treatment, only *z(v)*, denoted "heat", is sufficient. To produce a hot zone map, heat values at all locations on the map are computed by Eq. 2, then they are mapped to appropriate colors based on a desired color scheme. In practice, it is more convenient for marketing personnel to be able to adjust a color scheme so that some ranges of visiting frequencies become more striking at desired degrees. Instead of directly changing a color scheme, this can be achieved easily by introducing another parameter to globally manipulate heat values. Then, the manipulated heat values can be mapped on a same color scheme, but the resulting hot zone map appears as if it is produced on a different mapping color scale. A manipulated heat value is called "heat intensity." One simple manipulation is to power a normalized heat value to a fraction of the manipulation parameter. That is, given heat value *z* and parameter *u*, heat intensity is calculated by,

$$z' = \left(\frac{z - z_{\min}}{z_{\max} - z_{\min}} \right)^{\frac{1}{u}} . \qquad (3)$$

Fig. 2 shows examples of hot zone maps produced from the same heat values, but different values of parameter *u*. It should be noted that using *u > 1* leads to a heat intensification effect, which allows lower visiting frequencies to be more noticeable. The left most picture shows a hot zone map without intensification (*u = 1*). Without intensification, only the most frequently visited area, which is around cashier counter, is noticeable. This is trivial and provides virtually no marketing insight. With different degrees of intensification, the second most and other less frequently visited areas can be identified and examined, as shown in other pictures (*u* = 2, 3, 5, 8, 10).

This is a figure example:



$u = 1$ $u = 2$ $u = 3$ $u = 5$ $u = 8$ $u = 10$

Figure 2
Hot zone maps at various intensities

4 Integration of Multiple Cues

A classical object detection approach relies only on evaluating visual features containing in a window. Window is a very limited focal image area, compared to an entire image frame. Complementarily, contextual information can provide important cues for visual perception, especially when image quality is poor [2]. Contextual information can be in many forms. Torralba [1] used a visual features of an entire scene as contextual information.

The notion of contextual information is based on a single frame detection. This study uses a term “additional cue” for a broader notion that also accommodates any useful addition for either a single frame or a video setting. For example, for video analytics, we may be able to use a state-based cue, such as frame continuity that a location of a customer in a subsequent frame is likely to appear near the one of its previous frame. The notion also can accommodate prior knowledge, such as an awareness of an entrance or an exit, where a customer can appear or disappear regarding frame continuity. This example of prior knowledge can provide relaxation on the state-based cue. Although our notion of additional cues initially developed based on additional information, it can be extended to integrate results from many decisive models. For example, a main cue may be drawn from a generic classifier, while an additional cue could be from a task-specific classifier. This allows a local task-specific adjustment to a global generic inference system, which may be available off-the-shelf.

A simple integration scheme is derived based on a probabilistic approach. It follows an approach of Torralba [1]’s contextual priming with two major distinctions. Firstly, it is generality of a notion of an additional cue. Secondly, we relax probabilistic relation to explicitly distinguish two main components, a generic object detection model and a local characteristic model. This explicit discernment allows utilization of a well-trained generic object detection model with an enhancement tailored for a specific task. This approach aligns with an enticing concept of transfer and hierarchical learning. A classical object detection counts on a classifier to evaluate a set of visual features inside a window and determine if the window contains an object. A probabilistic approach is either to directly determine the likelihood that an object of interest is present or determine it through a generative model $P(O|M) = P(M|O)\ P(O)/P(M)$, where $P(O|M)$ is a conditional probability density function (PDF) of presence of the object O given a set of main features M. Since P(M) does not have any effect on an object inference and it is difficult to determine, it is omitted and the relation is left to $P(O|M) \propto P(M|O)\cdot P(O)$.

Given that additional cue A is available, the presence of the object can be determined with the likelihood $P(O|M,A)$. Applying Bayes’s rule, we get $P(O|M,A) = P(O, M, A)/P(M,A) = P(M,A|O)\ P(O)/P(M,A)$. Denominator $P(M,A)$ does not have any effect on the final decision, the likelihood can be written as, $P(O|M,A) \propto P(M,A|O)\cdot P(O)$. Given training data, $P(M,A|O)$ can be estimated. Choices of estimating models are plentiful, e.g., Gaussian Mixture Model (GMM) and Expectation-Maximization method (EM), Self Organizing Map (SOM) and Artificial Neural Network (ANN), and, for a small set of data, Kernel Density Estimation (KDE).

With Bayes' product rule, we also can write $P(O|M,A) \propto P(A|M,O) \cdot P(M|O) \cdot P(O)$. This expression distinguishes a generic model $P(M|O)$ and a local characteristics $P(A|M,O)$. A generic model is a function of only primary information M. This modularization allows a use of an available good generic model with its adjustment to local characteristics. This may be interpreted as a local adjustment to exploit specific aspects in order to fine tune the combination to better fit a particular task. However, PDF $P(A|M,O)$ is difficult to estimate. One possible remedy is to relax $P(A|M,O)$ with an assumption that primary and additional cues are independent. That is, $P(A|M,O) \approx P(A|O)$, which leads to $P(O|M,A) \propto P(A|O) \cdot P(M|O) \cdot P(O)$. Both $P(A|O)$ and $P(M|O)$ can be estimated efficiently based on training data. Both $P(A|O)$ and $P(M|O)$ require generative models. An equivalent form for a main discriminative model is $P(O|M,A) \propto P(A|O) \cdot P(O|M)$. Term $P(M)$ is also omitted here for the reasons discussed earlier. Similarly, when additional discriminative model is easier to acquired, the expression can be manipulated to $P(O|M,A) \propto P(O|A) \cdot P(O|M)/P(O)$. The term $P(O)$ can be simply estimated by N_o/N, where N_o is a number of windows containing the object and N is a number of all windows in a training set. Given such relation, define a decision score

$$s_d = f_a(\hat{X}) \cdot f_m(X), \tag{4}$$

where $f_a(\cdot)$ and $f_m(\cdot)$ are score functions related to $P(O = +1|A = \hat{X})$ and $P(O = +1|M = X)$, respectively. Vectors X and $\hat{X}$ represent main and additional cues, respectively. The integrated model predicts a positive window class when $s_d > \tau$, otherwise it predicts a negative class. Parameterτ is a user specified threshold.

5 Experiments

Our system was built as the pipeline discussed in §3. The integration was meant for detection decision (Window Classification stage, Fig. 1). Our experiments were designed to demonstrate potential of the integration framework. Four treatments were examined. Three of them were represented by detectors of the same type, but trained on different datasets. The three datasets are a generic dataset, a task-specific dataset, and a combining dataset. These generic and task-specific notions were to simultaneously examined as another goal. That was to figure out how a local task specific cue could be used to enhance a generically well-tuned detector, so that the resulting model could perform better on a specific task without having to rebuild everything from scratch. A generic dataset, denoted "Gdataset," acquired data from Inria person dataset [22]. A task-specific dataset, denoted "Tdataset," acquired data from a retailer video dataset (details discussed later). The third set, denoted "GT," was a combination of both G and T datasets. The treatments or models trained on G, T, and GT datasets were referred to as G, T, and GT, respectively. The last treatment, denoted "G+AT", represented a detector built on the integration framework with G model as its main cue and T model as its additional cue. Our experiment used $1/\{1+exp(-s')\}$ for score functions, $f_a(\cdot)$ and $f_m(\cdot)$ in Eq. 4, where s' was s/s_{max} when s was SVM decision score (Eq. 1) and s_{max} was the maximum decision score.

G dataset was comprised of 2,416 positive, 4,872 negative, and 1,000 hard negative examples[1]. T dataset had 80 positive, 315 negative, 141 hard positive, and 582 hard negative examples. Video data recording activities in a retailer store, donated by our funder, was used for both training (as T dataset) and evaluation. A total of 20 video clips, each lasted about 30 sec. to 3 min., were separated into 15 and 5 clips for training and evaluating sets, respectively. Image frames were captured from video clips at a rate 1:30, which made it 1 frame/sec. All frames were 704x576-pixel RBG-color images. A region of interest (ROI) is defined to be an area of 250x150-pixel around a store entry. Each ROI was processed in 3 scales, 0.86x, 1x (original scale), and 1.2x. Windows were sampled by sliding window scheme at a window size of 64x128 pixels and step sizes of 4 in both x- and y-directions. Each window was passed through window classification process, which was central to our investigation, before gone through redundancy removal at an overlapping ratio over a threshold of 0.5.

All models, G, T, and GT, were HOG-based radial-basis SVM classifiers, but trained with three different datasets, as mentioned earlier. HOG features were computed with 9 orientations, cells of 8x8 pixels, and blocks of 2x2 cells. The SVM model was set with parameters C=10.0 and radial basis $\gamma = 0.1$. Detection performances, miss rate (MR) and false positive per window (FPW), of all treatments were evaluated against ground truth of the evaluation set of the retailer video data. The evaluation set contained 609 image frames.

It was worth emphasizing that treatment G was our implementation intended to replicate a classic Dalal-Triggs human detection [22]. Treatment G used Dalal-Triggs method and was trained with the same Inria person dataset. There were only two major differences. Firstly, SVM was trained on a smaller number of examples in order to mitigate a memory issue. Secondly, our implementation of HOG did not have a downweighing mechanism for pixels near the edges of the block, which Dalal and Triggs reported to contribute to only about 1% improvement. However, it should be emphasized again that our study was not proposing a competing method against a classic Dalal-Triggs human detection

[1] Hard negative examples are negative examples that were incorrectly classified by a simple classifier. We identified such examples in our preliminary study by applying a classifier trained with regular positive and negative examples on a set of negative examples. Then, negative examples that were incorrectly classified were hand-picked to be the hard-negative examples. Due to our memory limitation, we had to hand-picked negative examples that look distinct, so that they would be beneficial in a training process, while did not exhaust our computer memory.

[22]. Our framework was proposed as an approach to extend any object detection method, not limited to only Dalal-Triggs method [22]. Classical Dalal-Triggs schemes, mainly employing HOG features and SVM, were extensively used in all of our four treatments. They were to represent a generic detector, a task-specific detector, a conventional combined detector, and a combined detector based on our proposed framework.

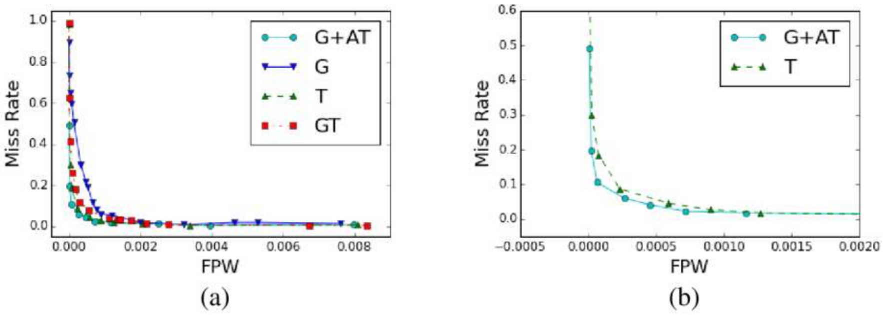


Figure 3
Detection performances in MR-FPW plots: (a) all treatments and (b) treatments T and G+AT in a closer view.

Fig. 3a showed MR-FPW plots of 4 treatments. Treatment G+AT apparently outperformed treatments G and GT, but comparison between treatment G+AT and T was better seen in Fig. 3b, where treatment G+AT was shown to perform slightly better than treatment T. At FPW of about 0.0001, treatments G, T, GT, and G+AT delivered MRs at 0.507, 0.184, 0.263, and 0.106, respectively. That is, the integration framework showed 42% improvement over treatment T, the best performing treatment without the framework.

Discussion and Conclusions

The result shows promising potential of the framework. An integration of a task specific model to a generic model clearly improves over a generic model. This improvement is emphasized, since the integration also outperforms the model with a combining datasets. Therefore, this approach shows a benefit over simply combining datasets. Excluding the integration, model T outperforms the other two models. For model G, the explanation is obvious, but for model GT, which also has T dataset in its training, the explanation lies in the proportion of the training data. Generally, generic data is easier to be acquired than task-specific data is. That reflects in sizes of training data. Here, sizes of G dataset to T dataset is 7.4:1. This large difference may weigh down inference from T dataset excessively. Building separate models and integrating them later with the integration framework allow some distinct characteristic inference of minority to prevail, while still retain principal values of the global majority. Those retaining global gumptions are those that are indispensable, which in turn deliver as an improvement seen over other models, including a task specific model. Our findings are only preliminary and it requires a more thoroughly investigation to realize implication of this framework, as a key to tweak a global inference system with a local sense, as an ensemble of various models, as a fusion of different sources of information, or as a mélange of models and cues. Regarding worth investigating cues, specific characteristics of customers, e.g., constantly moving nature and common trajectory, frame continuity, and possible locations on a scene seem to be able to provide promising cues. Customer trajectory and locations may also provide a key to distinguish a high-level deduction, such as recognition of the difference between staff and customers. Frequently visited locations, conventionally an end result, itself can be fed back in the pipeline and used to deduce likeliness of presence of customers to improve detection quality, which in turn results in more accurate frequently visited locations. Customer trajectory is interesting as a propitious cue and as insightful visualization for understanding customer behavior. For customer behavior analysis via video analytics, issues of distortion and an application of object tracking appear worth prioritizing.

To summarize, this article provides a detailed discussion on an entire procedure for customer analysis via video analytics, as well as demonstrates potential of the integration framework for customer detection.

Acknowledgement

The authors appreciate Betagro Co.Ltd. for valuable data and funding our study and E-SAAN Center for Business and Economic Research (ECBER), Khon Kaen University for initiating and facilitating the cooperation.

References

[1] Torralba, A. “Contextual priming for object detection”, International Journal of Computer Vision 53(2), 169–191 (2003).

[2] Mottaghi, R., Chen, X., Liu, X., Cho, N.-G., Lee, S.-W., Fidler, S., Urtasun, R., and Yuille, A., “The role of context for object detection and semantic segmentation in the wild,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., 2014.

[3] Popa, M.C., Rothkrantz, L.J.M., Yang, Z., Wiggers, P., Braspenning, R. and Shan, C., “Analysis of shopping behavior based on Surveillance System,” in Proc. IEEE Conf. Systems, Man, and Cybernetics (SMC), 2010.

[4] Popa, M.C., Rothkrantz, L.J.M., Shan, C., Gritti, T., and Wiggers, P., “Semantic assessment of shopping behavior using trajectories, shopping related actions, and context Information,” Pattern Recognition Letters 34(7), pp. 809–819 (2013).

[5] Connell, J., Fan, Q., Gabbur, P., Haas, N., Pankanti, S., and Trinh, H., “Retail video analytics: an overview and survey,” Proc. SPIE 8663, 2013.

[6] Jing, G., Siong, C.E., and Rajan, D., “Foreground motion detection by difference-based spatial temporal entropy image,” IEEE Region 10 Conference (TENCON), vol. A, pp. 379–392, 2004.

[7] Tang, Z., and Miao, Z., “Fast background subtraction and shadow elimination using improved gaussian mixture model,” IEEE International Workshop on Haptic Audio Visual Environments and Their Applications, pp. 38–41, 2007.

[8] Benenson, R., Omran, M., Hosang, J., and Schiele, B., “Ten years of pedestrian detection, what have we learned?,” European Conference on Computer Vision (ECCV), 2014.

[9] Dollar, P., Wojek, C., Schiele, B., and Perona, P., “Pedestrian detection: an evaluation of the state of the art,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 99, 2011.

[10] Hosang, J., Benenson, R., and Schiele, B., “How good are detection proposals, really?,” British Machine Vision Conference (BMVC), 2014.

[11] Ullman, S., Assif, L., Fetaya, E., and Harari, D., “Atoms of recognition in human and computer vision,” Proc. Natl. Acad. Sci. USA 2016.

[12] Yang, B., Yan, J., Lei, Z., and Li, S. Z., “CRAFT objects from images,” CVPR 2016.

[13] Morre, O., Veillard, A., Lin, J., Petta, J., Chandrasekhar, V., and Poggio, T., “Group invariant deep representations for image instance retrieval,” Journal of Brains, Minds, and Machines 43, 2016.

[14] Yilmaz, A., Javed, O., and Shah, M., “Object tracking: a survey,” ACM Computing Surveys 38(4), 2006.

[15] Comaniciu, D., Ramesh, V., and Meer, P., “Real-time tracking of non-rigid objects using mean shift,” IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 142–149, 2000.

[16] Chan-Hon-Tong, A., Achard, C., and Lucat, L., “Simultaneous segmentation and classification of human actions in video streams using deeply optimized Hough transform,” Pattern Recognition 47(12), pp. 3807–3818, 2014.

[17] Pereira, E.M., Ciobanu, L., and Cardoso, J.S., “Context-based trajectory descriptor for human activity profiling,” IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 2385–2390, 2014.

[18] Yu, J., Jeon, M., and Pedrycz, W., “Weighted feature trajectories and concatenated bag-of-features for action recognition,” Neurocomputing 131, pp. 200–207, 2014.

[19] Ko, T. “A survey on behavior analysis in video surveillance applications,” Video Surveillance, Prof. Weiyao Lin (Ed.), InTech (2011), DOI: 10.5772/15302.

[20] Szegedy, C. , Liu, W., Jia, Y. , Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A., “Going deeper with convolutions,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015.

[21] Dollar, P., Belongie, S., and Perona, P., “The fast pedestrian detector in the west,” BMVC 2010.

[22] Dalal, N.andTriggs,B.,“Histograms of oriented gradients for human detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., 2005.

[23] Walk, S., Majer, N., Schindler, K., and Schiele, B., “New features and insights for pedestrian detection,” IEEE Conf. Computer Vision and Pattern Recognition, 2010.

[24] Viola, P. and Jones, M., “Rapid object detection using a boosted cascade of simple features,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., 2001.

[25] Fei-Fei, L., and Perona, P., “A bayesian hierarchical model for learning natural scene categories,” Computer Vision and Pattern Recognition, 2005.

[26] Felzenszwalb, R., McAllester, D., and Ramanan, D., “A discriminatively trained, multiscale, deformable part model,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008.

[27] Malisiewicz, T., Gupta, A., and Efros, A. A., “Ensemble of exemplar-SVMs for object detection and beyond,” ICCV 2011.

[28] Cortes, C. and Vapnik, V., “Support-vector networks,”Machine Learning, 20, pp. 273–297 (1995).

[29] Chang, C.-C. and Lin, C.-J., “LIBSVM : a library for support vector machines,” ACM Transactions on Intelligent Systems and Technology, 2(27), pp. 1–27, (2011).

[30] Milgram, J., Cheriet, M., and Sabourin, R., “One against one or one against all: which one is better for handwriting recognition with SVMs?,”10th International Workshop on Frontiers in Handwriting Recognition, 2006, La Baule (France), Suvisoft.

[31] Canny, J., “A computational approach To edge detection,” IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6), pp. 679-698, 1986.

[32] Bishop, C., Pattern Recognition and Machine Learning. Springer, 2006.

[33] van der Maaten, L.J.P. and Hinton, G.E., “Visualizing high-dimensional data using t-SNE, ” Journal of Machine Learning Research, 9, pp. 2579-2605, 2008.

Appendix: Multivariate Analysis

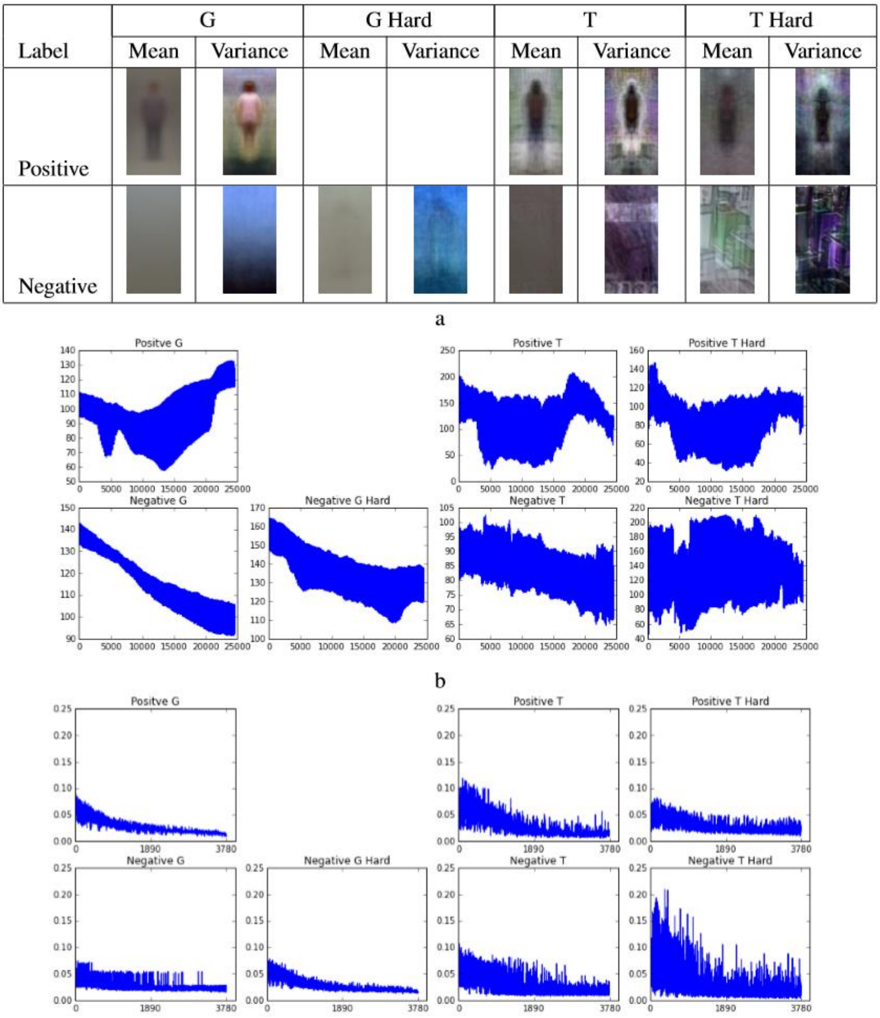
Means and variances of all datasets are shown in Figure A.1. In Figure A.1 (a), a mean of positive examples resembles a rough outline of a standing human. A mean of negative examples from G and T datasets looks like a simple gray patch. This shows a well balance of variety of negative examples that are averaged to medium values throughout every pixel. Although a mean of hard positive examples from T dataset still resembles a rough outline of a human, it is less noticeable than those of regular positive examples. Hard positive examples from G datasets are excluded from our experiments or analysis for an economical reason. A mean of hard negative examples from T dataset vaguely resembles an area in the store where the classifier is often confused. A variance of hard negative examples from G datasets reveals a barely noticeable trace similar to an outline of human's head and shoulder, which may be a reason that makes those examples difficult to classify. There is no sign of this trace in a variance of regular negative examples. A variance of hard negative examples from T datasets also reveals a lighter spot in the middle of the area. This high variation coincidentally locates around the middle of the area where critical classification is supposedly to take place. Therefore, it contributes to confusion and consequently makes those hard negative examples difficult to classify.

While human perceives each image patch effortlessly, a classifier takes each 64×128 pixel color image as a vector of 24576 values. Figure A.1 (b) shows means of all datasets as series of pixel-intensity values. Processing directly on this highly dimensional information requires a considerable amount of computing resources. A general approach is to convert the high-dimension data to more manageable lower dimension form. Our study employs Histogram of Oriented Gradient (HOG) [22] to map from 24576-dimension data to 3780-dimension HOG features. Figure A.1 (c) shows means of HOG features of all datasets. It should be noted that a mean of regular positive examples looks distinguishable from a mean of regular negative examples regardless of whether it is G or T dataset. However, patterns of hard examples are less distinguishable: a mean of hard negative examples from G dataset looks similar to the one of positive examples.

Examination of correlation, clustering, and visualisation of data with high dimensionality is less straightforward, but it can be mitigated using dimension reduction projection, e.g., t-SNE [33]. Figure A.2 shows scatter plots based on t-SNE projection from high dimensional datapoints onto a two-dimension space. Since a mechanism of t-SNE is to project high dimensional datapoints onto a lower dimensional space such that a projected relative distance between similar datapoints is preserved, while a projected relative distance between dissimilar datapoints is allowed to have a higher degree of relaxation. Specifically, given datapoints in high dimensional space $\{x_1,x_2,\cdots,x_N\}$ and $x_i \in \boldsymbol{R}^D$, t-SNE is to find corresponding datapoints in low dimensional space $\{y_1,y_2,\cdots,y_N\}$, $y_i \in \boldsymbol{R}^d$, and $d << D$. That is, $\{y_1^*,y_2^*,\cdots,y_N^*\}=argmin_{y_1,y_2,\cdots,y_N} \Sigma_i \Sigma_{j\neq i} p_{ij} \cdot log(p_{ij}/q_{ij})$. A degree of relative similarity between high-dimension datapoints i and j is defined as $p_{ij} = (p_{j|i} +$

$p_{i|j})/(2N)$, where $p_{b|a} = (exp\{-||x_a-x_b||^2/2\sigma_i^2\})/(\Sigma_{c\neq a}\ exp\{-||x_a-x_c||^2/2\sigma_i^2\})$ and σ_i's are user specific parameters, called "perplexity." A degree of relative similarity between low-dimension datapoints i and j is defined as $q_{ij} = (1+||y_i-y_j||^2)^{-1}/\ \Sigma_k\ \Sigma_{l\neq k}\ (1+||y_k-y_l||^2)^{-1}$. Problem formulation of t-SNE directly enforces that projections of similar datapoints must be projected onto close locations. It does not enforce the projections of dissimilar datapoints in the same degree. Distance between two dissimilar datapoints is indirectly enforced through mechanism of relativity. That is $\Sigma_i\Sigma_{j\neq i}\ q_{ij} = 1$, therefore when a projected distance between two dissimilar datapoints is too small, the corresponding projected relative similarity q_{ij} of those two dissimilar datapoints will be too large in the expense of that other projected relative similarities including the ones corresponding to similar datapoints will be too small. Consequently, that reflects to the objective function through too large values of terms corresponding to similar datapoints and the process of minimization will regulate to discourage projecting two dissimilar datapoints onto nearby locations.

Regarding t-SNE projections of original and HOG features (Figure A.2), HOG features do not seem to help much in term of data separation. The scale of Figure A.2 (a) may make it appear less separable than Figure A.2 (b), however, after a close investigation the t-SNE projection of original datapoints do not appear to be less separable than the t-SNE projection of HOG-mapped datapoints. At this point, an obvious advantage of using HOG features seems to be putting a number of dimensions down to a manageable size.



a

b

c

Figure A.1

Multivariate analysis of the datasets: (a) mean and variance of pixel intensities in each training dataset, presented as color images (each has 64x128x3 pixel intensities); (b) mean of pixel intensities in each training dataset, presented as a series of pixel intensities (each series has 24576 values); and (c) mean of HOG features in each training dataset. HOG scheme reduces dimensionality from 24576 of original pixel intensities to 3780 of HOG features. Acronyms “G”, “G Hard”, “T”, and “T Hard” indicate association to G dataset, hard examples from G dataset, T dataset, and hard examples from T dataset, respectively. Words “Positive” and “Negative” indicate association to positive examples and negative examples, respectively.

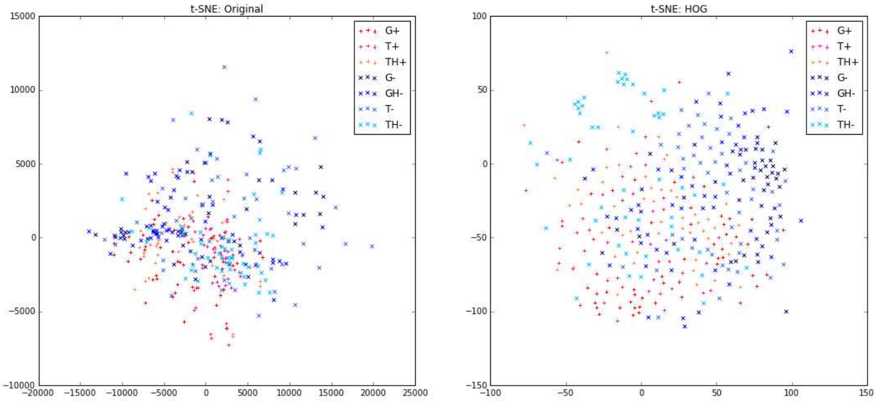


Figure A.2

Scatter plots of t-SNE projections of (a) original datapoints and (b) HOG-mapped datapoints. Perplexity σ_i = 10 for all i's. In order to mitigate memory issue, 50 representative points of each category are used instead of real datapoints. Representative points are centroids of clusters based on K-Means clustering. Symbols 'G+', 'T+', 'TH+', 'G-', 'GH-', 'T-', and 'TH-' indicate representative points for positive examples of G dataset, positive examples of T dataset, hard positive examples of T dataset, negative examples of G dataset, hard negative examples of G dataset, negative examples of T dataset, and hard negative examples of T dataset, respectively. (The image is best viewed in colors.)

Use of Machine Learning to Analyze and – Hopefully – Predict Volcano Activity

Justin Parra[1], **Olac Fuentes**[1], **Elizabeth Anthony**[2], **and Vladik Kreinovich**[1]

Departments of [1]Computer Science and [2]Geological Sciences
University of Texas at El Paso, 500 W. University, El Paso, TX 79968, USA,
jrparra2@miners.utep.edu, ofuentes@utep.edu, eanthony@utep.edu,
vladik@utep.edu

Abstract: Volcanic eruptions cause significant loss of lives and property around the world each year. Their importance is highlighted by the sheer number of volcanoes for which eruptive activity is probable. These volcanoes are classified as in a state of unrest. The Global Volcano Project maintained by the Smithsonian Institution estimates that approximately 600 volcanoes, many proximal to major urban areas, are currently in this state of unrest. A spectrum of phenomena serve as precursors to eruption, including ground deformation, emission of gases, and seismic activity. The precursors are caused by magma upwelling from the Moho to the shallow (2-5 km) subsurface and magma movement in the volcano conduit immediately preceding eruption.

Precursors have in common the fundamental petrologic processes of melt generation in the lithosphere and subsequent magma differentiation. Our ultimate objective is to apply state-of-the-art machine learning techniques to volcano eruption forecasting. In this paper, we applied machine learning techniques to the precursor data, such as the 1999 eruption of Redoubt volcano, Alaska, for which a comprehensive record of precursor activity exists as USGS public domain files and global data bases, such as the Smithsonian Institution Global Volcanology Project and Aerocom (which is part of the HEMCO data base). As a result, we get geophysically meaningful results.

Keywords: machine learning; volcano activities; clustering

1 Volcano Eruption Forecasting: Formulation of the Problem and State-of-the-Art

1.1 Need for volcano eruption forecasting

Because of the possible catastrophic consequences, researchers have always been trying to develop methods for predicting volcano eruptions.

We believe that volcano eruption forecasting is possible. The hope for predicting volcano eruptions comes from the fact that most eruptions are preceded by different types of unusual activities.

In geophysical terms, volcanoes that will erupt in the near future are classified by the community of volcanologists as in a state of “unrest”. Unrest is manifested as a combination of changes in the amount and chemical composition of volcanic gas emissions [17], ground deformation above the volcanic edifice [3], and seismic activity [13]. The activity is the result of subsurface movement of magma as it ascends to the surface.

1.2 Volcano eruption forecasting is difficult

Unfortunately, in spite of the seemingly clear relation between these precursors and the following eruptions, there is still no good way to make long-term predictions of volcanic activity: no matter what combination of precursors we select:

- sometimes, a similar combination results in an eruption, while
- in other cases, a seemingly similar activity is not followed by an eruption.

1.3 Need for probabilistic forecasting

In general, the relation between the precursors and the eruptions has a probabilistic character: the presence of precursors does not necessarily indicate that the eruption is imminent, but it seems to increase the probability of the eruption.

From this viewpoint, we can only predict probabilities of eruptions of different strength and type.

1.4 Probabilistic methods of volcano eruption forecasting: state-of-the-art

Several research papers use probabilistic methods to predict the eruption probabilities; see, e.g., [2, 12, 15].

These methods start with the known power-law models that describe the relation between the different characteristics – e.g., between the eruption strength and the time to the next eruption – and add appropriate probabilistic models to describe the inaccuracy of these relations. The parameters of the corresponding multi-parameter models are then tuned to match the observed phenomena. The resulting tuned model is then used for forecasting.

This statistical approach works perfectly well in many applications to engineering and science. For volcanic eruptions, this approach has led to several reasonable short-term and long-term probabilistic forecasts.

However, these predictions are still far from perfect. It is therefore desirable to improve the accuracy and reliability of the existing predictions.

2 Analysis of the Problem and the Resulting Ideas

2.1 Why predicting volcano eruptions is different from other types of predictions

In our opinion, two specific features of volcanic eruptions limit the potential of such purely statistical approach.

2.2 First specific feature of predicting volcano eruptions

2.2.1 Description of the feature

The first specific feature of volcano eruptions is related to the fact that successful statistical methods require that we know the parameters of the corresponding probabilistic models.

To accurately determine the values of these parameters in a statistical setting, we need to have reasonably large data samples. This is a big problem for volcanic studies, since, in contrast to many engineering and scientific phenomena, volcano eruptions are relatively rare events.

2.2.2 What has been done to overcome this difficulty: Bayesian approach

One approach to compensate for the smallness of samples is to add expert knowledge, which can be described in terms of subjective prior probabilities of different events.

These approximate prior values of the corresponding probabilities are then updated based on the observations; the formulas for such an update were first discovered by Bayes; because of this fact, such an approach is known as *Bayesian*; see, e.g., [4]. This approach has been successfully used to predict volcanic activity; see, e.g., [15].

2.2.3 Limitations of the Bayesian approach

The problem with applying Bayesian approach to volcanic eruptions is that different experts may have different opinions, so we end up with different prior probabilities – and thus, different predictions.

When we have a reasonably large data sample, the observation-based update tilts the original subjective probabilities towards the observed frequencies. As a result, the dependence on the initial (prior) probabilities drastically decreases.

However, for situations like volcanic eruptions, when the sample sizes are small, the resulting predictions remain strongly dependent on the original subjective probabilities.

2.3 Second specific feature of predicting volcano eruptions

2.3.1 Description of the feature

The second specific feature of predicting volcano eruptions is as follows.

In many engineering and scientific phenomena, we know reasonably accurate formulas describing the dependence between different quantities – e.g., differential equations describing elasticity, Navier-Stokes equations describing liquids, etc.

In contrast, for volcanic activities, we do not know the exact shape of the corresponding dependencies.

When we use the traditional finite-parametric probabilistic models, e.g., power law models (which are known to be a rather crude approximation to real-life phenomena), we are thus limiting ourselves to these crude models, and hence, restricting our ability to forecast.

2.3.2 How to overcome the corresponding difficulty: need for machine learning techniques

To overcome this problem, it is therefore desirable to use non-parametric prediction models.

Such methods, when we do not fix the shape of the dependence from the very beginning, but let the data determine this shape, are known as *machine learning* techniques; see, e.g., [4].

When we apply such techniques, then, instead of a researcher trying to guess the corresponding relation – such as a power law – the computer-based system determines this relation by itself, based only on the observations.

Machine learning algorithms start the observed data: both

- the values of the quantities that we want to predict and
- the values of the possible related quantities that we would like to use in this prediction.

Based on this data, machine learning algorithms eventually come up with a computer model that makes accurate predictions in all given situations – and, in many applications, makes successful predictions in new situations as well.

Machine learning techniques are currently ubiquitous in many applications, they underlie the ability of modern cellphones to recognize voices, they provide security against hackers and spam, they are behind the recent successes of Artificial Intelligence such as computers winning over Go masters, and many other applications; see, e.g., [4, 6, 14].

Our eventual goal is to apply machine learning techniques to the volcanic data to come up with effective forecasting techniques.

2.4 Why we believe that machine learning methods will be helpful in volcano eruption forecasting

Our belief in machine learning techniques comes not only from their successes in modern appliances, but also from our experience of successful using these techniques in different applications.

In our previous research efforts, we have used these techniques to predict the best strategy for a robot [9, 10], to determine the parameters of stellar atmospheres based on astronomic observations [8], and in many other applications.

Last but not the least, it should be mentioned that machine learning techniques have been successfully used for predicting volcanic activities, often leading to better results that the traditional probabilistic methods; see, e.g., [7, 11] and references therein.

3 Our Study: Description and Results

3.1 Description of the problem

To test our belief, we did some preliminary proof-of-concept analysis.

Specifically, for two volcanoes for which there is an extensive record of small nearby earthquakes – Popo in Mexico and and Readout in Alaska – we analyzed the spatial locations of these earthquakes in comparison with the location of the volcano itself.

3.2 What data we used

In this study, we use open source data of precursor activity for the Aleutian chain of volcanoes [5].

The Aleutians are an arcuate chain of active volcanoes that reaches from Alaska to Russia. They represent the subduction (underthrusting) of Pacific lithosphere beneath North America. Because of their location, silicate ash erupted from them into the atmosphere impacts air traffic across major flight paths in the Pacific. We have begun our analysis with the seismic record for the volcanoes to maximize the data elements available.

3.3 What data processing methods we used

We started with the simplest type of learning, when instead of trying to predict the numerical value of a real-valued quantity, we try to predict a simple quantity with a very small number of possible values.

In such a prediction, we thus classify different objects or events into one of the few groups – corresponding to different values of the predicted few-valued quantity.

In other words, we cluster the events or objects into a small number of clusters, so that ideally,

- the events/objects within each cluster are similar to each other, while
- events/objects from different clusters are different.

For this pilot study, we use one of the simplest clustering algorithms – k-means.

In this algorithm, we iteratively compute the values of the cluster centers. In the beginning, these centers are selected at random. At each iteration:

- based on the previous selection of centers, we allocate each point to the cluster whose center is the closest – in the sense of the usual 3-D Euclidean distance – to this point;
- after that, we re-calculate the center location as the arithmetic average of all the points allocated to this particular cluster.

This process continues until it converges, i.e., until some iteration leaves the clusters and centers unchanged.

3.4 First result and its geophysical interpretation

We analyzed the data from the Redoubt volcano. For this volcano, we applied this clustering algorithm to the locations of all the nearby earthquakes occurring from January 1, 1995 to January 1, 2016. Specifically, we used earthquakes whose hypocenters are at depth not exceeding 20 km, and whose latitude and longitude differ from the volcano location by no more than 0.2 degrees. As events, we used the 3-D hypocenters of the selected earthquakes.

We then applied the k-means clustering algorithm to cluster the locations of these hypocenters in the 3-D space.

The information about these earthquakes was taken from the existing databases [1, 16]. Specifically, the information about the earthquake hypocenters magnitudes was taken from the databases listed in Table 1.

The number of selected earthquakes by year is presented on Fig. 1.

We use the “elbow” method (see, e.g., [4]) to select the number of clusters: we increased the number of clusters until we reach a point where adding one more

Year	URL
1994–1999	http://pubs.usgs.gov/of/2001/0189/
2000–2001	https://pubs.er.usgs.gov/publication/ofr02342
2002	http://pubs.usgs.gov/of/2003/0267/
2003	http://pubs.usgs.gov/of/2004/1234/
2004	http://pubs.usgs.gov/of/2005/1312/
2005	http://pubs.usgs.gov/of/2006/1264/
2006	http://pubs.usgs.gov/ds/326/
2007	http://pubs.usgs.gov/ds/367/
2008	http://pubs.usgs.gov/ds/467/
2011	http://pubs.usgs.gov/ds/730/
2012	http://pubs.usgs.gov/ds/789/
General	http://earthquake.usgs.gov/earthquakes/search/
General	http://www.ncedc.org/anss/catalog-search.html

Table 1
Sources of information about the earthquakes

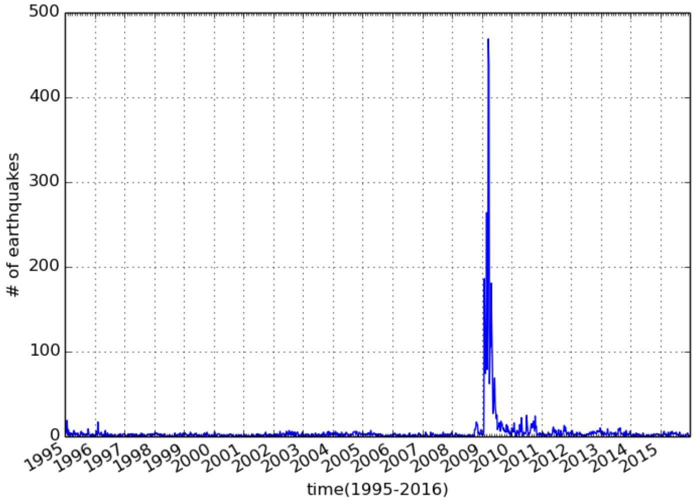


Figure 1
Number of earthquakes near the Redoubt volcano

cluster does not lead to a significant decrease in the average within-cluster variation. This resulted in $k = 3$ clusters.

The selected earthquake locations formed three clearly distinguished clusters; these clusters are described by three different colors on Fig. 2.

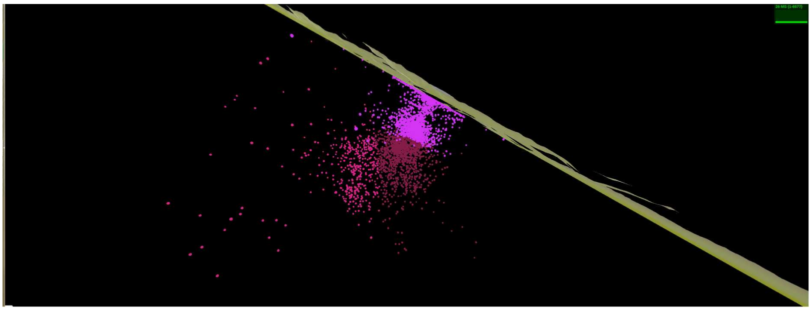


Figure 2
Clusters of earthquake locations (North is left)

Earthquakes from the first two clusters are mostly vertically located right beneath the volcano. Depth-wide, they seem to correspond to the volcano pipe and to the place where the magma goes from the magma chamber into the pipe.

The third, deeper cluster is spread mostly horizontally, it seems to correspond to a sill-shaped magma chamber.

Interestingly, the center of this third cluster is shifted in comparison to the volcano itself, so that the volcano is approximately at the edge of the cluster. In other words, it looks like the magma accumulated in the magma chamber finds the way up along the edges of the chamber – which seems to be in good accordance with the observed asymmetry of volcanic eruptions, which also usually start not at the center of the volcano, but on one of the edges of the volcano's throat (which explain the visible asymmetry of many volcanic calderas).

3.5 Second result and its geophysical interpretation

In the above clustering, we only took into account the locations of the earthquakes, but not their magnitude. In other words, very weak, barely detectable earthquakes were given the same weight as the most powerful ones. It is therefore reasonable to consider different weight for different earthquakes. In this paper, we used weights proportional to the earthquake's energy.

The strength of an earthquake is usually described by its magnitude M on the Richter's scale. Richter's scale is a logarithmic space, so the energy of an earthquake is proportional to $\left(10^{1.5}\right)^M$. This is the weight that we assigned to each earthquake.

We then used these weights to perform the weighted k-means clustering. This clustering method is similar to the usual k-means, the only difference is that when we re-calculate the location of the center, then instead of the arithmetic average

$$\frac{x_1+\ldots+x_n}{n}$$

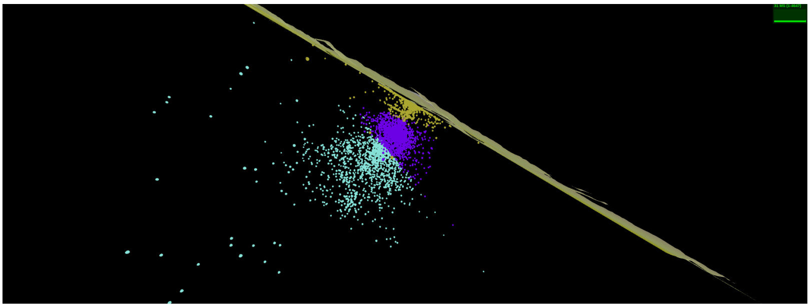


Figure 3
Clusters of earthquake locations based on weighted clustering: 3-D picture (North is left)

of the locations of all the points $x_1,\ldots,x_n$ from the cluster, we use the *weighted* average with the weights $w_i = \left(10^{1.5}\right)^{M_i}$, where M_i is the magnitude of the i-th earthquake:

$$\frac{\sum\limits_{i=1}^{n} w_i \cdot x_i}{\sum\limits_{i=1}^{n} w_i}.$$

The resulting algorithm is as follows. In the beginning, the centers are selected at random. Then, at each iteration:

- based on the previous selection of centers, we allocate each point to the cluster whose center is the closest to this point;
- after that, we re-calculate the center location as the *weighted* average of all the points allocated to this particular cluster.

This process continues until it converges, i.e., until some iteration leaves the clusters and centers unchanged.

We therefore repeated our clustering experiment, this time with weighted clustering. As a result,

- we still got three clusters at different depths, but
- this time, all three clusters were vertically aligned; see Fig. 3–7.

This also makes geophysical sense:

- while we have seismic activity throughout the whole magma chamber,
- we expect stronger activity in locations were the magma is most active, i.e., in the location where the magma is going up – which is directly beneath the pipe.

Conclusion. Of course, these are preliminary results that need to be further analyzed and confirmed.

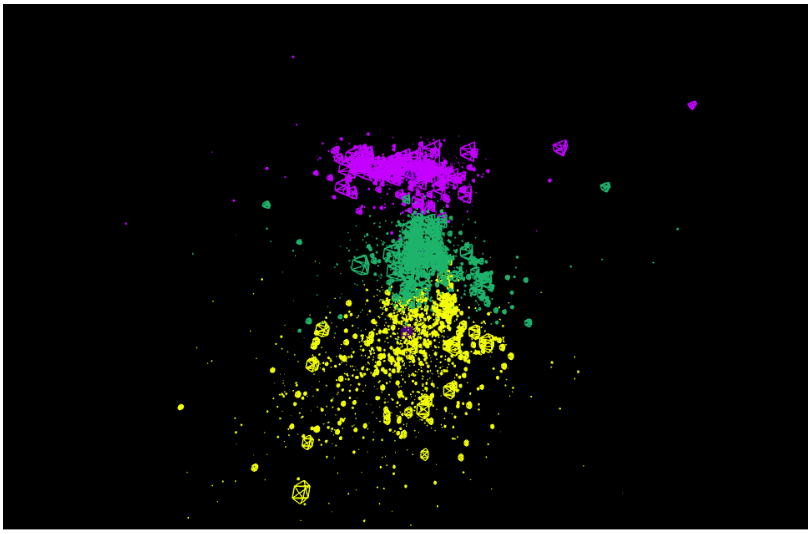


Figure 4
Clusters of earthquake locations based on weighted clustering: looking North

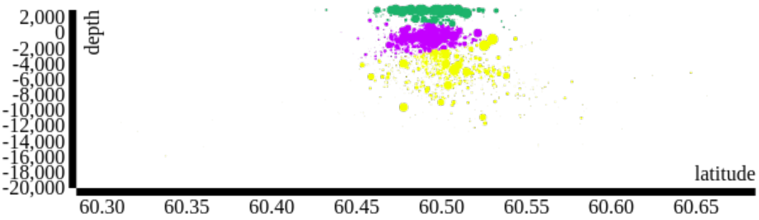


Figure 5
Clusters of earthquake locations based on weighted clustering:
latitude vs. depth

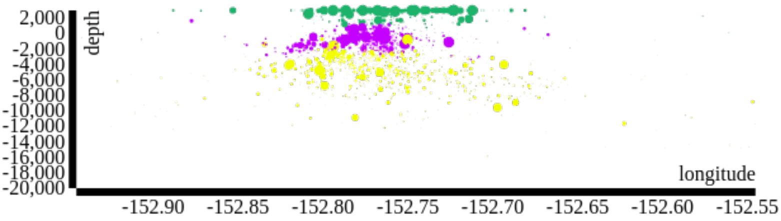


Figure 6
Clusters of earthquake locations based on weighted clustering:
longitude vs. depth

However, the very fact that, without inputting any geophysical knowledge into our computations, by simply applying general algorithms to observed data, we got geo-

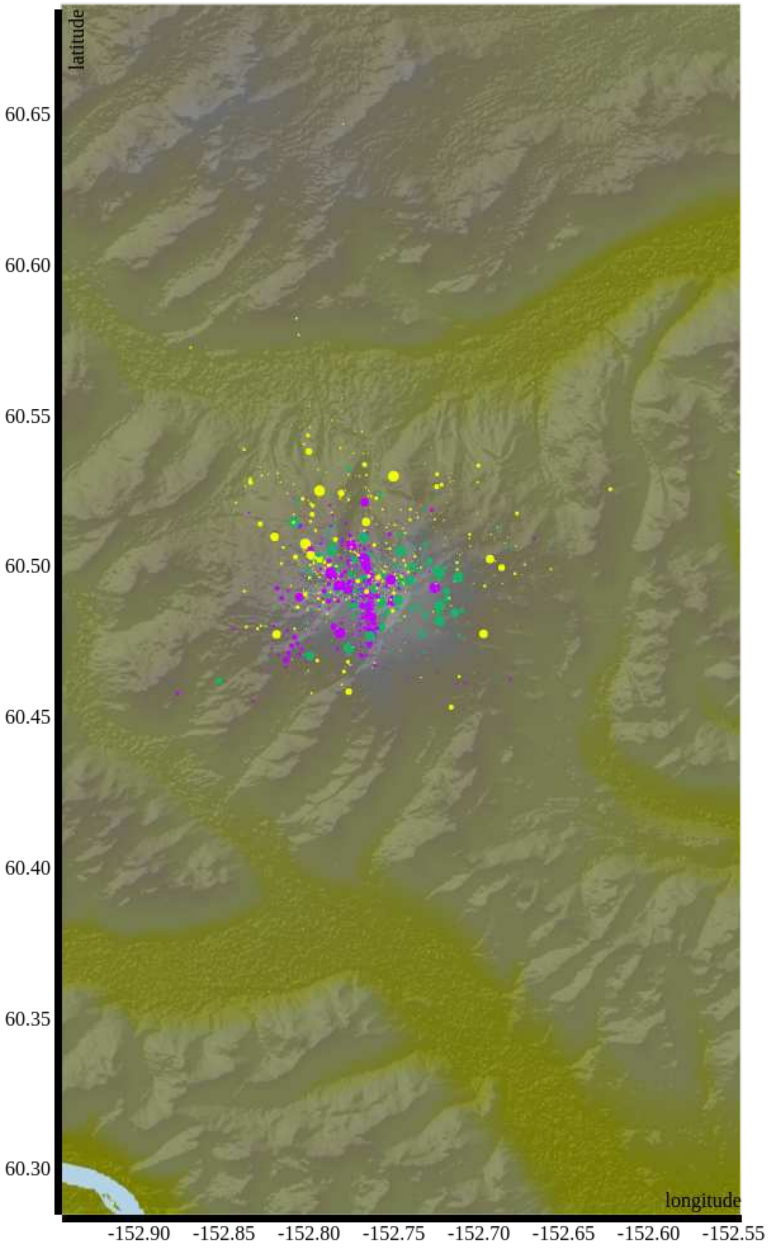


Figure 7
Clusters of earthquake locations based on weighted clustering:
latitude vs. longitude

physically meaningful results, makes us confident that by applying more sophisticated machine learning techniques to volcanic data, we will be able to capture the corresponding geophysical phenomena and thus, make reasonable forecasts.

Acknowledgements

This work was supported in part by the National Science Foundation grants HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and DUE-0926721, and by an award "UTEP and Prudential Actuarial Science Academy and Pipeline Initiative" from Prudential Foundation.

The authors are thankful to Ildar Batyrshin, Grigory Sidorov, Alexander Gelbukh, and all the organizers of MICAI'2016 for their support and encouragement, and to the anonymous referees for valuable suggestions.

References

[1] Advanced National Seismic System (ANSS) Database, hosted by Northern California Earthquake Data Center,
http://www.ncedc.org/anss/catalog-search.html

[2] M. S. Bebbington and W. Marzocchi: Stochastic models for earthquake triggering of volcanic eruptions, Journal of Geophysical Research, 2011, Vol. 116, Paper B05204.

[3] J. Biggs, E. Y. Anthony, and C. J. Ebinger: Multiple inflation and deflation events at Kenyan volcanoes, East African Rift, Geology, 2009, Vol. 37, pp. 979–982.

[4] C. M. Bishop: Pattern Recognition and Machine Learning, Springer, New York, 2006.

[5] K. F. Bull and H. Buurman: An overview of the 2009 eruption of Redoubt Volcano, Alaska, Journal of Volcanology and Geothermal Research, 2013, Vol. 259, pp. 2–15.

[6] K. Buza and J. Koller: Classification of electroencephalograph data: a hubness-aware approach, Acta Polytechnica Hungarica, 2016, Vol. 13, No. 2, pp. 27–46.

[7] M. Curilem, F. Huenupan, C. San Martin, G. Fuentealba, C. Cardona, L. Franco, G. Acuña, and M. Chacón: Feature analysis for the classification of volcanic seismic events using support vector machines. In: A. Gelbukh, F. C. Espinoza, and S.-N. Galicia-Haro (eds): Nature-Inspired Computation and Machine Learning: Proceedings of MICAI'2014, Springer Lecture Notes in Computer Science, Vol. 8857, 2014, pp. 160–171.

[8] O. Fuentes: Automatic determination of stellar atmospheric parameters using neural networks and instance-based learning, Experimental Astronomy, 2001, Vol. 12, No. 1, pp. 21–31.

[9] O. Fuentes and R. C. Nelson: Learning dextrous manipulation skills for multifingered robot hands using the evolution strategy, Machine Learning, 1998, Vol. 31, pp. 223–237.

[10] O. Fuentes and R. C. Nelson: Learning dextrous manipulation skills for multifingered robot hands using the evolution strategy, Autonomous Robots, 1998, Vol. 5, pp. 395–405.

[11] M. Masotti, S. Falsaperla, H. Langer, S. Spampinato, and R. Campanini: Application of Support Vector Machine to the classification of volcanic tremor at Etna, Italy, Geophysical Research Letters, 2006, Vol. 33, Paper L20304.

[12] W. Marzocchi and M. S. Bebbington: Probabilistic eruption forecasting at short and long time scales, Bulletin of Volcanology, 2012, Vol. 74, pp. 1777–1805.

[13] J. A. Power, S. D. Stihler, B. A. Chouet, M. M. Haney, and D. M. Ketner: Seismic observations of Redoubt Volcano, Alaska – 1989-2010 and a conceptual model of the Redoubt magmatic system, Journal of Volcanology and Geothermal Research, 2013, Vol. 259, p. 14.

[14] S. Preitl, R. E. Precup, and Z. Preitl: Development of conventional and fuzzy controllers and Takagi-Sugeno fuzzy models dedicated for control of low order, Act Polytechnica Hingarica, 2005, Vol. 2, No. 1, pp. 75–92.

[15] R. Tonini, L. Sandri, D. Rouwet, C. Caudron, W. Marzocchi, and Suparjan: A new Bayesian Event Tree tool to track and quantify volcanic unrest and its application to Kawah Ijen volcano, Geochemistry, Geophysics, Geosystems, 2016, Vol. 17, pp. 2539–2555.

[16] United States Geological Survey (USGS) website http://pubs.usgs.gov/

[17] C. Werner, P. J. Kelly, M. Doukas, T. Lopez, M. Pfeffer, R. McGimsey, and C. Neal: Degassing of CO_2, SO_2, and H_2S associated with the 2009 eruption of Redoubt Volcano, Alaska, Journal of Volcanology and Geothermal Research, 2013, Vol. 259, pp. 270–284.

Clustering of Metagenomic Data by Combining Different Distance Functions

Isis Bonet[1], Adriana Escobar[1], Andrea Mesa-Múnera[1], Juan Fernando Alzate[2]

[1] Universidad EIA, km 2 + 200 Vía al Aeropuerto José María Córdova, Envigado, Antioquia, Colombia

[2] Centro Nacional de Secuenciación Genómica-CNSG, Facultad de Medicina, Universidad de Antioquia, Calle 67 Número 53-108, Medellín, Antioquia, Colombia

isis.bonet@eia.edu.co, aescobarvasco@mail.stmarytx.edu, andrea.mesa28@eia.edu.co, jfernando.alzate@udea.edu.co

Abstract: Metagenomics allows researchers to sequence genomes of many microorganisms directly from a natural environment, without the need to isolate them. The results of this type of sequencing are a huge set of DNA fragments of different organisms. These results pose a new computational challenge to identify the groups of DNA sequences that belong to the same organism. Even when there are big databases of known species genomes and some similarity-based supervised algorithms, they only have a very small representation of existing microorganisms and the process to identify a set of short fragments is very time consuming. For all those reasons, the reconstruction and identification process in a set of metagenomics fragments has a binning process, as a preprocess step, in order to join fragments into groups of the same taxonomic levels. In this paper, we propose a clustering algorithm based on k-means iterative and a consensus of clusters using different distance functions. The results achieved by the proposed method are divided using different lengths of sequences and different combinations of distances. The proposed method outperforms the simple and iterative k-means.

Keywords: Metagenomics; consensus clustering; sequences binning; k-means; distances function

1 Introduction

The study of microorganisms gives us a better understanding of global cycles that keep the biosphere in balance. Furthermore, it is important to know their functions in order to develop antimicrobial therapies and provide solutions to the environmental challenges of today.

A few years ago, the study of microorganisms consisted of isolating them in a laboratory under artificial culture conditions. After this step, suitable for a minor fraction of them, microbes can be studied to understand its biochemical and molecular properties. All its genetic information, the genome, was studied sequencing millions of partial fragments of its chromosome using sequencing machines. This short sequenced fragments are generally called reads and require an assembly process, which consists of reconstructing the entire chromosome DNA sequence. The most difficult part of the assembly process is combining the pieces of the puzzle because the fragments vary in size and some can be very similar, albeit they come from different regions in the genome [1].

Previous efforts have focused on methods developed to isolate and cultivate more microorganisms. The problem with this strategy is that only a small percentage of microorganisms can be isolated and cultivated in a laboratory setting [1, 2].

The development of more advanced sequencing technologies has led to the emergence of the metagenomics field, which made the dream of sequencing of samples directly from their natural habitats a reality. This new field has made it possible to study communities of microorganisms from different environments such as land, sea, or even the human gut, without the need to culture them [3-5].

Metagenomics does have limitations and arises new problems: now we can obtain DNA genomic sequences without the need to isolate and cultivate the organisms in a laboratory, but with this method we cannot obtain the entire genome of an organism, we can only obtain DNA fragments [2]. The presence of a variety of organisms increases the difficulty of reconstructing the DNA sequence. Metagenomics provides scientists with a set of DNA fragments from a variety of organisms that need to be sorted for processing, this process is called binning, and it consists of identifying which groups of DNA fragments belong to a single organism, a single chromosome. In order to improve the results of binning, is common to make a partial assembly to obtain largest fragments called contigs.

Research of the binning process have focused on two methodologies: composition-based and similarity-based methods [6]. Similarity-based binning is a supervised method, which uses similarity techniques such as alignment, comparing the metagenomic sequences with known genes or proteins in available databases, such as BLAST [7].

Composition-based binning is based on representing the sequences with characteristics that allow them to be separated into taxonomical groups. The most common features used to describe the sequences are GC content, codon usage or oligonucleotide frequencies. Composition-based methods can be implemented as either supervised or unsupervised depending on the use of a reference training set. NBC [8], TACOA [9] and Phymm [10] are some examples of supervised implementations.

Although supervised methods are more accurate than unsupervised methods, the availability of enough reference training sets are small which leads to the use of unsupervised methods or the combination of both methods.

There have been some research on unsupervised binning methods, which use different clustering algorithms, distance measures and features to characterize the DNA fragments. One of the first reported was TETRA [11], which uses the *k*-mers feature, with *k*=4 also known as tetranucleotide frequencies. MetaCAA [12] is another program which also uses *k*-mers as feature representation. In [13] a Self-Organizing Maps (SOM) method was used to efficiently cluster complex data using the oligonucleotide frequencies calculation, while in [14] growing self-organizing maps was used. In [15] the authors used a fuzzy *k*-means algorithm based on GC percentage and oligonucleotides frequencies. MetaCluster is another method that employs a *k*-median algorithm and *k*-mers to represent the features [16, 17]. Other researchers have used clustering methods based on expectation maximization (EM) [18] [19].

Also, some authors have presented hybrid algorithms that combine the composition-based methodology along with alignment-based methods such as PhymmBL [10] and new versions of MetaCluster [17]. The alignment-based methods are limited when dealing with large-scale sequence data due to their computational complexity and are time-consuming. Taking into account that, we focused on composition-based methodologies.

There are some issues that can arise from a binning process such as: the databases are large and heterogeneous, the number of species in a sample is unknown, fragments vary in size and the number of fragments from each species is different, which results in an unbalanced database. These problems increase the difficulty for unsupervised binning, and require better attributes to represent the DNA fragments to be determined and improved algorithms that can handle large amounts of complex data.

In this paper we propose a clustering method based on *k*-means++ and the mixture of different distance functions. We use *k*-mers frequencies as representation of features. The results of our new method were compared with the results from a simple clustering algorithm by comparing the purity of the groups created using each method.

The remainder of this paper is structured as follows. Section 2 describes the data used to create the metagenomic database, the features selected to describe the sequences and the *k*-means++ clustering method. Section 3 introduces our proposed method based on *k*-means ++ iterative. Section 4 discusses the results obtained. The paper ends with the conclusions.

2 Methods and Data

The aim of this section is to introduce the data and methods used in our experiments. We describe the composition of our database with regard to size of metagenomic sequences and the diversity of the organisms.

We present the composition-based feature used to represent metagenomic sequences. We also introduce the *k*-means++ clustering method because it is the base of our proposed algorithm. In addition, we present the distance functions used in the algorithm and the quality measures used to compare the results.

2.1 Data

Assembled genomic sequences at contig level of different organisms including viruses, bacteria and eukaryotes were downloaded from the FTP site of the Sanger institute (ftp://ftp.sanger.ac.uk). In order to have representations of different groups of domains, but also a variety within each group, the database consists of 9 eukaryotes, 2 bacteria and 5 viruses.

Table 1. Organisms in the database

Organism	Domain	Contigs	Min Length	Max Length
Ascaris suum	Eukaryote	137650	50	30000
Aspergillus fumigatus	Eukaryote	295	1001	29660
Bacteroides dorei	Bacteria	1928	500	29906
Bifidobacterium longum	Bacteria	18	540	26797
Bos taurus	Eukaryote	315841	101	5000
Candida parasilopsis	Eukaryote	1540	1003	29956
Chikungunya	Virus	1	11826	11826
Dengue	Virus	64	10392	10785
Ebola	Virus	1	18957	18957
Glossina morsitans	Eukaryote	20334	101	29996
HIV	Virus	1	9181	9181
Influenza	Virus	8	853	2309
Malus domestica	Eukaryote	66739	102	5000
Manihot esculenta	Eukaryote	7192	1998	4998
Pantholops hodgsonii	Eukaryote	159729	50	5000
Zea mays	Eukaryote	161235	102	5000
		872576	**50**	**30000**

Selected viral sequences include HIV, Chikungunya, Ebola, Influenza and Dengue virus genomes. Bacterial sequences come from Bacteroides dorei and Bifidobacterium longum. Eukaryotes include 9 species: Ascaris suum (parasitic nematode), Aspergillus fumigatus (Filamentous fungi), Bos taurus (Cow), Candida parasilopsis (Yeast fungi), Glossina morsitans (Insect), Malus domestica (Apple tree), Manihot esculenta (Cassava), Pantholops hodgsonii (Tibetan antelope) and Zea mays (Corn plant).

Table 1 shows the description of each species included in the database, and provides the number of contigs for each species and the range of lengths for each (minimum and maximum). The table shows how heterogeneous the database is.

The variation in the number of contigs for each organism as well as the size of the contigs is very large. For example, Ascaris suum is an eukaryote has 137650 contigs that range between 50 and 30000 bases, while HIV only has a single contig with 9181 bases.

2.2 Features

The database consists of 872576 contigs in total which vary in size between 50 and 30000 nucleotides bases, we used a composition-based feature to represent the DNA fragments.

Taking into account some previous results [20], we select *k*-mer (*k*=4) as the features to represent the contigs.

A tetranucleotide is a 4-combination of the nucleotides that means, there are 256 possible tetranucleotides. For each tetranucleotide *t* a 4-mer feature define follows, resulting in 256 features.

4-mer$_t$ (contig $_i$): number of each tetranucleotide (*t*) and normalized with the total of tetranucleotides in the contig.

This feature was represented as the percent of each tetranucleotide in the fragment, because it was normalized with the total of tetranucleotides in the contig.

2.3 Clustering Method

K-means is one of the most popular clustering methods [21]. *K*-means++ is a variant of *k*-means, which improves the selection of centroids for the clusters [22]. This algorithm finds a set of *k* centroids based on a weighted probability distribution where a point *x* is chosen with a probability proportional to a distance function. This selection ensures that the centroids are distant from one another. After the centroids are chosen, the algorithm proceeds as the standard *k*-means clustering.

We test different clustering algorithms as SOM, EM and *k*-means, but only the *k*-means++ method results are included because they provided the best results.

Some of the most used distance functions in this problem are Euclidean (1), Cosine (2) and Jaccard distance (3).

$$Euclidean(X,Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{1}$$

$$Cosine(X,Y) = 1 - \frac{\sum_{i=1}^{n} (x_i \times y_i)}{\sqrt{\sum_{i=1}^{n} (x_i)^2} \times \sqrt{\sum_{i=1}^{n} (y_i)^2}} \tag{2}$$

$$Jaccard(X,Y) = \frac{\sum_{i=1}^{n} (x_i - y_i)^2}{\sum_{i=1}^{n} (x_i)^2 + \sum_{i=1}^{n} (y_i)^2 - \sum_{i=1}^{n} (x_i \times y_i)} \tag{3}$$

where *X* and *Y* are the instance to compare, with dimension *n* (features number), and x_i and y_i denote the *i*-th feature of *X* and *Y* respectively.

To assess the final quality of clustering methods we use a labeled database, intra and inter-cluster, and purity measure (4) [20].

$$Purity(C_j) = \frac{\max(n_{ij})}{n_j} \tag{4}$$

where n_j is the number of organisms in cluster *j* (C_j) and n_ij is the number of organisms of class *i* in cluster *j.*

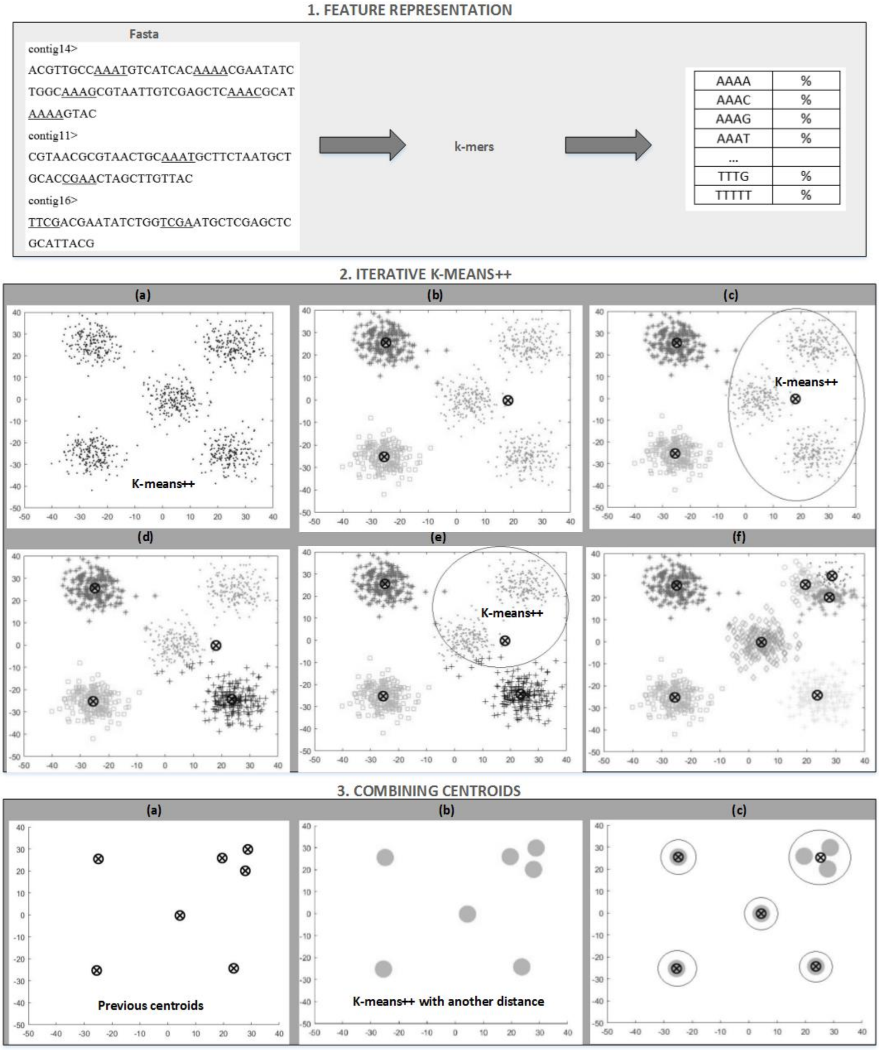
For the implementation of the clustering methods, we used Weka 3.9 [23], which is a free machine learning package that has implemented *k*-means++. Furthermore, it has the advantage that it is easy to add a new clustering method.

3 Clustering Method based on Combine Different Distances

We have proposed a clustering method based on iterative clustering [20] with *k*-mean++ as the base method. The main idea behind this proposal is the usage of multiple distances, in order to make different divisions of space input.

The proposal is a general clustering method, which is easily adaptable to any clustering problem. The principal parameters to adjust as in k-means method are the value of k and the distance function. Now we need to select not only one distance function, but two or more. We can suggest a distance function, because the best distance functions may be different depending on the problem. Nevertheless, for metagenomics we can make a suggestion based on the results obtained in that experiment.

We use Euclidean (1), Cosine (2) and Jaccard (3) distance in order to test our method in the metagenomic database.


1. FEATURE REPRESENTATION
Fasta
contig14>
ACGTTGCCAAATGTCATCACAAAACGAATATC
TGGCAAAGCGTAATTGTCGAGCTCAAACGCAT
AAAAGTAC
contig11>
CGTAACGCGTAACTGCAAATGCTTCTAATGCT
GCACCGAACTAGCTTGTTAC
contig16>
TTCGACGAATATCTGGTCGAATGCTCGAGCTC
GCATTACG
k-mers

AAAA	%
AAAC	%
AAAG	%
AAAT	%
...	
TTTG	%
TTTTT	%

Figure 1. Steps of Proposed Clustering Method

The distance functions used to compare the length of the contigs was the Euclidean and Cosine. The second step of the method joins some clusters using the Jaccard distance.

Figure 1 represents the process of our proposed clustering method, which is based on the following steps:

Step 1: The fasta file containing metagenomic sequence data is converted into a composition-based file using *k*-mer features. In that case we use an .arff file representation with Weka.

To clarify the algorithm, the following steps are depicted in the following example.

Step 2: The given sample of data points (figure 2a), are intentionally distributed in five clearly separable groups. We expect that a clustering algorithm selects the centroids in such a way that each one are in different visible groups, but we don't know the number of clusters that will be created. For example using *k*=3 you can obtain the result displayed in 2b, here it's obvious that the cluster on the far right is not compact, and it can be divided again. This is where the iterative method begins by applying *k*-means++ while there is at least one non-compact cluster.

To measure the compactness of the clusters an internal evaluation based on the intra-cluster distance is used. The method seeks clusters with high intra-cluster distance and uses data that belongs to them as the input for the next clustering. In the example the cluster on the right side of figure 2c was selected and the *k*-means++ was applied again with *k*=2, which resulted in the partition displayed in 2d. It is clear that the cluster at top right of figure 2e is still non-compact, so the *k*-means++ is applied again, now with *k*=4. Figure 2f represents the final partition results.

The threshold used to evaluate the compactness of clusters is based on the intra-cluster distance and the standard deviation as shown in equation 5.

$$Threshold = Mean_{i=1}^{n}(intra-cluster\,distance_i) + Std_{i=1}^{n}(intra-cluster\,distance_i) \quad (5)$$

where n is the number of clusters, and the intra-cluster distance for a given cluster is calculated as the average of distances between each instance that are contained in the cluster and the centroid of the cluster.

The application of iterative clustering methods generally provides more clusters than necessary, but this is the intention of this step. We want to obtain an over-estimated number of clusters. Indeed, the intention is to define double the number of clusters provided by the expert, in an attempt to define clusters with members of a single species, even when the species are divided in different groups.

Step 3: After the application of the iterative clustering method, we have a large number of clusters, some of which can be close to each other. The final step is to decrease the number of clusters. The new number of clusters is approximated using an inter-cluster measure. Once again, *k*-means++ is used, but this time with the final centroids obtained in the previous step as the input (figure 3a and 3b). Additionally, in order to partition the data space in a different way, a different distance measure was used (we suggest Jaccard distance). The results of this final clustering are shown in figure 3c, and the total number of clusters is smaller.

The previous example is an extremely simple case which was only intended to clarify the application of our proposed method. In metagenomics applications a greater number of clusters will be created than the real number of groups present in the sample, but we manage to decrease the number of clusters generated keeping the pure, and belonging to a single species by using our method. This was corroborated by using the purity measure to compare the results obtained.

It is important to remember that high purity is easy to achieve when the number of clusters is large [20]. This method provided positive results generating less clusters that are equally pure.

4 Results and Discussion

A metagenomic database built from 16 different organisms is used to evaluate the method. *K*-mers with *k*=4 are some of the characteristics selected to describe the metagenomic sequences. Euclidean and Cosine distances were used for the iterative *k*-means++ algorithms, while Jaccard distance was used in the last step of the algorithm to calculate the intra-cluster distances. Multiple tests to train the clustering method were performed, beginning with *k* of 15 and increasing until it was equal to 40. *K* was increased in the second clustering process by a factor of 2 to 10 times.

With the objective of comparing species in the same domain, we divide the database into three datasets: Bacteria, Virus and Eukaryotes. We also test the algorithm with the complete database.

The results obtained with the species divided in domain were very good for all the domains. Viruses, which are normally the most difficult to separate from the rest of organisms obtained the best results when they were analyzed alone.

Figure 2 illustrates the best results obtained using the proposed method with the dataset of Virus. The results shown on the left side of the drawing are based on an iterative method with cosine distance of two phases of *k*-means++ with *k*=10 for first phase and *k*=25 for second phase resulting 26 clusters. The last step of the method applying *k*-means++ with Jaccard distance and *k*=14 is shown at the top right of figure.

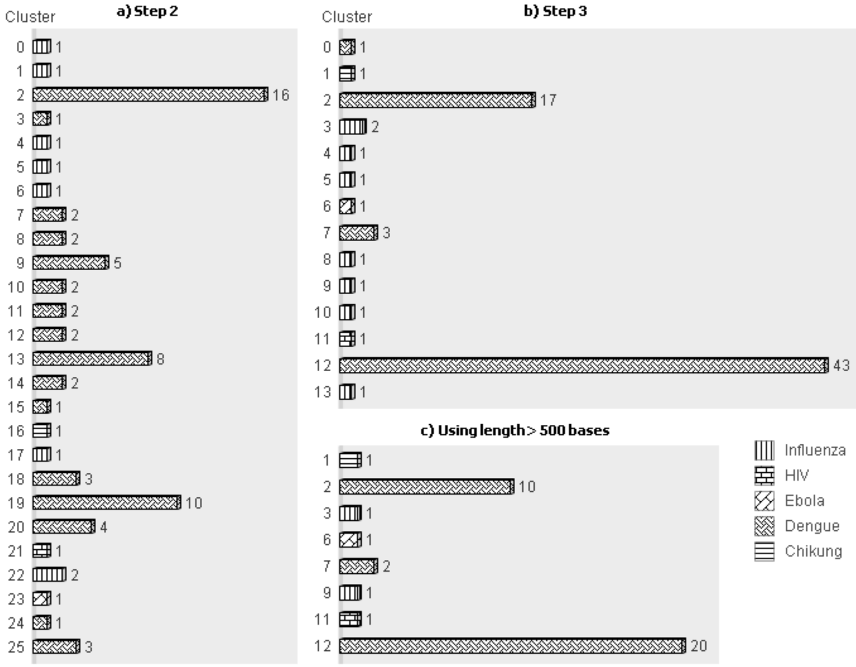


Figure 2. Results using proposed method with dataset of Virus.

Even when we obtain 14 clusters for 5 species, these clusters are 100% pure. Analyzing the results of iterative clustering described in step 2 (left side) it can be seen that Chikungunya, Ebola and HIV, which have one contig, could be separated in one cluster of each one. On the other hand, Dengue and Influenza are divided into 16 and 7 clusters respectively. The last clustering in the centroids corresponding to step 3 (top right), demonstrated that clusters of Chikungunya, Ebola and HIV remain the same and Dengue and Influenza are reduce to 4 (clusters 0, 2, 7 and 12) and 7 (clusters 3, 4, 5, 8, 9, 10 and 13) clusters.

The best results for Bacteria were obtained with *k*1=10 and *k*2=20. The second phase was using *k*=4 obtaining 97.5% of purity. Even when we have only two bacteria, Bifidobacterium longum was difficult to separate. Viruses have more number of species than the Bacteria, but Bacteria have more contigs and are bigger than Viruses. Bacteria have 1946 contigs ranging between 500 and 29906 bases, whereas Viruses present only 75 contigs in a range between 92 and 8748.

On the other hand Eukaryote which are the biggest in both number of species and contigs, achieved 99.7% of purity. The results were obtained with *k*1=25 and *k*2=30 in iterative process and *k*=4 for the last part. Candida parasilopsis and Aspergillus fumigatus are scatted across many clusters and could not be separated from the rest.

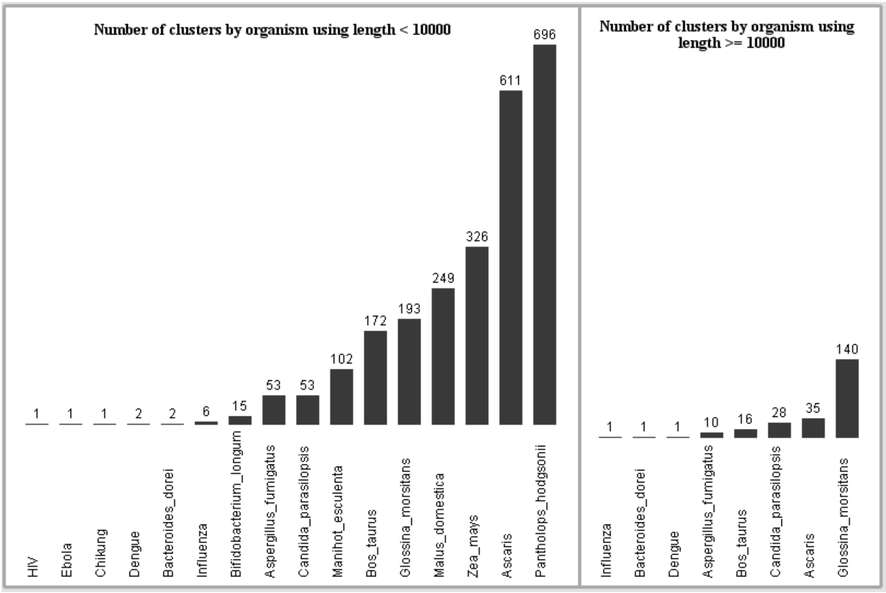


Figure 3. Results of proposed clustering methods with all domain of species, divided according the length of contigs.

Figure 3 shows the results of the whole database, now divided according to length of contigs. Left side of drawing illustrates the outcomes of proposed clustering method with length of contigs inferior to 10000 and using $k1$=15, $k2$=2500 for the iterative process resulting 2483 clusters. Right side of figure shows results also about the performance of iterative phase, but with contigs length greater than or equal to 10000 and using $k1$=15 and $k2$=230. And this time 232 clusters were obtained.

Although the organisms are very scattered we obtain a high purity and we can separate the virus in independent clusters. Using lengths larger than 10000, the algorithm achieves 100% of purity for all clusters, and 98.11% when the lengths are shorter than 10000.

In order to reduce the number of clusters we join the results obtained before having now 2715 centroids and apply the last phase of the algorithm. The number of clusters was reduced from 2715 to 125 yielding a 99% of purity. Figure 4 displays the number of clusters by organism which oscillate into 1 to 37. Even when at least a cluster was obtained for each organism, three species cannot be totally separated from the rest: Bifidobacterium longum, Influenza and Aspergillus fumigatus. Some of their contigs are distributed in different clusters with other organisms.

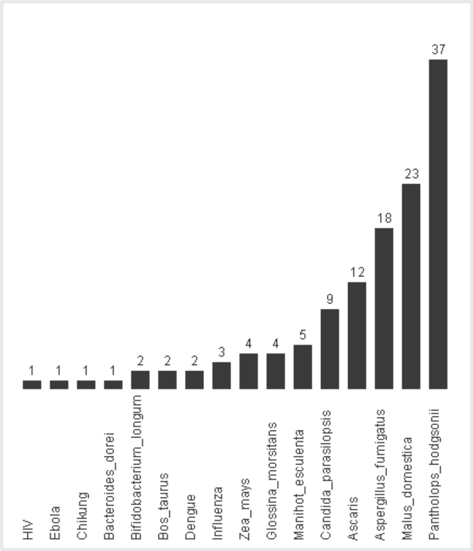


Fig. 4. Final result by the application of step 3 with k=125

Summarizing, the proposed method based on two process of clustering one iterative and another with the centroids resulting for the first improves the results of binning in metagenomics. The key of the method is to use different distances for iterative clustering and centroids clustering. The combination of different distances can generate a significant change in the separation of the space. Here we use Cosine and Jaccard distances in the first and in the last clustering process respectively.

Additionally, it is important to take into account the lengths of the sequences. In order to minimize error we can divide the problem and create models that are focused on the short sequences and models that are based on the large sequences.

Conclusions

In this paper we proposed a clustering method based on *k*-means++ with two training phases. The first phase is an iterative clustering process, training a consecutive set of *k*-means++ with different inputs, decreasing in each one depending on the compact clusters determined in the previous run of clustering. The second phase is another clustering but using the set of centroids obtained from the first phase. Each phase is trained with a different distance function.

The proposed method is applied to a metagenomic database that composed of 16 different organisms from three different domains: Bacteria, Virus and Eukaryote.

We obtained the best result using Cosine and Jaccard distance for the first and second phase respectively. The results obtained, based on the purity of clusters, outperforms results obtained with a simple *k*-means++ and also compared with an iterative *k*-means++.

We can conclude that longer DNA fragments can improve performance in a binning process. Although, the number of clusters is higher than the number of organisms, the proposed method provided pure clusters for organisms, achieving 100% of purity in all clusters when the lengths of contigs is greater than 10000, and 99% for all possible lengths.

The results shown have only been applied to one database, but the method is a promising development for clustering larger sequences or as a prior step in the taxonomy assigned process.

References

[1] R. S. Lasken and J. S. McLean, "Recent advances in genomic DNA sequencing of microbial species from single cells," *Nat Rev Genet,* Progress vol. 15, no. 9, pp. 577-584, 2014.

[2] N. R. Council, *The New Science of Metagenomics: Revealing the Secrets of Our Microbial Planet*. The National Academies Press, 2007.

[3] L. Nanni and A. Lumini, "MppS: An ensemble of support vector machine based on multiple physicochemical properties of amino acids," (in English), *Neurocomputing,* Article vol. 69, no. 13-15, pp. 1688-1690, Aug 2006.

[4] A. Oulas *et al.*, "Metagenomics: Tools and Insights for Analyzing Next-Generation Sequencing Data Derived from Biodiversity Studies," in *Bioinform Biol Insights*, vol. 9, 2015, pp. 75-88.

[5] C.-K. Chan, A. Hsu, S. Halgamuge, and S.-L. Tang, "Binning sequences using very sparse labels within a metagenome," *BMC Bioinformatics,* vol. 9, no. 1, p. 215, 2008.

[6] A. Kislyuk, S. Bhatnagar, J. Dushoff, and J. Weitz, "Unsupervised statistical clustering of environmental shotgun sequences," *BMC Bioinformatics,* vol. 10, no. 1, p. 316, 2009.

[7] C. Camacho *et al.*, "BLAST+: architecture and applications," *BMC Bioinformatics,* vol. 10, no. 1, p. 421, 2009.

[8] G. L. Rosen, E. Reichenberger, and A. Rosenfeld, "NBC: The Naïve Bayes Classification Tool Webserver for Taxonomic Classification of Metagenomic Reads," *Bioinformatics,* November 8, 2010 2010.

[9] N. N. Diaz, L. Krause, A. Goesmann, K. Niehaus, and T. W. Nattkemper, "TACOA – Taxonomic classification of environmental genomic fragments using a kernelized nearest neighbor approach," *BMC Bioinformatics,* vol. 10, pp. 56-56, 2009.

[10] A. Brady and S. L. Salzberg, "Phymm and PhymmBL: Metagenomic Phylogenetic Classification with Interpolated Markov Models," *Nature methods,* vol. 6, no. 9, pp. 673-676, 2009.

[11] H. Teeling, J. Waldmann, T. Lombardot, M. Bauer, and F. Glockner, "TETRA: a web-service and a stand-alone program for the analysis and

comparison of tetranucleotide usage patterns in DNA sequences," *BMC Bioinformatics,* vol. 5, no. 1, p. 163, 2004.

[12] R. M. Reddy, M. H. Mohammed, and S. S. Mande, "MetaCAA: A clustering-aided methodology for efficient assembly of metagenomic datasets," *Genomics,* vol. 103, no. 2–3, pp. 161-168, 2014.

[13] T. Abe, S. Kanaya, M. Kinouchi, Y. Ichiba, T. Kozuki, and T. Ikemura, "Informatics for Unveiling Hidden Genome Signatures," *Genome Research,* vol. 13, no. 4, pp. 693-702, April 1, 2003 2003.

[14] H. Zouari, L. Heutte, and Y. Lecourtier, "Controlling the diversity in classifier ensembles through a measure of agreement," (in English), *Pattern Recognition,* Article vol. 38, no. 11, pp. 2195-2199, Nov 2005.

[15] K. Woods, W. P. Kegelmeyer, and K. Bowyer, "Combination of multiple classifiers using local accuracy estimates," (in English), *Ieee Transactions on Pattern Analysis and Machine Intelligence,* Article vol. 19, no. 4, pp. 405-410, Apr 1997.

[16] H. C. Leung *et al.*, "A robust and accurate binning algorithm for metagenomic sequences with arbitrary species abundance ratio," (in eng), *Bioinformatics,* vol. 27, no. 11, pp. 1489-95, Jun 1 2011.

[17] Y. Wang, H. Leung, S. Yiu, and F. Chin, "MetaCluster-TA: taxonomic annotation for metagenomic data based on assembly-assisted binning," (in English), *BMC Genomics,* vol. 15, no. 1, pp. 1-9, 2014/01/24, Art. no. S12

[18] I. Partalas, G. Tsoumakas, I. Katakis, and I. Vlahavas, "Ensemble pruning using reinforcement learning," in *Advances in Artificial Intelligence, Proceedings*, vol. 3955(Lecture Notes in Computer Science, Berlin: Springer-Verlag Berlin, 2006, pp. 301-310.

[19] L. Nanni and A. Lumini, "FuzzyBagging: A novel ensemble of classifiers," *Pattern Recognition,* vol. 39, no. 3, pp. 488-490, Mar 2006.

[20] I. Bonet, W. Montoya, A. Mesa-Múnera, and J. Alzate, "Iterative Clustering Method for Metagenomic Sequences," in *Mining Intelligence and Knowledge Exploration*, vol. 8891, R. Prasath, P. O'Reilly, and T. Kathirvalavakumar, Eds. (Lecture Notes in Computer Science: Springer International Publishing, 2014, pp. 145-154.

[21] J. MacQueen, "Some methods for classification and analysis of multivariate observations," presented at the Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics, Berkeley, Calif., 1967, 1967. Available: http://projecteuclid.org/euclid.bsmsp/1200512992

[22] D. Arthur and S. Vassilvitskii, "K-Means ++: The Advantages of Careful Seeding," in 8th *Annual ACM-SIAM Symposium on Discrete Algorithms*, New Orleans, 2007, pp. 1027-1035.

[23] I. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed. San Francisco: Morgan Kaufmann, 2005, p. 525

Gaussian and Cauchy Functions in the Filled Function Method – Why and What Next: On the Example of Optimizing Road Tolls

José Guadalupe Flores Muñiz[1], Vyacheslav V. Kalashnikov[2,3], Vladik Kreinovich[4], and Nataliya Kalashnykova[1,5]

[1]Department of Physics and Mathematics
Universidad Autónoma de Nuevo León
Av. Universidad S/N, Ciudad Universitaria
San Nicolás de los Garza, México 66455
jose.floresmnz@uanl.edu.mx, nataliya.kalashnykova@uanl.edu.mx
[2]Department of Systems and Industrial Engineering
Instituto Tecnológico y de Estudios Superiores de Monterrey
Av. Eugenio Garza Sada 2501
Monterrey, Nuevo León, México 64849
kalash@itesm.mx
[3]Department of Experimental Economics
Central Economics and Mathematics Institute (CEMI)
47 Nakhimovsky prospect, 117418, Moscow, Russia
[4]Department of Computer Science, University of Texas at El Paso
500 W. University, El Paso, Texas 79968, USA, vladik@utep.edu
[5]Department of Computer Science, Sumy State University,
Ryms'koho-Korsakova Str., 2, 40007, Sumy, Ukraine

Abstract: In many practical problems, we need to find the values of the parameters that optimize the desired objective function. For example, for the toll roads, it is important to set the toll values that lead to the fastest return on investment.

There exist many optimization algorithms, the problem is that these algorithms often end up in a local optimum. One of the promising methods to avoid the local optima is the filled function method, in which we, in effect, first optimize a smoothed version of the objective function, and then use the resulting optimum to look for the optimum of the original function. It turns out that empirically, the best smoothing functions to use in this method are the Gaussian and the Cauchy functions. In this paper, we show that from the viewpoint of computational complexity, these two smoothing functions are indeed the simplest.

The Gaussian and Cauchy functions are not a panacea: in some cases, they still leave us with a local optimum. In this paper, we use the computational complexity analysis to describe the next-simplest smoothing functions which are worth trying in such situations.

Keywords: optimization; toll roads; filled function method; Gaussian and Cauchy smoothing

functions

1 Optimizing Road Tolls: A Brief Introduction to the Case Study

1.1 Optimizing road tolls: a general description of the problem

In many practical problems, we need to optimize an appropriate objective function. In this paper, as a case study, we consider the problem of optimizing road tolls; see [7] for details.

The need for road tolls comes from the fact that in many geographic locations, traffic is congested, there is a need to build new roads that would decrease this congestion. Often, however, the corresponding governments do not have the funds to build the new roads.

A solution is to build the *toll roads*, i.e., to request that the drivers pay for driving on these roads – and thus, to get back the money that was spent on building these roads. Sometimes, the governments borrow the money to build the roads, and use the collected tolls to pay back the loan. In other cases, a private company is selected to build the road: the company invests the money, and get its investment back from the collected tolls.

In both arrangements, for a system of toll roads, it is important to select the toll values that will lead to the fastest possible return on investment. This is a complex problem:

- if the tolls are too small, it will take forever to get back the investments;
- on the other hand, if the tolls are too large, then most drivers will prefer to use the existing toll-free roads, and again, it will take a long time to get back the investment.

It is therefore important to find the optimal toll values that minimize the amount of time needed to return the investment.

Let us describe this optimization problem in detail.

1.2 Describing the road network

The transportation network is usually modeled as a graph, in which nodes are spatial locations (points), and arcs (edges) are road segments.

The set of all the nodes ("points") of this graph is denoted by P, and the set of all arcs ("edges") connecting the nodes is denoted by E.

Some of the road-segments are one-way. For each node p:

- the set of all road segments that have p as origin is denoted by p^+, and

- the set of all road segments that have p as the arrival node is denoted by p^-.

Some of the arcs are toll road segments. The set of all such segments is denoted by E_1. The remaining toll-free arcs is denoted by $E_2 \stackrel{\text{def}}{=} E - E_1$.

For each arc e, there is an upper bound ℓ_e on its capacity.

1.3 Describing travel costs

For each arc $e \in E$, we know the cost d_e of moving a unit of cargo along this arc. This cost comes from the fuel spent on this trip, driver's salary, wear and tear of the vehicle, etc.

For the toll roads, the drivers also have to pay the appropriate toll c_e per unit, so the cost per unit weight is now $d_e + c_e$.

Usually, for each road segment, there is some pre-negotiated limit $c_e^{\max}$ on how much toll we can connect. So, possible toll values c_e must satisfy the inequality

$$0 \le c_e \le c_e^{\max}.$$

1.4 Describing the travel demand

Theoretically, we could have the need for transporting goods between all possible pairs of points. In reality, the number of such pairs is limited. Let C denote the set of all origin-destination pairs.

For each pair $k \in C$:

- its origin (home point) is denoted by $h(k)$,
- its destination (aim) is denoted by $a(k)$, and
- the overall amount of goods to be transported is denoted by q^k.

It is convenient to use the following auxiliary notation n_p^k, where $p \in P$:

- $n_p^k = -q^k$ if $p = h(k)$ is the origin node;
- $n_p^k = q^k$ if $p = a(k)$ is the destination node, and
- $n_p^k = 0$ for all other nodes p.

1.5 For each origin-destination pair, how the optimal routes are selected

For each origin-destination pair $k \in C$, we need to select the traffic $x_e^k \ge 0$ along each road segment is such a way that:

- the overall traffic leaving the starting node $h(k)$ is equal to q^k,

- the overall traffic arriving at the destination node $a(k)$ is equal to q^k, and
- in all other nodes, the amount of incoming traffic is equal to the amount of outgoing traffic.

Because of the above notation n_p^k, these three conditions can be described in a similar way for all the nodes p:

$$\sum_{e\in p^+} x_e^k - \sum_{e\in p^-} x_e^k = n_p^k.$$

Among all the arrangements $x_e^k \geq 0$ that satisfy all these equalities, we need to select the one that minimizes the overall cost

$$\sum_{e\in E_1} (d_e + c_e)\cdot x_e^k + \sum_{e\in E_2} d_e \cdot x_e^k.$$

1.6 Final formulation of the problem: how should we select the toll amounts?

We need to select the tolls $c_e \in [0, c_e^{\max}]$ in such a way that when all the customers $k \in C$ optimize their routes, the overall traffic on each road segment e does not exceed the capacity of this segment:

$$\sum_{k\in C} x_e^k \leq \ell_e.$$

Among all the toll arrangements c_e that satisfy this condition, we must select the one that maximizes the overall return on our investment, i.e., that maximizes the sum

$$\sum_{k\in C}\sum_{e\in E_1} c_e \cdot x_e^k.$$

2 Optimization in General: How to Avoid Local Optima?

2.1 Local optima: a problem

There exist many optimization algorithms, including both traditional techniques and meta-heuristic algorithms such as simulated annealing, genetic algorithms, differential evolution, ant colony optimization, bee algorithm, particle swarm optimization, tabu search, harmony search, firefly algorithm, cuckoo search, etc.; see, e.g., [4].

However, often, they lead to a local optimum; see, e.g., [2, 4, 5, 6]. To be more precise, the need to avoid a local optimum – or at least get to a different local optimum which is closer to the global one – is one of the main reasons why meta-heuristic optimization techniques were invented in the first place. Each of the meta-heuristic methods has indeed been successful in improving the local optima in many practical situations. However, the very fact that there exist many different meta-heuristic techniques – and that new meta-heuristics are appearing all the time – is a good indication that none of these methods is a panacea. In many practical situations, even after applying the latest meta-heuristic methods, we are still in a local optimum. There is, therefore, a need for developing new techniques that would help us avoid the local optima.

How to avoid local optima: the filled function method. One of the promising methods to avoid the local optima is the *filled function* method, in which we, in effect,

- first optimize a smoothed version of the objective function, and
- then use the resulting optimum to look for the optimum of the original function.

This method was originally proposed in [9]; see also [1, 7, 10, 11]. In particular, in these papers, it was shown that in some practical situations, this method indeed enables us to improve the solution in comparison with a local optimum x^* produced by either one of the traditional optimization techniques, or by one of the meta-heuristic optimization methods.

In the filled function method, once we reach a local optimum x^*, then we optimize an auxiliary expression

$$K\left(\frac{x-x^*}{\sigma}\right)\cdot F(f(x),f(x^*),x)+G(f(x),f(x^*),x),$$

for appropriate functions $K(x)$, $F(f,f^*,x)$, and $G(f,f^*,x)$, and for an appropriate value σ. Once we find the optimum of this auxiliary expression – by using traditional optimization or by using one of the known meta-heuristic optimization methods – we use the optimum of the auxiliary expression as a new first approximation to find the optimum of the original objective function $f(x)$.

2.2 Filled function method: results

How well we can avoid the local optimum depends on the choice of the smoothing function $K(x)$. In [10], it was shown that for several optimization problem, the best choice is to use the Cauchy smoothing function

$$K(x)=\frac{1}{1+\|x\|^2}.$$

For toll optimization and for several similar problems, it turned out that the Gaussian smoothing function $K(x)=\exp(-\|x\|^2)$ leads to the best results; see, e.g., [7].

In some cases, none of the known smoothing functions worked well.

2.3 Filled function method: details

Specifically, the paper [7] maximizes the following auxiliary expression:

$$\exp(-\|x-x^*\|^2)\cdot g\left(\frac{f(x)}{f(x^*)}\right)+\rho\cdot s(f(x),f(x^*)),$$

where $\rho > 0$ is an appropriate parameter, the function $g(v)$ is defined as follows:

- $g(v)=0$ if $v \le \dfrac{2}{5}$,
- $g(v)=5-30\cdot v+\dfrac{225}{4}\cdot v^2-\dfrac{125}{4}\cdot v^3$ if $\dfrac{2}{5}\le v\le \dfrac{4}{5}$, and
- $g(v)=1$ if $v\ge \dfrac{4}{5}$,

and the function $s(v,b)$ is defined as follows:

- $s(v,b)=v-\dfrac{2}{5}$ if $v\le \dfrac{2}{5}\cdot b$;
- $s(v,b)=5-\dfrac{8}{5}\cdot b+\left(8-\dfrac{30}{b}\right)\cdot v-\dfrac{25}{2b}\cdot\left(1-\dfrac{9}{2b}\right)\cdot v^2+\dfrac{25}{4b^2}\cdot\left(1-\dfrac{5}{b}\right)\cdot v^3$ if $\dfrac{2}{5}\cdot b\le v\le \dfrac{4}{5}\cdot b$;
- $s(v,b)=1$ if $\dfrac{4}{5}\cdot b\le v\le \dfrac{8}{5}\cdot b$;
- $s(v,b)=1217-2160\cdot\dfrac{v}{b}+1275\cdot\left(\dfrac{v}{b}\right)^2-250\cdot\left(\dfrac{v}{b}\right)^3$ if $\dfrac{8}{5}\cdot b\le v\le\dfrac{9}{5}\cdot b$; and
- $s(v,b)=2$ if $v\ge\dfrac{9}{5}\cdot b$.

2.4 Filled function method: open problems

Due to the above general empirical evidence, we arrive at the following natural problems:

- why are the Gaussian and Cauchy smoothing functions empirically the best?
- which smoothing function should we choose if neither Gaussian nor Cauchy smoothing functions work well?

2.5 What we do in this paper

In this paper, we provide answers to both questions.

3 Computational Complexity as a Natural Criterion for Selecting a Smoothing Function

3.1 Why computational complexity

What criterion should we use to select a smoothing function? We can always avoid a local optimum if we repeatedly start the same optimization process at several randomly selected points: if we start at many such points, one of them will be close to the global optimum. However, this will drastically increase the computation time.

The main advantage of the filled function method is that it allows us to decrease the computation time. From this viewpoint, the less time we need to compute the smoothing function, the better.

Each computation consists of several elementary computational steps, and the computation time is thus proportional to the number of such steps – maybe taken with weights. This (weighted) number of steps is known as *computational complexity*; see, e.g., [3, 8]. From this viewpoint, we want a smoothing function which has the smallest possible computational complexity.

3.2 How can we measure computational complexity

Most programming languages use the following elementary computational operations:

- arithmetic operations: unary minus $(-x)$, addition, subtraction, multiplication, and division, and
- elementary functions: $\exp(x)$, $\ln(x)$, $\sin(x)$, $\cos(x)$, $\tan(x)$, $\arcsin(x)$, $\arccos(x)$, and $\arctan(x)$.

Thus, first, we need to minimize the overall number of such computational steps.

Not all these steps require the same computation time:

- unary minus $(-x)$ is the fastest operation, it requires that we only change one bit: the bit describing the sign;
- addition and subtraction are next in complexity;
- multiplication takes somewhat longer, since multiplication, in effect, means several additions;
- finally, computation of elementary functions requires even longer time, since each such computation requires several multiplications and additions.

We will take this difference into account when deciding which smoothing function is the fastest to compute.

4 Analysis of the Problem and the Main Result

4.1 Natural requirements on a smoothing function

The smoothing function should be symmetric, since we have no reason to prefer different orientation of coordinates. Thus, it should depend only on $v \overset{\text{def}}{=} \|x\|^2$: $K(x) = g(v)$ for some function $g(v)$.

This function $g(v)$ should be finite and non-negative for all $v \geq 0$, and it should tend to 0 when $v \to +\infty$.

It is easy to see that both Gaussian and Cauchy smoothing functions satisfy these requirements, correspondingly with $g(v) = \exp(-v)$ and $g(v) = \dfrac{1}{1+v}$.

4.2 Computational complexity of the Gaussian and Cauchy smoothing functions

The function $g(v) = \exp(-v)$ (corresponding to Gaussian smoothing) requires two operations to compute:

- a unary minus, to compute $-v$, and
- the exponential function, to transform $-v$ into $\exp(-v)$.

Similarly, the function $g(v) = \dfrac{1}{1+v}$ (corresponding to Cauchy smoothing) consists of two operations:

- addition, to compute $1+v$, and
- division, to transform $1+v$ into $g(v)$.

4.3 Our first result

Our first result is a classification of all smoothing functions that can be computed in two or fewer computational steps.

Definition 1. *By a* smoothing function*, we mean a non-zero non-negative function $g(v)$ which is defined for all $v \geq 0$ and which tends to 0 as $v \to +\infty$.*

Definition 2.

- *We say that a function $g(v)$ is* computable in 0 steps *if it is either an identity $g(v) = v$ or a constant $g(v) =$* const.
- *By an* elementary operation*, we mean either an arithmetic operation (unary minus, addition, subtraction, multiplication, or division), or an elementary function (*$\exp(x)$*,* $\ln(x)$*,* $\sin(x)$*,* $\cos(x)$*,* $\tan(x)$*,* $\arcsin(x)$*,* $\arccos(x)$*, or* $\arctan(x)$*).*

- *If $F(x)$ is an elementary operation, and $h(v)$ is computable in k steps, then we say that the function $g(v) = F(h(v))$ is* computable in $k+1$ steps.
- *If $F(x,y)$ is an elementary operation, and the functions $h(v)$ and $h'(v)$ are computable, correspondingly, in k and k' steps, then we say that the function $g(v) = F(h(v), h'(v))$ is* computable in $k+k'+1$ steps.

Proposition 1. *A smoothing function is computable in 2 steps if and only if it has one of the following forms:*

- $g(v) = \dfrac{c'}{c+v}$, *for some constants c and c',*
- $g(v) = \text{const} \cdot \exp(-c \cdot v)$,
- $g(v) = \dfrac{\pi}{2} - \arctan(v)$,
- $g(v) = \arctan\left(\dfrac{1}{v}\right)$, *or*
- $g(v) = \cos(\arctan(v))$.

Comment. For convenience, the proof of this Proposition is given in the next section.

4.4 Among these five, which are the fastest to compute?

Which of the above five functions is the fastest to compute?

1. The function $g(v) = \dfrac{1}{1+v}$ requires one addition and one multiplication.
2. The function $g(v) = \exp(-v)$ requires one unary minus and one application of an elementary function.
3. The function $g(v) = \dfrac{\pi}{2} - \arctan(v)$ requires one subtraction and one application of an elementary function.
4. The function $g(v) = \arctan\left(\dfrac{1}{v}\right)$ requires one division and one application of an elementary function.
5. Finally, the function $g(v) = \cos(\arctan(v))$ requires two applications of elementary functions.

We can now make the following comparisons:

- Since multiplication/division is faster than an application of an elementary function, and addition is faster than multiplication/division and than elementary functions, the function 1 is faster to compute than functions 3, 4, and 5.
- Similarly, since the unary minus is faster than any other operation, function 2 is faster to compute than functions 3, 4, and 5.

- Since subtraction is faster than division, function 3 is faster than function 4.
- Finally, since multiplication/division is faster than an application of an elementary function, function 4 is faster than function 5.

Thus, we arrive at the following conclusion.

4.5 Conclusion

Among all smoothing functions that can be computed in two computational steps:

- the functions $g(v) = \dfrac{1}{1+v}$ and $g(v) = \exp(-v)$ corresponding to Cauchy and Gaussian smoothing are the fastest to compute;
- next fastest is the function $g(v) = \dfrac{\pi}{2} - \arctan(v)$;
- next fastest is the function $g(v) = \arctan\left(\dfrac{1}{v}\right)$; and
- finally, the slowest to compute is the function $g(v) = \cos(\arctan(v))$.

This explains why the Gaussian and Cauchy functions are indeed empirically the best, and this also show what to do when these smoothing functions do not work well: try smoothing functions $K(x) = g(\|x\|^2)$ corresponding to

$$g(v) = \frac{\pi}{2} - \arctan(v),\ \ g(v) = \arctan\left(\frac{1}{v}\right),\ \text{and}\ g(v) = \cos(\arctan(v)).$$

5 Proof of the Main Result

1°. Clearly, functions $g(v) = v$ and $g(v) = \text{const}$ which are computable in 0 steps are not smoothing functions, since they do not tend to 0 when $v \to +\infty$.

Let us show that similarly, no smoothing function can be computed in 1 step. Indeed, we can easily list all functions computable in 1 step:

$$g(v) = v + c,\ \ g(v) = v - c,\ \ g(v) = c - v,\ \ g(v) = c \cdot v,\ \ g(v) = \frac{c}{v},$$

$$g(v) = \frac{v}{c},\ \ g(v) = \exp(v),\ \ g(v) = \ln(v),\ \ g(v) = \sin(v),\ \ g(v) = \cos(v),$$

$$g(v) = \tan(v),\ \ g(v) = \arcsin(v),\ \ g(v) = \arccos(v),\ \ g(v) = \arctan(v),$$

where c is a constant.

From the above functions, the function $g(v) = \dfrac{c}{v}$ is not a smoothing function since it is not defined for $v = 0$, and all other functions are not smoothing functions since they do not satisfy the condition that $\lim\limits_{v\to+\infty} g(v) = 0$.

Thus, a smoothing function must have at least two computational steps.

2°. By definition, a function computable in 2 steps has the form $F(h(v))$, where $h(v)$ is computable in 1 step, or the form $F(h(v),h'(v))$, where $h(v)$ is computable in one step and $h'(v)$ is computable in 0 steps (i.e., is either an identity or a constant).

We have already listed all possible functions $h(v)$ which can be computed in one step. Let us consider these functions one by one.

3°. If $h(v)=v+c$, then $h(+\infty)=+\infty$. So, for the composition to be a smoothing function, we must have $F(+\infty)=0$.

As we showed in the 1-step case, the only operation that satisfies this condition is $g(w)=\frac{c'}{w}$, so we get $g(v)=\frac{c'}{v+c}$. This case corresponds to the Cauchy function.

4°. The case $h(v)=v-c$ is equivalent to $h(v)=v+(-c)$, so it is the same case that we have already considered.

5°. If $h(v)=c_1-v$, then, $h(+\infty)=-\infty$, so the function $F(w)$ must satisfy the condition $F(-\infty)=0$.

Two operations satisfy this condition: $F(w)=\frac{c'}{w}$ and $F(w)=\exp(w)$.

- If $F(w)=\frac{c'}{w}$, then, $g(v)=\frac{c'}{c-v}$. This is equal to $g(v)=\frac{(-c')}{v+(-c)}$, i.e., to the Cauchy case that we have already considered.
- If $F(w)=\exp(w)$, then, $g(v)=\exp(c-v)$, i.e., $g(v)=\text{const}\cdot\exp(-v)$, where $\text{const}=\exp(c)$. This case corresponds to the Gaussian smoothing.

6°. If $h(v)=c\cdot v$, then, depending on the sign of c, we have different asymptotic behaviors for $h(v)$.

If $c>0$, then we have $h(+\infty)=+\infty$. In this case, the only possibility to get $g(h)\to 0$ as $h\to+\infty$ is to have $F(w)=\frac{c'}{w}$, but in this case $g(v)=\frac{c'}{c\cdot v}$ is not defined for $v=0$.

If $c<0$, then $h(+\infty)=-\infty$. In this case, we similarly cannot have $F(w)=\frac{c'}{w}$, but now we have a second option $F(w)=\exp(w)$, in which case $g(v)=\exp(c\cdot v)$. This case corresponds to the Gaussian function.

7°. If $h(v)=\frac{c}{v}$, then, $h(+\infty)=0$, so the function $F(w)$ must satisfy the condition $F(0)=0$. Six operations satisfy this condition:

- $F(w)=c'\cdot w$,
- $F(w)=\frac{w}{c}$,
- $F(w)=\sin(w)$,
- $F(w)=\tan(w)$,

- $F(w) = \arcsin(w)$, and
- $F(w) = \arctan(w)$.

When $c > 0$, then $h(0) = +\infty$, so, additionally, $F(+\infty)$ must be finite and non-negative. This condition is satisfy only by $F(w) = \arctan(w)$, so we get

$$g(v) = \arctan\left(\frac{c}{v}\right).$$

When $c < 0$, then $h(0) = -\infty$, so, additionally, $F(-\infty)$ must be finite and non-negative. This condition is not met by any of the above functions $F(w)$.

8°. The case $h(v) = \frac{v}{c}$ is equivalent to the already analyzed case $h(v) = v \cdot \text{const}$, with $\text{const} = \frac{1}{c}$.

9°. If $h(v) = \exp(v)$, then, $h(+\infty) = +\infty$, so we must have $F(w) = c'w$ and

$$g(v) = \frac{c'}{\exp(v)}.$$

This is equal to $g(v) = \text{const} \cdot \exp(-v)$, i.e., corresponds to the Gaussian case.

10°. If $h(v) = \ln(v)$, then, $h(+\infty) = +\infty$, so we must have $F(w) = \frac{c'}{w}$ and, thus,

$$g(v) = \frac{c'}{\ln(v)}.$$

This function is not defined (is infinite) when $v = 1$ and thus, is not a smoothing function.

11°. If $h(v) = \sin(v)$ or $h(v) = \cos(v)$, then $h(v)$ oscillates between -1 and 1 and has no limit when $v \to +\infty$. So, for $g(v) \to 0$, the function $F(w)$ must be equal to 0 for all the values $w \in [-1, 1]$, but no elementary operation has this property.

Similarly, it is not possible to have $h(v) = \tan(v)$.

12°. If $h(v) = \arcsin(v)$ or $h(v) = \arccos(v)$, then $g(v) = F(h(v))$ cannot be a smoothing function since $h(v)$ is not defined for $v > 1$.

13°. If $h(v) = \arctan(v)$, then, $h(+\infty) = \pi/2$, so the function $F(w)$ must satisfy the condition $F\left(\frac{\pi}{2}\right) = 0$. Three elementary functions satisfy this condition:

- $F(w) = w - \frac{\pi}{2}$,
- $F(w) = \frac{\pi}{2} - w$, and

- $F(w) = \cos(w)$.

When $F(w) = w - \dfrac{\pi}{2}$, then the function $g(v) = \arctan(v) - \dfrac{\pi}{2}$ has negative values, so it cannot be a smoothing function.

The other two cases correspond to the last two function in the formulation of the Proposition.

The Proposition is thus proven.

6 Conclusions

In many practical situations, we need to find the values of the quantities that maximize the desired objective function. As an example, in this paper, we consider a difficult-to-solve problem of selecting the optimal toll values for the toll roads.

There exist many optimization techniques, from the more traditional optimization algorithms to meta-heuristic algorithms such as simulated annealing, genetic algorithms, etc. In many practical situations, the existing optimization algorithms work well. However, in many other practical cases, the existing algorithms end up with a local optimum which is far from the desired global one. To deal with such cases, when the existing optimization techniques cannot get us out of a local optimum, it is necessary to develop new optimization ideas. One of such ideas – that works well in many practical situations, including the toll road problem – is the *filled function method*.

According to this method, once we read a local optimum, we build an auxiliary smoothed (and thus, easier to optimize) objective function, use the existing techniques to optimize this auxiliary objective function, and then use the resulting optimum as a new starting point for optimizing the original objective function. In this method, different functions have been used for smoothing. It turns out that empirically, two classes of smoothing functions work best: Gaussian and Cauchy smoothing functions.

In this paper, we provide a theoretical explanation for their efficiency. Specifically, we show that these smoothing functions have the smallest computational complexity. For cases when these two smoothing functions do not work perfectly well and there is a need to try different smoothing functions, we explicit describe next-simplest smoothing functions that can be used in such situations.

Acknowledgement

This work was supported by grant CB-2013-01-221676 from Mexico Consejo Nacional de Ciencia y Tecnología (CONACYT). It was also partly supported by the US National Science Foundation grants HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and DUE-0926721, and by an award “UTEP and Prudential Actuarial Science Academy and Pipeline Initiative” from Prudential Foundation.

The authors are thankful to Ildar Batyrshin, Grigory Sidorov, Alexander Gelbukh, and to all the organizers of MICAI'2016 for their support and encouragement, and to the anonymous referees for valuable suggestions.

This work was performed when José Guadalupe Flores Muñiz visited the University of Texas at El Paso.

References

[1] B. Addis, M. Locatelli, and F. Schoen: Local optima smoothing for global optimization, Optimization Methods and Software, 2005, Vol. 20, No. 4–5, pp. 417–437.

[2] P. Cisar, S. Maravic Cisar, D. Subošic, P. Dikanovic, and S. Dukanovic: Optimization Algorithms in Function of Binary Character Recognition, Acta Polytechnica Hungarica, 2015, Vol. 12, No. 7, pp. 77–87.

[3] Th. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein: Introduction to Algorithms, MIT Press, Cambridge, Massachusetts, 2009.

[4] K.-L. Du and M. N. S. Swamy: Search and Optimization by Metaheuristics: Techniques and Algorithms Inspired by Nature, Birkhäuser, Cham, Switzerland, 2016.

[5] K. Farkas: Placement Optimization of Reference Sensors for Indoor Tracking, Acta Polytechnica Hungarica, 2015, Vol. 12, No. 2, pp. 123–139.

[6] A. Horák, M. Prýmek, L. Prokop, and S. Mišék: Economic Aspects of Multi-Source Demand-Side Consumption Optimization in the Smart Home Concept, Acta Polytechnica Hungarica, 2015, Vol. 12, No. 7, pp. 89–108.

[7] V. V. Kalashnikov, R. C. Herrera Maldonado, and J.-F. Camacho-Vallejo: A heuristic algorithm solving bilevel toll optimization problem, The International Journal of Logistics Management, 2016, Vol. 27, No. 1, pp. 31–51.

[8] C. Papadimitriou: Computational Complexity, Addison Welsey, Reading, Massachusetts, 1994.

[9] G. E. Renpu: A filled function method for finding a global minimizer of a function of several variables, Mathematical Programming, 1988, Vol. 46, No. 1, pp. 57–67.

[10] Z. Y. Wu, F. S. Bai, Y. J. Yang, and M. Mammadov: A new auxiliary function method for general constrained global optimization, Optimization, 2013, Vol. 62, No. 2, pp. 193–210.

[11] Z. Y. Wu, M. Mammadov, F. S. Bai, and Y. J. Yang: A filled function method for nonlinear equations, Applied Mathematics and Computation, 2007, Vol. 189, No. 2, pp. 1196–1204.

External Disturbances Rejection by Differential Single-Mass Vibratory Gyroscope

Valerii V. Chikovani, Olha A. Sushchenko, Hanna V. Tsiruk

National Aviation University
Kosmonavt Komarov avenue, Kyiv, 03058, Ukraine
v_chikovani@nau.edu.ua, olha_sushch@nau.edu.ua, hanna.tsiruk@nau.edu.ua

Abstract: Vibratory gyroscopes are now most applicable for such intelligent systems as drones, for motion stabilization, robots for accurate positioning of end-effectors, virtual reality systems to change image orientation with turn of a head and many others. During motion these systems can be exposed to mechanical shocks and vibrations. To provide required accuracy, working in such environmental conditions, gyroscopes shall have property of robustness to operating disturbances. This paper proposes differential mode of operation for single-mass vibratory gyroscope as a new operating mode that has higher rejection factors for different external disturbances like shocks and vibrations allowing meeting the requirements of many important applications in intelligent systems. Test results presented in this paper show excellent disturbance rejection properties of differential mode of operation in comparison to well-known rate mode. Despite excellent disturbance rejection results have been obtained for non MEMS gyro, the same results can certainly be obtained for MEMS gyro, too.

Keywords: differential vibratory gyroscope; shock rejection factor; vibration sensitivity

1 Introduction

Coriolis vibratory gyro (CVG) is one of the chronologically latest gyroscopic technology appeared in the world market in the 90s of the previous century. This technology for the sufficiently short time spread out all over the world mainly due to its micro-miniature variant based on micro-electro-mechanical system (MEMS). Vibratory gyros in MEMS implementation have many applications in intellectual systems such as drone for motion stabilization and attitude control, robots for positioning of end-effectors and indoor and outdoor navigation, virtual reality helmet to change image orientation in accordance to the turning of a head and many others including medical and space applications.

There are two well-known modes of CVG operation:

1) Closed loop mode or rate mode [6] [10] [11] [22] where primary stable amplitude standing wave excited in a vibrating structure at one of its resonant frequencies is retained in the vicinity of the drive electrode by the control forces.

Gyro rotation produces Coriolis forces exciting a secondary standing wave that is compensated for by applying the control forces to keep the standing wave at the stable position in the vicinity of the drive electrode. The amplitude of the control signal that compensates for the Coriolis force is proportional to the angle rate.

2) Whole angle or rate-integrating mode [4] [10] [11] [12] when under gyro rotation, Coriolis forces provide transformation of vibration energy from primary to secondary modes and vice versa. Quadrature signal is the only one that is compensated for, to reduce CVG errors [11]. In this case, gyro rotation angle is proportional to standing wave rotation angle caused by Coriolis forces.

There is also an open loop mode of operation, where a standing wave, excited in a resonator, is not controlled by any forces, except the drive force [6]. This mode of operation is rarely used in practice, because of a large bias instability and a low dynamic accuracy [10].

The first, rate, mode of CVG operation is most popular one because of lower influence of manufacturing imperfections, lower noise when measuring small angle rate and it has an acceptable bandwidth for most applications. The second, rate-integrating, mode can have extremely high dynamic range, high bandwidth and very stable scale factor, but it has higher sensitivity to manufacturing imperfections.

Comparatively recent investigations [17] [18] [19] [20], resulted in the third, differential, mode of CVG operation which complements first two modes possessing additional capabilities for suppression of external disturbances. The differential mode of operation can also be implemented in well-known tuning fork design of MEMS or non-MEMS CVGs, because they have two anti-phase vibrating resonators and they can reject disturbances, for example shocks [3] [14] by subtracting signals coming from two resonators, or can survive after very high shocks [13]. In spite of tuning fork and other multi-mass resonator MEMS designs have external disturbance rejection properties they cannot effectively be used in practical applications because their rejection factor is high enough for small shocks and capability to survive does not provide measurements during shock. This is because different resonators have no equal parameters such as $\underline{Q}$ factor, rigidity, resonant frequency and, as a consequence, they have different responses to the same external disturbance.

The proposed differential mode of operation can be implemented in single-mass resonator CVG by keeping a standing wave between the electrodes by applying two stable amplitude control voltages on $\underline{X}$ and $\underline{Y}$ drive electrodes. In this case two magnitudes of angle rates with opposite signs can be picked up from $\underline{X}$ and $\underline{Y}$ sense electrodes. The resulting angular rate can be obtained by subtraction of the two measurement channel signals. Rejection factor in this case is increased because responses of the two ($\underline{X}$ and $\underline{Y}$) channels of the single-mass resonator are much closer to each other, than in case of different mass resonators.

At proper angular alignment of standing wave $\theta^* \neq \underline{m}\pi/4$, $\underline{m}=0, 1\ldots$, cross damping bias component is compensated for. This can be reached at standing wave angular position that equalizes $\underline{X}$ and $\underline{Y}$ measurement channel scale factors $\underline{SF_x}$ and $\underline{SF_y}$ [18].

CVGs defer from other gyroscope technologies by that all practically interesting, which we are numbered here as the first, second and third modes of operation can be implemented in a single triple-mode vibratory gyroscope with automatic switching from one mode to other [21]. It should be noted that dual mode CVG (the first and second ones) have been implemented and described in [7]. The latter gives undeniable advantages of CVGs over competitive technologies, ring laser and fiber optic gyros, in terms of dynamic range, bandwidth, dynamic error in measuring high angle rate, lower noise in measuring small angle rate and reliability [8]. For example, under measuring of small angle rate it is advisable to operate in the rate mode, since the measurement errors are mainly determined by noise and bias drift which can be lower, than that of for rate-integrating mode of operation. Under measuring of high angle rate (more than 500 deg/s) or higher, it is advisable to operate in the second, rate-integrating, mode of operation since the measurement errors are mainly determined by multiplicative error $\Delta\Omega$ caused by scale factor uncertainty, $\Delta\underline{SF}$, $\Delta\Omega=\Delta\underline{SF}*\Omega$. Scale factor for rate-integrating mode of operation is a stable constant (Bryan coefficient). It can reach 35 ppm and its dynamic range is up to 7×10^3 deg/s and more for even low-cost gyros [2].

When gyro is operating under high external disturbances (shocks, vibrations, magnetic fields or others), it is advisable to operate in the third (differential) mode of operation, since this mode of operation for single-mass CVG has higher disturbance rejection factor, than the first and second modes of the same CVG and the third mode for a multi-mass CVG. Switching from one mode to another can be implemented in accordance with changing environmental conditions using intelligent algorithm based on, for example, fuzzy logic.

This paper presents test results showing excellent disturbance rejection properties of differential mode of operation for single-mass CVG in comparison to rate mode under action of external mechanical shocks and vibrations.

2 Differential Mode of Operation

Because rate and rate-integrating modes of CVG operation are well-known [4] [6] [10] [11] [12] [22] let's shortly describe the differential mode of operation for single-mass resonator gyro.

In differential CVG standing wave is located between the electrodes so that wave angle $\theta \neq \underline{m}\pi/4$, $\underline{m}=0,1,2\ldots$, that is standing wave oscillation direction is not coincident with any of electrodes, as depicted, in Figure 1.

In this case CVG output signals in voltages, z_x and z_y, in the differential mode of operation, under nulling quadrature signal, can be written as follows [20]

$$\begin{aligned} -2k\Omega D_y tg2\theta + D_x d_{xx} + d_{xy} D_y tg2\theta = z_x \\ 2k\Omega D_x ctg2\theta + D_y d_{yy} + d_{xy} D_x ctg2\theta = z_y \end{aligned} \tag{1}$$

where,

$$d_{xx} = \left[\frac{2}{\tau} + h\cos 2(\theta - \theta_\tau)\right];\ h = \Delta\left(\frac{1}{\tau}\right) = \frac{1}{\tau_1} - \frac{1}{\tau_2};$$
$$\frac{2}{\tau} = \frac{1}{\tau_1} + \frac{1}{\tau_2};\ d_{yy} = \left[\frac{2}{\tau} - h\cos 2(\theta - \theta_\tau)\right];$$
$$d_{xy} = h\sin 2(\theta - \theta_\tau);$$

D_x and D_y are transformation coefficients of resonator deformations into voltage for X and Y electrodes, respectively; τ_1, τ_2 are time constants along resonator damping pricipal axis, $\tau_1=\tau_{\min}$ which is located under angle θ_τ relative to direction of standing wave oscilation and $\tau_2=\tau_{\max}$ which is located under angle 45 deg to the direction of $\tau_{\min}$ axis, $\tau_{\max}$ axis not shown in figure 1; *k* is Bryan coefficient.

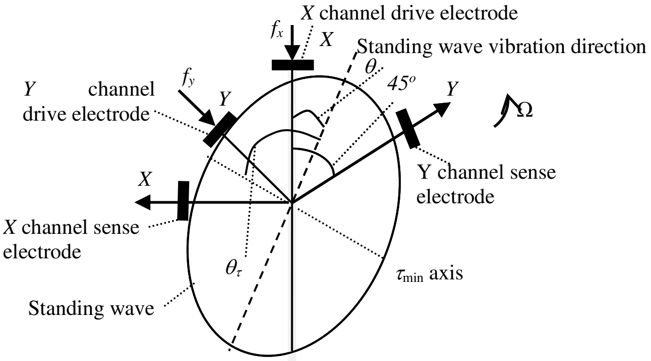

X channel drive electrode
f_x
X
Standing wave vibration direction
θ
45°
Y
Ω
Y channel drive electrode
f_y
Y
X
Y channel sense electrode
X channel sense electrode
θ_τ
$\tau_{\min}$ axis
Standing wave

Figure 1

Standing wave position under angle $\theta \neq m\pi/4$, m=0,1,2…, in differential mode of CVG operation

As can be seen from equations (1) there are two X and Y measurement channels, with negative -Ω and positive Ω angle rates, respectively. Thus, control system that retains the standing wave between electrodes implements differential mode of operation for single-mass resonator CVG. As can also be seen from (1), X and Y channels scale factors SF_x, SF_y and biases B_x, B_y are dependent on angle θ as follows

$$\begin{aligned} SF_x &= 2kD_y \tan 2\theta \qquad & B_x &= D_x d_{xx} + d_{xy} D_y \tan 2\theta \\ SF_y &= 2kD_x \cot 2\theta \qquad & B_y &= D_y d_{yy} + d_{xy} D_x \cot 2\theta \end{aligned} \tag{2}$$

One can choose angle θ, from application point of view, such that $\underline{B_x}=\underline{B_y}$, or $\underline{SF_x}=\underline{SF_y}$, both of these angles are different and close to 22.5 deg [21]. To effectively implement differential mode of operation it is advisable to align standing wave under angle θ* at which $\underline{SF_x}=\underline{SF_y}$, so using (2) the following relationship can be written down

$$D_y \tan 2\theta^* = D_x \cot 2\theta^* \text{ or } \theta^* = \frac{1}{2}\arctan\sqrt{\frac{D_x}{D_y}} = \frac{1}{2}\arctan\sqrt{\frac{SF_y}{SF_x}} \tag{3}$$

When standing wave angle is θ*, half sum and half difference of the $\underline{X}$ and $\underline{Y}$ measurement channels can be represented as follows [19]

$$\begin{aligned} &(z_x - z_y)/2 = -SF_d\Omega + D_y d_{yy} - D_x d_{xx} \\ &(z_y + z_x)/2 = D_y d_{yy} + D_x d_{xx} + 2d_{xy}\sqrt{D_x D_y} \\ &SF_d = 2k\sqrt{D_x D_y} \end{aligned} \tag{4}$$

where, $\underline{SF_d}$ is a scale factor of differential CVG, when θ=θ*. As can be seen from (4) difference signal of $\underline{X}$ and $\underline{Y}$ measurement channels has no damping cross coupling term $\underline{d_{xy}}=\underline{h\sin}2(\theta-\theta_\tau)$, and sum of these channels has no angle rate, but contains current information about main bias components that can supposedly be used for their on-line estimation. In case when angle θ≠θ*, and θ≠$\underline{m}$π/4, $\underline{m}$=0, 1, 2,…, then difference and sum of the channels have different scale factors, $\underline{SF_{x-y}}=\underline{SF_x}+\underline{SF_y}$ and $\underline{SF_{x+y}}=\underline{SF_x}-\underline{SF_y}$ [22].

Differential CVG control system block diagram operating in differential mode of operation under command signal $\theta_{\text{comm}}=-\theta^*$ is presented in Figure 2. It should be noted that when $\theta_{comm}=0$ it operates in the rate mode and when connection of θ_{real} signal to proportional and integral (PI) controller is open, it operates in rate-integrating mode [21].

In the differential mode of operation two drive signals $\underline{X_{in}}$ and $\underline{Y_{in}}$ excite standing wave so that it is located between the electrodes at an angle θ=θ*. Frequency tracking subsystem presented in Figure 2 is based on phase lock loop (PLL) implemented in digital form and consists of phase detector, loop filter and voltage controlled oscillator [1] (numerically controlled oscillator in digital case). CVG uses PI controllers to compensate for Coriolis force and quadrature signals.

Figure 3 demonstrates $\underline{X}$, $\underline{Y}$, ($\underline{X}$-$\underline{Y}$)/2 and ($\underline{X}$+$\underline{Y}$)/2 signals under measuring of constant angle rates ±30 deg/s after each channel bias subtraction. For the gyro being under test angle θ*=25.067 deg. Figure 3 shows absence of angle rate in the ($\underline{X}$+$\underline{Y}$)/2 channel that can help one to see current measurement error components.

The next sections present test results that allow us to quantitatively determine disturbance rejection factors of differential mode of operation for single-mass ring type resonator CVG in comparison to rate mode of the same CVG, when disturbances are external mechanical shocks and vibrations.

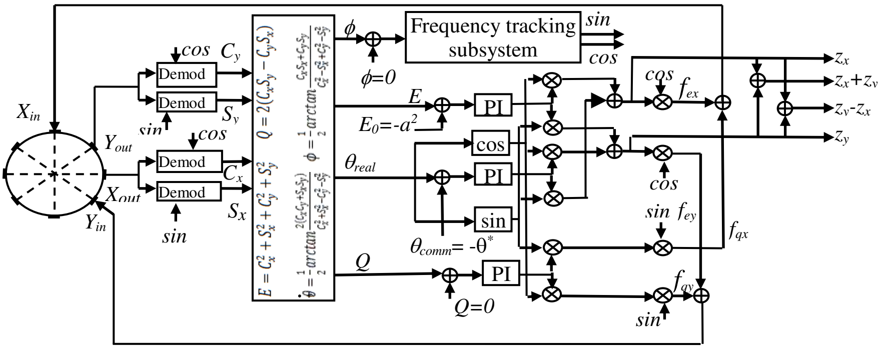


Figure 2

CVG control system block diagram operating in differential mode of operation

It should be noted that in all tests dampers, screens and other disturbance protection means did not use, so all rejection factors and external disturbance sensitivities have been determined for exposed differential CVG.

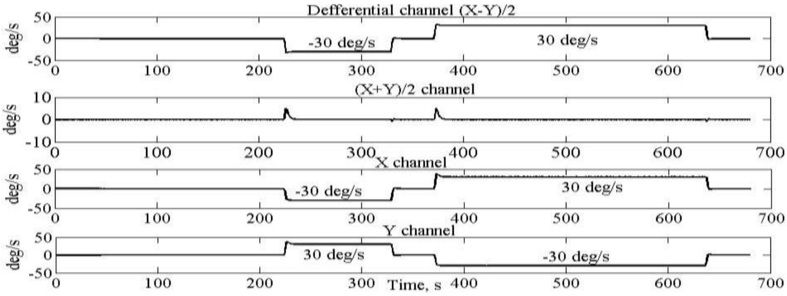


Figure 3

Differential CVG Output signals

3 Shock Rejection Factor

Since each of the two X and Y measurement channels operate, in essence, in the rate mode and their difference presents differential mode signal, then ratio of one of the X and Y channel signals to half difference one, (X-Y)/2, will determine disturbance rejection factor R_s of CVG rate mode as compared with differential one. Ratio will be determined after bias subtraction for each of the channels. Thus, rejection factor for external mechanical shock will be calculated as follows

$$R_s = \min\left\{\frac{mean(abs(X_i))}{mean\left[abs\left((X_i - Y_i)/2\right)\right]}, \frac{mean(abs(Y_i))}{mean\left[abs\left((X_i - Y_i)/2\right)\right]}\right\}, i = 1...5 \quad (5)$$

where, X_i and Y_i are peak values of X and Y channel responses for an i^{th} shock.

Bias sensitivity coefficient $\underline{S_g}$ to shock or vibration acceleration $\underline{A_g}$ will be calculated as a ratio

$$S_g = mean\left[abs\left(X_i - Y_i/2\right)\right]/A_g\ ,\ i = 1...5 \tag{6}$$

3.1 Shocks along Input Axis

Figure 4 shows superposed $\underline{X}$ and $\underline{Y}$ channel signals of differential CVG after exposure to low amplitude mechanical shock (less, than 5 $\underline{g}$, where g is free fall acceleration) along gyro input axis (IA). As can be seen from figure 4 $\underline{X}$ and $\underline{Y}$ channel responses are almost equal to each other, so differential channel response is close to zero. The latter means that rejection factor to low shock is very high, as has also been demonstrated in [15] for tuning fork MEMS gyro. There is not a noticeable angle rate that usually presents in high $\underline{g}$ shocks and, as a consequence, peak values almost coincide. The same result can be obtained for low amplitude lateral shock (perpendicular to IA).

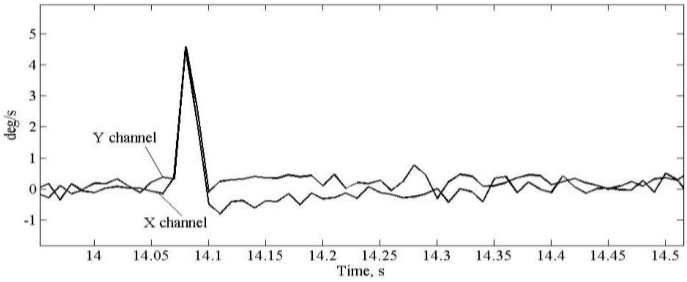


Figure 4

$\underline{X}$ and $\underline{Y}$ channels superposed responses to small shock

When shock increases to 20 g of 2 ms duration, rejection factor decreases and appears angle rate that goes with linear acceleration during high shock. Figure 5 shows responses of three channels ($\underline{X}$, $\underline{Y}$ and differential ones) of differential CVG under 5 shocks along IA. As can be seen from the peak response values indicated on the figure 5 rejection factor, calculated by (5), is close to $\underline{R_{sp}}$=2.

Let's analyze response to shock in detail. Figure 6 shows four signals responses ($\underline{X}$, $\underline{Y}$, ($\underline{X}$ – $\underline{Y}$)/2 and ($\underline{X}$ + $\underline{Y}$)/2) of differential CVG to the first of five shocks along gyro IA. Due to presence of angle rate during shock, $\underline{X}$ and $\underline{Y}$ channels have different amplitudes.

To the equal error signals caused by shock, angle rate is added to $\underline{X}$ channel signal and it is subtracted from $\underline{Y}$ channel signal. The difference channel, ($\underline{X}$ -$\underline{Y}$)/2, presents angle rate acting during shock, and fourth signal ($\underline{X}$+$\underline{Y}$)/2, in accordance with (4) does not contain angle rate and presents error signal caused by sensor deformation during shock.

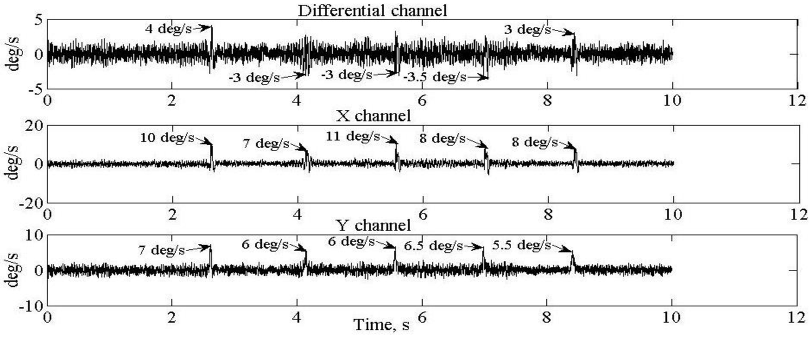


Figure 5

Responses to five shocks of 20$\underline{g}$ amplitude and 2 ms duration along IA

From Figure 6 one can see that peak value of error signal is about 2 times greater, than that of angle rate. Therefore, if we integrate $\underline{X}$ or $\underline{Y}$ signal to calculate angle error accumulated during shock, it is very important parameter for gyro application in stabilization system operating in harsh environment, then much higher angle error can be obtained in comparison with integration of differential signal.

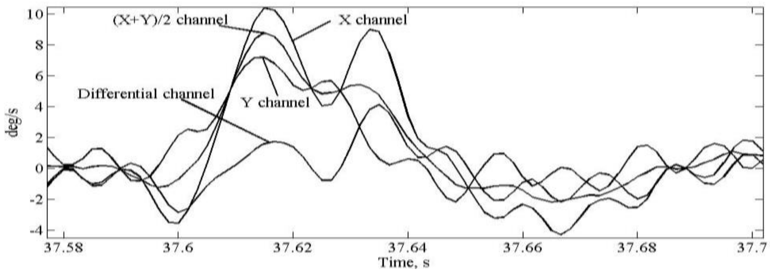


Figure 6

Four signal responses to first of five 20g shocks along IA

Figure 7 shows angle errors obtained by integration of $\underline{X}$, $\underline{Y}$ and differential channel signals during 5 shocks along IA. Differential channel angle error is about 5 times less, than that of $\underline{Y}$ rate channel, and is no more than 2 arc min. Thus, shock rejection factor in terms of angle error for differential CVG increases to $\underline{R_{sa}}$=5 in comparison with rate gyro for 20$\underline{g}$ shock along IA.

Important error component arising under shock is a bias change before and after shock. Figure 8 shows change in biases after each of five of 20 *g* shocks along gyro IA. Shock rejection factor in terms of bias change can be calculated using (5), where $\underline{X_i}$ and $\underline{Y_i}$ are change of biases of corresponding channels before and after $\underline{i}$-th shock. Average over 5 shocks change of absolute value of biases is 0.0097 deg/s for differential channel and for minimum of $\underline{X}$ and $\underline{Y}$ channels it is 0.03 deg/s, hence, $\underline{R_{sb}}$=0.03/0.0097≈3.

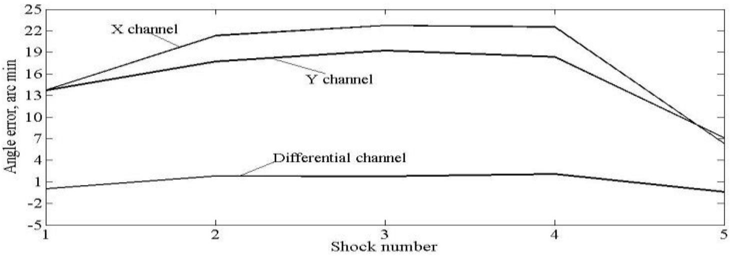


Figure 7

Angle errors during five 20g shocks along IA

Using the data presented in Figure 8 bias sensitivity to shock acceleration can also be calculated using (6). Bias sensitivity to shock acceleration for differential channel is about $\underline{S_{dg}}$=4.8*10^{-4} deg/s/g, and for minimum of the two ($\underline{X}$ and $\underline{Y}$ channels) it is about $\underline{S_{yg}}$ =1.5*10^{-3} deg/s/g. Thus, the sensitivity to shock acceleration acting along gyro IA for differential CVG is about 3 times less, than that of rate one.

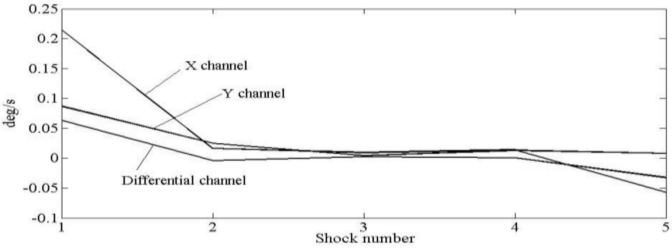


Figure 8

Change of biases after 20 g shocks along IA

It is expected that higher shocks of 100 g amplitude and 2 ms duration, will affect differential CVG channel responses more significant. Figure 9 shows superposed all four channel responses to the first of five of 100 g shocks along IA. As can be seen $\underline{X}$ channel response is out of measurement range (signal saturation occurs). The same results have been obtained for all 4 successive shocks. Half sum channel signal shows, that sensor deformation caused by 100 g shock results in equivalent wrong angle rate of more than 100 deg/s. Differential channel shows that angle rate during shock is more than 50 deg/s versus 4 deg/s for 20 g shock. Taking into account that $\underline{Y}$ channel signal amplitude is significantly lower than that of the $\underline{X}$ channel, one can conclude that asymmetry in sensor design and its attachment to the gyro casing resulted in that shock load deformed resonator region close to the $\underline{X}$ sense electrode significantly greater than that of the $\underline{Y}$ electrode. Despite of that designers are trying to design sensor with maximum symmetry (ring, hemisphere and cylinder), residual asymmetry of low-cost sensors remain too great to meet required measurement accuracy during high shock. This problem can be resolved by using dampers or by improving sensor design asymmetry remaining in low-cost category. It is not reasonable to calculate shock rejection factor $\underline{R_s}$ in terms of peak values for 100 g shocks along IA, because of $\underline{X}$ channel signal saturation.

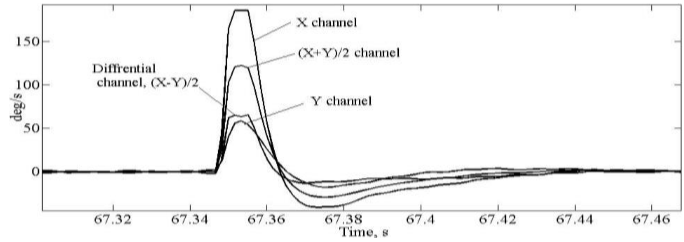


Figure 9

Four signal responses to the first of five 100g shocks along IA

Figure 10 shows angle errors of X, Y and differential channels during each of 5 shocks. Differential channel angle errors for all 5 shocks are almost constant at the level of 5 arc min. It is obvious that some part of these errors are due to X channel signal saturation. Nevertheless, shock rejection factor in terms of angle error can be determined at the level of R_{sa}≈5.

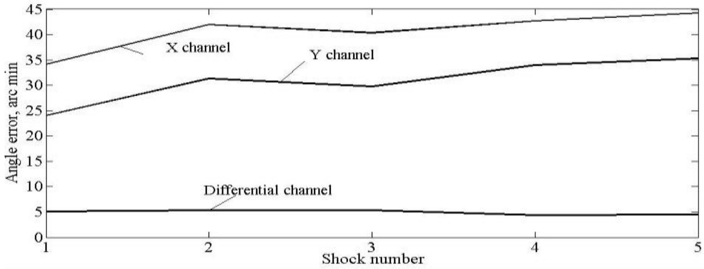


Figure 10

Angle errors during 100g shocks along IA

Shock rejection factor in terms of bias change and differential CVG bias sensitivity to shock acceleration along IA can be calculated, according to (5) and (6), respectively, from the data presented in Figure 11.

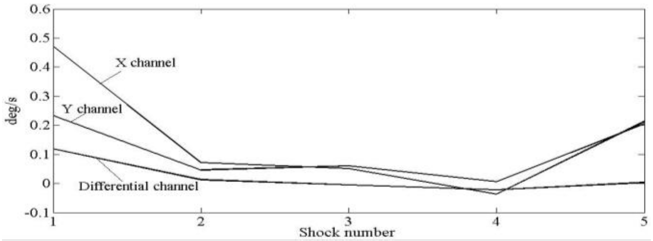


Figure 11

Bias change after five 100g shocks along IA

Average over 5 shocks absolute value of bias change for X channel is 0.171 deg/s, for Y channel it is 0.111 deg/s and for differential channel it is 0.033 deg/s, hence, R_{sb}=0.111/0.033≈3.4. Differential CVG bias sensitivity to shock acceleration along IA is 0.033/100=3.3*10^{-4} deg/s/g. The same parameter under 20 g shock, obtained above, is 4.8*10^{-4} deg/s/g, this difference is supposedly due to $\underline{X}$ channel signal saturation under 100 $\underline{g}$ shocks.

3.2 Lateral Shocks

Figure 12 shows $\underline{X}$, $\underline{Y}$ and differential channel responses to 5 lateral shocks of 20 g amplitude and 2 ms duration. Calculation of rejection factor in terms of peak value over 5 shock using (5) results in $\underline{R_{sp}}\approx 2$. Thus, rejection factors for lateral and along IA 20 $\underline{g}$ amplitude shocks are equal to each other.

Almost the same behavior of angle error for differential channel is observed during lateral shocks of 20 $\underline{g}$ amplitude, shown in Figure 13, but in this case shock rejection factor in terms of angle error for differential mode of operation increases to $\underline{R_{sa}}=6$ in comparison with rate mode, with no more, than 2 arc min angle error. Real gyro turn angle after shock is zero. This turn angle is monitored by an optical method.

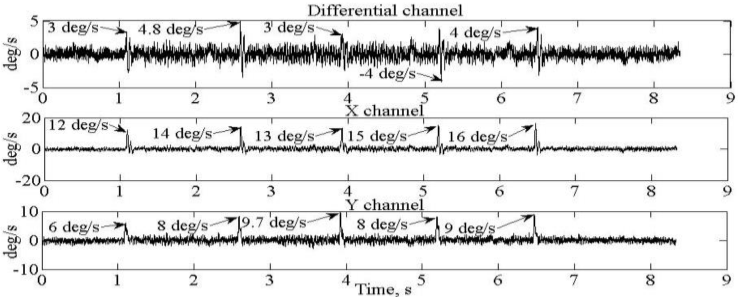


Figure 12

Responses to five lateral shocks of 20 g amplitude

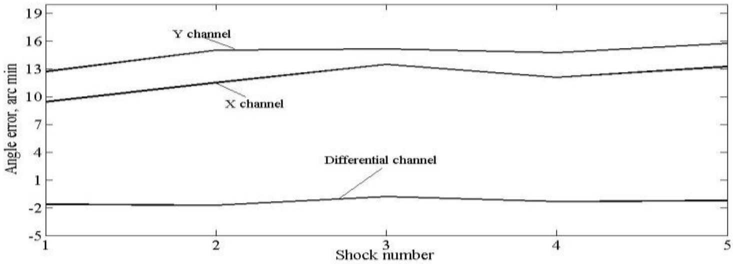


Figure 13

Angle errors during 20 g lateral shocks

Figure 14 shows change of biases after lateral shocks. In this case rejection factor in terms of bias change is greater than that of along IA and evaluated, using data presented in figure 14, as $\underline{R_{sb}}\approx 8$. Differential channel bias sensitivity to lateral shock acceleration is about $\underline{S_{gd}}=10^{-3}$ deg/s/g versus $\underline{Y}$ channel sensitivity which is evaluated as $\underline{S_{gy}}$ =8*10^{-3} deg/s/g.

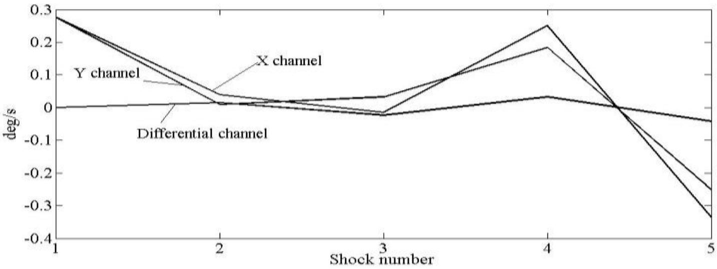


Figure 14

Bias change after 20 g lateral shocks

Figure 15 demonstrates superposed all four channels of differential CVG responses to the first of five of 100 g lateral shocks. Error signal peak value presented by (X+Y)/2 channel is about 3.5 time higher, than angle rate signal presented by differential channel.

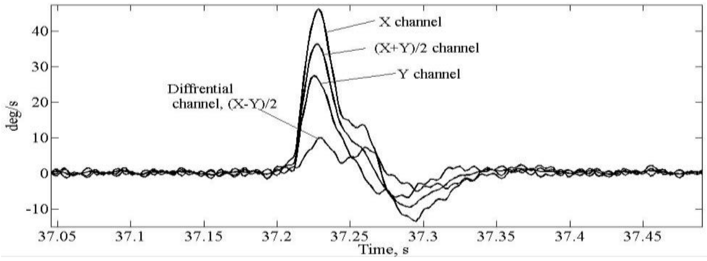


Figure 15

Four signal responses to the first of five 100 g lateral shocks

It should be noted that in this case there is no saturation in the measured signals. Figure 16 demonstrates responses to five lateral shocks of 100 g amplitude for

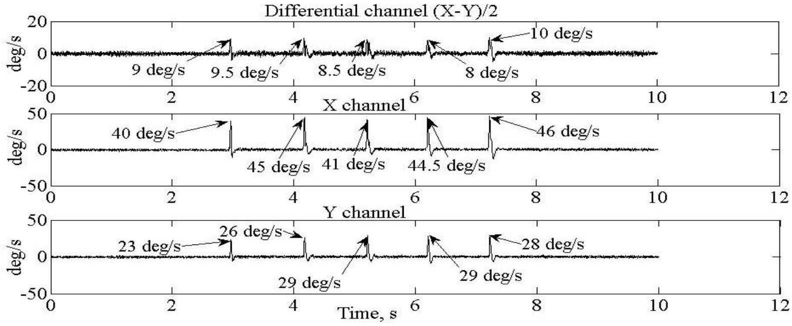


Figure 16

Responses to five 100 g lateral shocks

each of three differential CVG signals with indication of peak values for each of them. X and Y channel peak values do not always coincide with the differential channel peak because peak values do not always coincide in time. Calculation using (5) yields rejection factor $\underline{R_{sp}}\approx3$ for lateral 100 $\underline{g}$ shocks.

Figure 17 shows angle errors of $\underline{X}$, $\underline{Y}$ and differential channels during each of 5 lateral shocks. Differential channel maximum angle error reaches 18 arc min, and rejection factor for lateral shock of 100$\underline{g}$ amplitude is $\underline{R_{sa}}\approx2.5$.

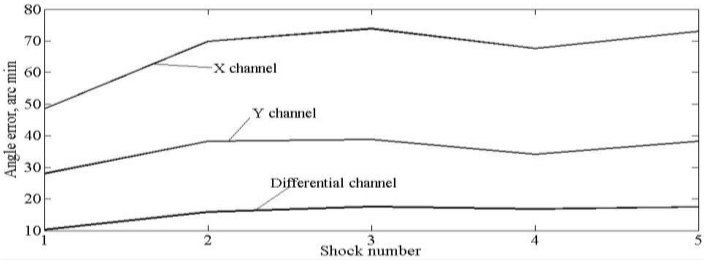


Figure 17

Angle errors during five 100 g lateral shocks

Figure 18 presents bias change after each of five lateral shocks of 100 $\underline{g}$ amplitude and 2 ms duration. Shock rejection factor in terms of bias change and differential CVG bias sensitivity to lateral shock acceleration yield $\underline{R_{sb}}\approx4$ and $3*10^{-4}$ deg/s/g, respectively.

Let's summarize shock tests in the Table 1. Table 1 data show that differential CVG shock rejection factor is minimum, 2 times greater, than that of rate CVG in terms of peak values. In terms of angle error shock rejection factor for differential CVG is minimum, 2.5 times greater for 100 $\underline{g}$ lateral shocks and is 6 times greater for 20 $\underline{g}$ shocks, than that of rate CVG.

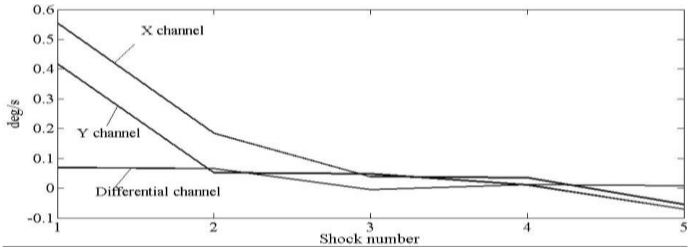


Figure 18

Bias change after five 100g lateral shocks

As to shocks along IA rejection factor in terms of angle error is 5 times greater for both 20 $\underline{g}$ and 100 $\underline{g}$ shocks, than that of for rate CVG. Shock rejection factor in terms of bias change before and after shocks is minimum, 3 times greater for both 20 $\underline{g}$ and 100 $\underline{g}$ shocks along IA and 4 times greater for lateral shocks, than that of for rate CVG.

Table 1

Summarized data on shock test results

Rejection factor in terms of peak value R_p		Rejection factor in terms of angle error R_a max error (arc min)		Rejection factor in terms of bias change R_b max bias change (deg/s)		Diff. CVG bias sensitivity to shock acceleration, deg/s/g	
Shock amplitude 20 *g*, 2 ms duration							
Along IA	Lateral	Along IA	Lateral	Along IA	Lateral	Along IA	Lateral
2	2	5 2	6 2	3 0.06	8 0.03	$4.8*10^{-4}$	10^{-3}
Shock amplitude 100 *g*, 2 ms duration							
saturation	3	5 5	2.5 18	3.4 0.12	4 0.07	$3.3*10^{-4}$	$3*10^{-4}$

4 Vibration Sensitivity

Gyroscopes in most applications operate on moving vehicles and they are subjected to vibrations during motion and at stops when engine is running. Gyros change their biases and other parameters under vibration. The main reason of this is also the design asymmetry discussed in the previous section. In many low-cost gyros bias change under vehicle vibration reduces measurement accuracy much greater, than in absence of vibration. Gyro sensitivity to linear vibration expressed in *g* units is no less important parameter from practical point of view, than bias stability in absence of vibration which value one can see in any gyro data sheet.

4.1 Vibration along IA

4.1.1 *g*-Dependent Bias

Figure 19 shows *X*, *Y* and differential channel biases change versus amplitude of sinusoidal vibration for 50, 100 and 300 Hz vibration frequencies. Bias change is calculated as difference between corresponding channel bias obtained at vibration and the same channel bias at no vibration. As can be seen from this figure the higher the frequency of vibration, the greater the change of the bias, especially for higher vibration amplitude. For 3 *g* vibration amplitude and 300 Hz frequency change of biases for *X* and *Y* channels are almost equal to each other at a value of 1.2 deg/s and for differential channel change of the bias is about 0.1 deg/s. So, change of the bias for differential channel 12 times less than that of for rate channels. As can visually be estimated from Figure 19 at any frequency and amplitude of vibration, in the considered here ranges, the bias change of differential channel is less than that of the *X* and *Y* channels of about 10 times.

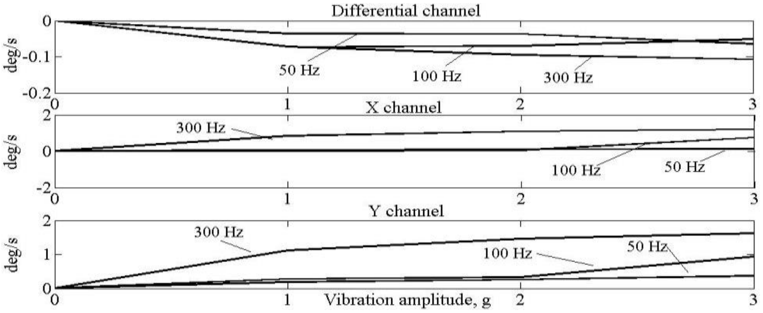


Figure 19

Bias change under sinusoidal vibration of different frequencies

Figure 20 shows bias sensitivities of $\underline{X}$, $\underline{Y}$ and differential channels to vibration amplitude versus frequency. These sensitivities have been calculated as a tangent of tilt angles of least squares straight line drawn by the data presented in Figure 19. As a result the bias sensitivities to vibration amplitude in deg/s/g within the range of [1 3] g for each of three vibration frequencies are obtained and graphed in Figure 20.

One can see from this graph that bias sensitivity to vibration amplitude is dependent on frequency. So, it will be difficult to calibrate this parameter using accelerometer measurement data, as it is often made in low-cost inertial measurement units [18], [13] and has been noted in [9]. Differential channel sensitivity to vibration at 300Hz is $\underline{S_{dg,300}}$= 0.034 deg/s/g and for $\underline{X}$ and $\underline{Y}$ channels they are $\underline{S_{xg,300}}$= 0.38 deg/s/g and $\underline{S_{yg,300}}$= 0.52 deg/s/g, respectively. It is about an order of magnitude greater, than for differential channel. Besides, using data presented in Figure 20, bias sensitivity to vibration amplitude and frequency can be calculated by the same technique with the aid of least squares straight line.

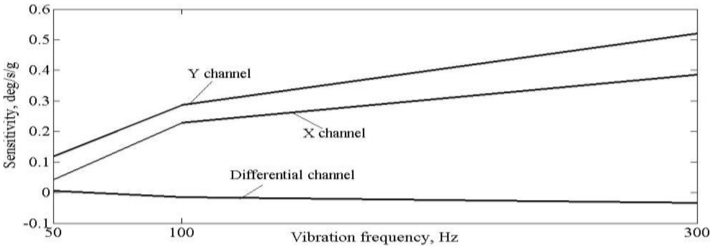


Figure 20

g-dependent bias sensitivity to vibration frequency

For differential CVG bias sensitivity to vibration amplitude and frequency is $\underline{S_{d,g,f}}$=1.3*10^{-4}deg/s/g/Hz. In the frequency range far from CVG resonant frequency it is reasonable to suppose that parameter $\underline{S_{d,g}}$ will be changed almost linearly versus frequency, as it is in the range of up to 300 Hz, hence, one can make a linear prediction of differential CVG bias change.

4.1.2 *g*-Dependent Noise

Figure 21 presents differential (left), $\underline{X}$ and $\underline{Y}$ (right) channels root of Allan variances for different vibration amplitudes at frequency 100 Hz. Figure 21 demonstrates that such noise components as white noise and random walk almost do not depend on vibration amplitude, whereas bias instability and rate random walk are g-dependable. Because bias instability is the most important gyro parameter we will focus on g-sensitivity of this noise component, but firstly let's discuss relationship between RMS values of $\underline{X}$, $\underline{Y}$ and differential channels total noises.

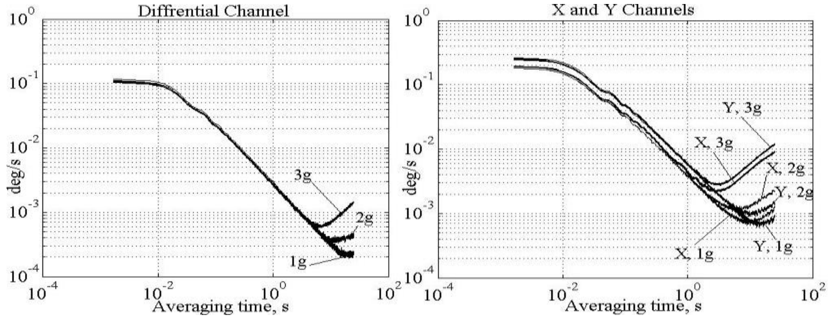


Figure 21

Differential CVG signals root of Allan variances at vibration frequency 100 Hz

From Figure 21 one can see that all noise components (white noise, random walk, bias instability and rate random walk) of differential channel are appreciably less than those of for $\underline{X}$ and $\underline{Y}$ channels. The same results are obtained for 50 Hz and 300 Hz vibration frequencies. This means that total noise RMS value for differential channel signal is less, than that of for $\underline{X}$ and $\underline{Y}$ channels. There is well known relationship between standard deviations of the considered channels that follows from the expression $d=(\underline{X}-\underline{Y})/2$, where d is differential channel signal

$$\sigma_d = \frac{1}{2}\sqrt{\sigma_x^2+\sigma_y^2-2\mathrm{cov}(X,Y)} \tag{7}$$

Where, σ_d , σ_x, σ_y are RMS values of noises for differential, $\underline{X}$ and $\underline{Y}$ channels, respectively; cov($\underline{X}$, $\underline{Y}$) is $\underline{X}$ and $\underline{Y}$ inter-channel covariance. If inter-channel covariance is positive and $\sigma_x \approx \sigma_y$, then σ_d is less than σ_x and σ_y. Inter-channel covariance in deferential CVG is dependent on standing wave drive technique and its value changes versus external disturbances. Figure 22 shows inter-channel correlation coefficients (covariance normalized by σ_x and σ_y) when there is absent and present external vibration with different parameters. All values of correlation coefficients are positive and sufficiently large, as a consequence, σ_d is less than minimum of σ_x and σ_y up to 2.8 times.

Let's now determine g-sensitivity of bias instability. Figure 23 shows dependence of bias instability versus vibration amplitude. Sensitivity of bias instability to vibration amplitude is calculated as a tangent of tilt angles of least squares straight

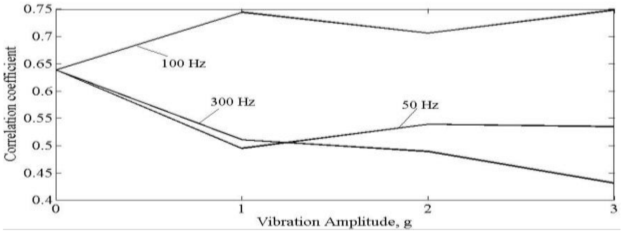


Figure 22

Inter-channel correlation coefficients for different vibration parameters

line drawn by the data presented in Figure 23 for each of three vibration frequencies. Calculation results are presented in the Table 2.

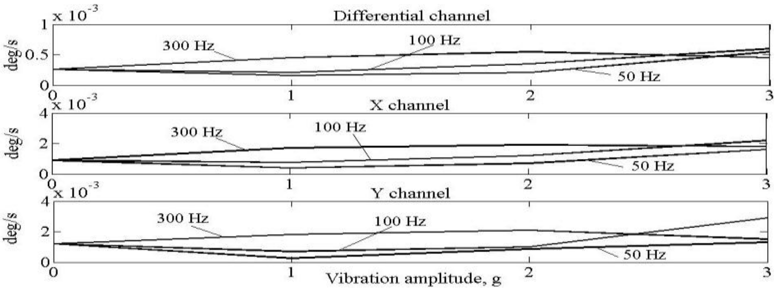


Figure 23

Bias instability for different vibration parameters

Table 2

g-sensitivity of bias instability during vibration along IA

Vibration frequency, Hz	Differential channel bias g-sensitivity, deg/s/g	X channel bias g-sensitivity, deg/s/g	Y channel bias g-sensitivity, deg/s/g
50	$9.2*10^{-5}$	$2.4*10^{-4}$	$9*10^{-5}$
100	$1.16*10^{-4}$	$4.35*10^{-4}$	$5.4*10^{-4}$
300	$6.7*10^{-5}$	$2.9*10^{-4}$	10^{-4}
Mean	$9.2*10^{-5}$	$3.2*10^{-4}$	$2.4*10^{-4}$

Mean value of bias g-sensitivity for differential channel is 2.6 times less than a minimum of that for $\underline{X}$ and $\underline{Y}$ rate channels.

4.2 Lateral Vibration

The behavior of noise components during lateral vibration in the same ranges of vibration parameters is similar to that of along IA. So, we present lateral vibration tests in the resultant Table 3. Differential channel g-sensitivity mean value of bias instability is 1.7 times less than the minimum of that for $\underline{X}$ and $\underline{Y}$ rate channels. Inter-channel correlation coefficient for lateral vibration is also positive and it is in the range of 0.43-0.75.

Table 3

g-sensitivity of bias instability during lateral vibration

Vibration frequency, Hz	Differential channel bias *g*-sensitivity, deg/s/g	*X* channel bias *g*-sensitivity, deg/s/g	*Y* channel bias *g*-sensitivity, deg/s/g
50	$8.5*10^{-5}$	$3.75*10^{-4}$	$2.95*10^{-4}$
100	$5.4*10^{-4}$	$1.75*10^{-3}$	$9.3*10^{-4}$
300	$5.25*10^{-4}$	$2.04*10^{-3}$	$7.1*10^{-4}$
Mean	$3.8*10^{-4}$	$1.4*10^{-3}$	$6.5*10^{-4}$

For these vibration parameters differential channel RMS value σ_d of total noise is 2.2 times less than minimum of σ_x and σ_y.

Conclusion

Differential CVG can be considered as a third mode of operation for a vibratory gyro. This mode of operation can be called an intelligent robustness enhancement to external disturbances one. A triple-mode gyro can be implemented for both MEMS and non-MEMS vibratory gyros using an intelligent algorithm for choosing a corresponding mode of operation in accordance with changing environmental conditions. In addition, it was found that the bias sensitivity to the amplitude of external vibrations is dependent on vibration frequency. It was also found that differential single mass CVG output noise is less than that of the rate CVG for both present and absent of external vibrations. The results obtained in this paper can be extended to MEMS gyros and have many important applications in intelligent systems.

References

[1] Alireza Namadmalan, Javad Shokrollahi Moghani: New Resonant Inverter Tuning for Three-Phase Current Source Parallel Resonant Inverters, Acta Polytechnica Hungarica, v.11, #5, pp. 217-234, 2014

[2] A. Jeanroy, P. Featonby, J-M. Caron: Low-Cost Miniature and Accurate Sensors for Tactical Applications, 10-th S. Petersburg Int. Conf. on Integrated Navigation Systems, pp. 286-293, May, 2003

[3] A. A. Trusov, A. R. Schofield, A. M. Shkel: Micromachined tuning fork gyroscopes with ultra-high sensitivity and shock rejection, US Patent #8322213, publ. date 4 Dec. 2011

[4] B. Gallacher, Zh. Hu, S. Bowles: Full control and compensation scheme for a rate-integrating MEMS gyroscope, 13th Int. Conf. on Dynamical Systems - Theory And Applications, Lodz, Poland, paper id: ENG267, Dec. 7-10, 2015

[5] Chen Fan, Xiaoping Hu, Xiaofeng He, Kanghua Tang, Bing Luo: Observability Analysis of a MEMS INS/GPS Integration System with Gyroscope G-Sensitivity Errors, *Sensors*, *14*, pp. 16003-16016, Sept., 2014. doi:10.3390/s140916003

[6] D. D. Lynch: Coriolis Vibratory Gyroscope, IEEE Standard Specification Format Guide and Test Procedure for Coriolis Vibratory Gyros, IEEE std.1431^{TM}, Annex B, pp. 56-66, Dec. 2004

[7] D. D. Lynch, A. Matthews: Dual Mode Hemispherical Resonator Gyro Operating Characteristics, 3-rd S. Petersburg Int. Conf. on Integrated Navigation Systems, part 1, pp. 37-44, May 1996

[8] David M. Rozelle: Hemispherical Resonator Gyro: From Wineglass to the Planets, http://www.northropgrumman.com/capabilities/hrg/documents/hrg.pdf

[9] H. Weinberg: Gyro Mechanical Performance: The Most Important Parameter, Analog Devices Inc., Technical article MS-2158, pp. 1-5, Sept. 2011, www.analog.com

[10] J. A. Gregory: Characterization Control and Compensation of MEMS Rate and Rate-Integrating Gyroscopes".- Ph.D. Dissertation, Michigan University, P.198, 2012

[11] J. Y. Cho: High-Performance Micromachined Vibratory Rate- And Rate-Integrating Gyroscopes, Ph.D. Dissertation, Michigan University, P.293, 2012

[12] J-K. Woo, J. Y. Cho, Ch. Boyd, Kh. Najafi: Whole-Angle-Mode Micromachined Fused-Silica Birdbath Resonator Gyroscope (Wa-Brg), IEEE MEMS Conf., San Francisco, CA, USA, January 26-30, 2014

[13] J. B. Bancroft and G. Lachapelle: Estimating MEMS Gyroscope G-Sensitivity Errors in Foot Mounted Navigation, 2nd Int. Conf. on Ubiquitous Positioning, Indoor Navigation and Location-Based Service, Helsinki, Finland, pp. 1-6, 2-5 Oct 2012

[14] P. Soobramaney: Mitigation of the Effects of High Levels of High-Frequency Noise on MEMS Gyroscopes, Ph.D. Dissertation, Auburn University, Alabama, p. 200, 3 Aug. 2013

[15] R. Schofield, A. A. Trusov, A. M. Shkel: Multi-Degree of Freedom Tuning Fork Gyroscope Demonstrating Shock Rejection, IEEE Conf. on Sensors, Atlanta, Georgia, USA, pp. 120-123, 28-31 Oct. 2007

[16] S. Dellea, F. Giacci, P. Rey, A. Capodici, G. Langfelder: Reliability of gyroscopes based on piezoresistive nano-gauges against shock and free-drop tests, 29th IEEE Int. Conf. on Micro Electro Mechanical Systems (MEMS), Shanghai, China, pp. 255-258, 24-28 Jan. 2016

[17] V. V. Chikovani, E. O. Umakhanov, P. I. Marusyk: The compensated differential CVG, Gyro Technology Symposium, Germany, Karlsruhe university, pp. 3.1-3.8, 16-17 Sept. 2008

[18] V. V. Chikovani: Method of angle rate measurement with Coriolis vibratory gyroscope, UA Pat. 95709 Ukraine, Icl. G01 C 19/02, Publ. date 25 Aug. 2011 (bul. # 16/2011 in Ukrainian)

[19] V. V. Chikovani, O. A. Suschenko: Differential mode of operation for ring-like resonator CVG, IEEE Proc. Intern. Conf. on Electronics and Nanotechnology, Kyiv, Ukraine, pp. 451-455, 15-18 April 2014

[20] V. V. Chikovani, G. V. Tsiruk: Bias Compensation in Differential Coriolis Vibratory Gyro, Electronics and control systems, NAU, Kyiv, Ukraine, #4 (38), pp. 99-103, 2013

[21] V. V. Chikovani, H. V. Tsiruk: Differential Mode of Operation For Multimode Vibratory Gyroscope, IEEE Proc. Intern. Conf. on Actual Problem of Unmanned Aerial vehicles Development (APUAVD), NAU, Kyiv, Ukraine, pp. 87-90, Oct. 13-15, 2015

[22] Zhong Su, Ning Liu, Qing Li, Mengyin Fu, Hong Liu, Junfang Fan: Research on the Signal Process of a Bell-Shaped Vibratory Angular Rate Gyro, Sensors, 14, 5254-5277, 2014; doi: 10.3390/s140305254