Toward Human Level Machine Intelligence – Is it Achievable? The Need for a Paradigm Shift

Lotfi A. Zadeh

Department of EECS, University of California,
Berkeley, CA 94720-1776; Telephone: 510-642-4959; Fax: 510-642-1712;
e-mail: zadeh@eecs.berkeley.edu

Abstract Officially, AI was born in 1956. Since then, very impressive progress has been made in many areas – but not in the realm of human level machine intelligence. Anyone who has been forced to use a dumb automated customer service system will readily agree. The Turing Test lies far beyond. Today, no machine can pass the Turing Test and none is likely to do so in the foreseeable future.

During much of its early history, AI was rife with exaggerated expectations. A headline in an article published in the late forties of last century was headlined, "Electric brain capable of translating foreign languages is being built." Today, more than half a century later, we do have translation software, but nothing that can approach the quality of human translation. Clearly, achievement of human level machine intelligence is a challenge that is hard to meet.

Humans have many remarkable capabilities; there are two that stand out in importance. First, the capability to reason, converse and make rational decisions in an environment of imprecision, uncertainty, incompleteness of information, partiality of truth and possibility. And second, the capability to perform a wide variety of physical and mental tasks without any measurements and any computations. A prerequisite to achievement of human level machine intelligence is mechanization of these capabilities and, in particular, mechanization of natural language understanding. In my view, mechanization of these capabilities is beyond the reach of the armamentarium of AI – an armamentarium which in large measure is based on classical, Aristotelian, bivalent logic and bivalent-logic-based probability theory.

To make significant progress toward achievement of human level machine intelligence a paradigm shift is needed. More specifically, what is needed is an addition to the armamentarium of AI of two methodologies: (a) a nontraditional methodology of computing with words (CW) or more generally, NL-Computation; and (b) a countertraditional methodology which involves a progression from computing with numbers to computing with words. The centerpiece of these methodologies is the concept of

precisiation of meaning. Addition of these methodologies to AI would be an important step toward the achievement of human level machine intelligence and its applications in decision-making, pattern recognition, analysis of evidence, diagnosis and assessment of causality. Such applications have a position of centrality in our infocentric society.

Keywords: Machine intelligence; theory of perceptions; fuzzy logic;

1. Introduction

Achievement of human level machine intelligence (HLMI) has profound implications for modern society – a society which is becoming increasingly infocentric in its quest for efficiency, convenience and enhancement of quality of life. Achievement of human level machine intelligence has long been one of the basic objectives of AI. In the fifties of last century, the question "Can machines think?" was an object of many spirited discussions and debates (Dreyfus and Dreyfus 2000). Exaggerated expectations were the norm, with few exceptions. In an article "Thinking machines – a new field in electrical engineering," published in January l950, I began with a sample of headlines of articles which appeared in the popular press in the late forties. One of them read "Electric brain capable of translating foreign languages is being built." Today, more than half a century later, we have translation software, but nothing that approaches the level of human translation. In l948, on the occasion of inauguration of IBM's Mark l relay computer, Howard Aiken, Director of Harvard's Computation Laboratory, said "There is no problem in applied mathematics that this computer cannot solve." Today, there is no dearth of problems which cannot be solved by any supercomputer. Exaggerated expectations should be forgiven. As Jules Verne wrote at the turn of last century, "Scientific progress is driven by exaggerated expectations."

Where do we stand today? What can we expect in the future? Officially, AI was born in l956. Today, half a century later, there is much that AI can be proud of – but not in the realm of human level machine intelligence. A telling benchmark is summarization. We have software that can passably summarize a class of documents but nothing that can summarize miscellaneous articles, much less books. We have humanoid robots but nothing that can compare in agility with that of a four year old child. We can automate driving a car in very light city traffic but there is nothing on the horizon that could automate driving in Cairo. Far too often, we have to struggle with a dumb automated customer service system which we are forced to use. Such experiences make us keenly aware that human level machine intelligence is an objective rather than reality. The Turing Test lies far beyond. What should be noted, however, is that authoritative views within the AI community tend to be substantially more optimistic. Representative views can be found in (Guha and Lenat 1994; Hibbard 2002; H. Kurzweil 2005; McCarthy 2007; Minsky et al 2004).

In an article "A new direction in AI – toward a computational theory of perceptions," AI Magazine, 200l, I argued that, in large measure, the lack of significant progress in many realms of human level machine intelligence is attributable to AI's failure to develop a machinery for dealing with perceptions. Underlying human level machine intelligence are two remarkable human capabilities. First, the capability to perform a wide variety of physical and mental tasks, such as driving a car in heavy city traffic,

without any measurements and any computations. And second, the capability to reason, converse and make rational decisions in an environment of imprecision, uncertainty, incompleteness of information, partiality of truth and partiality of possibility. A principal objective of human level intelligence is mechanization of these remarkable human capabilities.

What is widely unrecognized within the AI community is that mechanization of these capabilities is beyond the reach of methods based on classical, Aristotelian, bivalent logic and bivalent-logic-based probability theory. In short, if the question is: Can human level machine intelligence be achieved through the use of methods based on bivalent logic and bivalent-logic-based probability theory, then in my view the answer is: No. If the question is: Can human level machine intelligence be achieved sometime in the future, then my answer is: Possibly, but the challenge will be hard to meet. Extensions of existing techniques will not be sufficient. Basically, what is needed is a paradigm shift. More specifically, what is needed is an addition to the armamentarium of AI of two methodologies: (a) a nontraditional methodology of computing with words (CW) or, more generally, NL-Computation; and (b) a countertraditional methodology which involves a progression from computing with numbers to computing with words. The centerpiece of these methodologies is the concept of precisiation of meaning – a concept drawn from fuzzy logic.

What is fuzzy logic? What does it have to offer? There are many misconceptions about fuzzy logic. The following précis of fuzzy logic is intended to correct the misconceptions.

Fuzzy logic is not fuzzy. Basically, fuzzy logic is a precise logic of imprecision and approximate reasoning. In fact, fuzzy logic is much more than a logical system. It has many facets. The principal facets are logical, fuzzy-set-theoretic, epistemic and relational (Figure 1). Most of the applications of fuzzy logic involve the concept of a linguistic variable and the machinery of fuzzy if-then rules. The formalism of linguistic variables and fuzzy if-then rules is associated with the relational facet. The cornerstones of fuzzy logic are graduation, granulation, precisiation and the concept of a generalized constraint (Figure 2). Graduation should be understood as an association of a concept with grades or degrees.

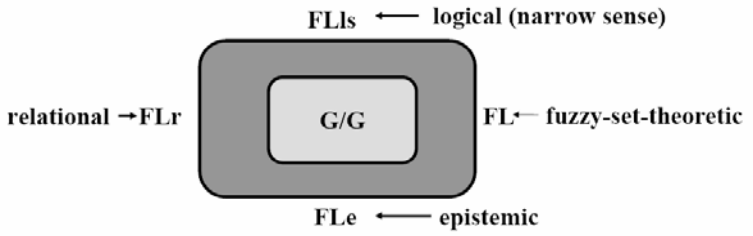


Figure 1. Principal facets of Fuzzy Logic (FL). The core of FL is Graduation/Granulation, G/G

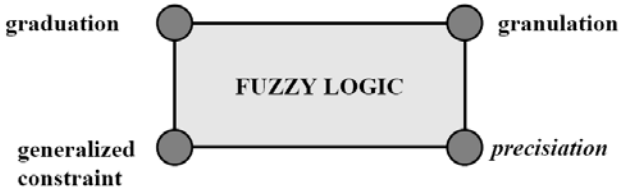


Figure 2. The cornerstones of a nontraditional view of fuzzy logic

In fuzzy logic, everything is or is allowed to be a matter of degree or, equivalently, fuzzy. Furthermore, in fuzzy logic everything is or is allowed to be granulated, with a granule being a clump of attribute values drawn together by indistinguishability, equivalence, proximity or functionality. Graduated granulation or, equivalently, fuzzy granulation is inspired by what humans employ to deal with complexity, imprecision and uncertainty. Graduated granulation underlies the concept of a linguistic variable. When Age, for example, is treated as a linguistic variable, its granular values may be young, middle-aged and old. The granular values of Age are labels of fuzzy sets. Informally, a fuzzy set is a class with unsharp boundary. A fuzzy set is defined by its membership function. A trapezoidal membership which defines middle-age is shown in Figure 3.

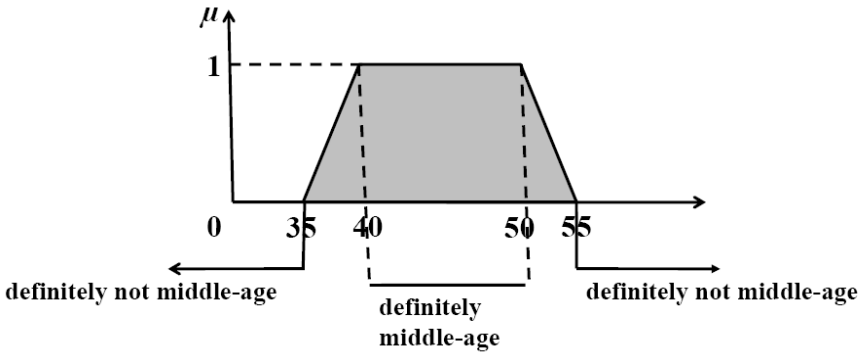


Figure 3. Imprecision of meaning

A concept which plays a pivotal role in fuzzy logic is that of a generalized constraint, (Zadeh 1986, 2008) represented as X isr R, where X is the constrained variable, R is the constraining relation and r is an indexical variable which defines the modality of the constraint, that is, its semantics. The principal generalized constraints are possibilistic, probabilistic and veristic. The fundamental thesis of fuzzy logic is that information may be represented as a generalized constraint. A consequence of the fundamental thesis is that the meaning of a proposition, p, may likewise be represented as a generalized constraint. The concept of a generalized constraint serves as a basis for representation of and computation with propositions drawn from a natural language. This is the province of NL-Computation/Computing with Words – computation with information described in natural language.

NL-Computation opens the door to achievement of human level machine intelligence. The validity of this assertion rests on two basic facts. First, much of human knowledge, and especially world knowledge, is described in natural language. And second, a natural language is basically a system for describing perceptions. What this implies is that NL-Computation serves two major functions: (a) providing a conceptual framework and techniques for precisiation of natural language in the context of human level machine intelligence; and (b) providing a capability to compute with natural language descriptions of perceptions. These capabilities play essential roles in progression toward human level machine intelligence.

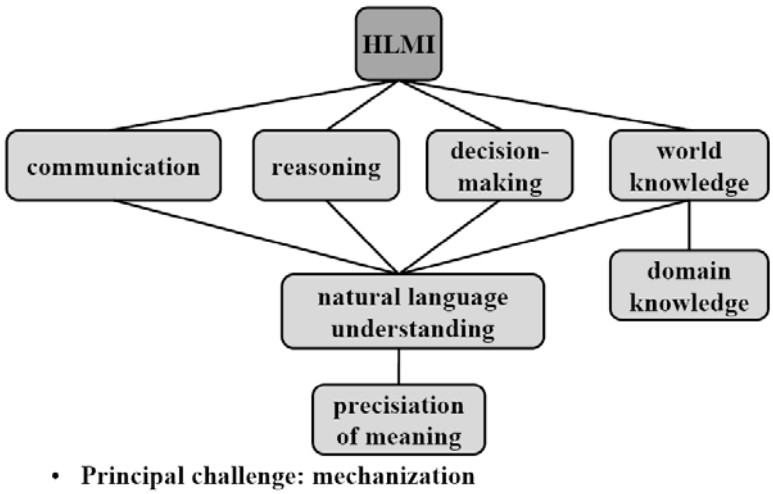


Figure 4. HLMI – Principal Concepts and Ideas

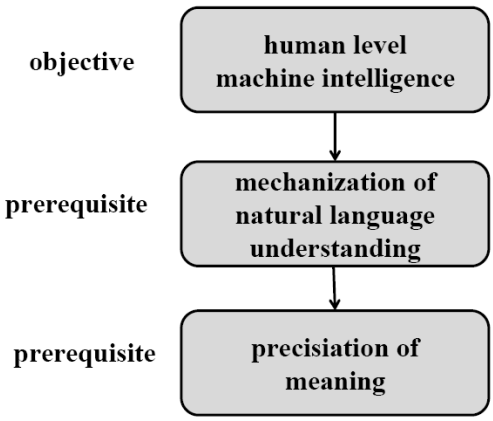


Figure 5. Achievement of human level machine intelligence

Human level machine intelligence has many components. The principal components are shown in Figure 4. Basically, achievement of human level machine intelligence requires a mechanization of the components of HLMI. Among the principal components of

HLMI the component which stands out in importance involves mechanization of natural language understanding. A prerequisite to mechanization of natural language understanding is precisiation of meaning. Humans can understand unprecisiated natural language but machines cannot (Figure 5).

What has been widely unrecognized is that in the final analysis, progress toward achievement of human level machine intelligence requires a resolution of a critical problem – the problem of precisiation of meaning. The two cornerstones of fuzzy logic - precisiation of meaning and the concept of a generalized constraint – are of direct relevance to human level machine intelligence. The primary purpose of this paper is to bring these concepts to the attention of the Computational Intelligence community. The exposition which follows is based on (Zadeh 2006) and (Zadeh 2008). Additional details may be found in these papers.

2. The Concept of Precisiation

In one form or another, precisiation of meaning has always played an important role in science. Mathematics is a quintessential example of what may be called a meaning precisiation language. Precisiation of meaning has direct relevance to mechanization of natural language understanding. For this reason, precisiation of meaning is an issue that is certain to grow in visibility and importance as we move further into the age of machine intelligence and automated reasoning.

Semantic imprecision of natural languages is a very basic characteristic – a characteristic which is rooted in imprecision of perceptions. Basically, a natural language is a system for describing perceptions. Perceptions are imprecise. Imprecision of perceptions entails semantic imprecision of natural languages.

The concept of precisiation has few precursors in the literature of logic, probability theory and philosophy of languages (Carnap 1950). The reason is that the conceptual structure of bivalent logic – on which the literature is based – is much too limited to allow a full development of the concept of precisiation. In HLMI what is used for this purpose is the conceptual structure of fuzzy logic.

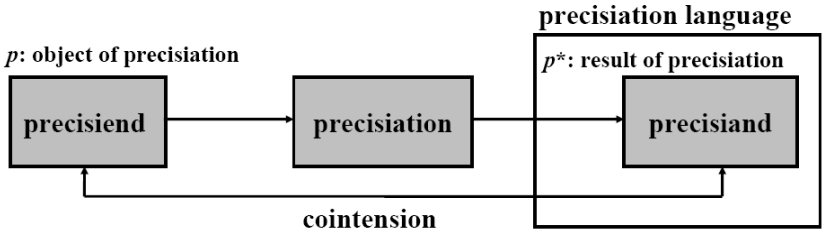


Figure 6. Basic concepts

Precisiation and precision have many facets. More specifically, it is expedient to consider what may be labeled λ-precisiation, with λ being an indexical variable whose values identify various modalities of precisiation. In particular, it is important to differentiate between precision in value (*v*-precision) and precision in meaning (*m*-precision). For example, proposition $X = 5$ is *v*-precise and *m*-precise, but proposition 2

$\leq X \leq 6$, is *v*-imprecise and *m*-precise. Similarly, proposition "*X* is a normally distributed random variable with mean 5 and variance 2," is *v*-imprecise and *m*-precise. Some of the basic concepts relating to precisiation are defined in Figure 6.

A further differentiation applies to *m*-precisiation. Thus, *mh*-precisiation is human-oriented meaning precisiation, while *mm*-precisiation is machine-oriented or, equivalently, mathematically-based meaning precisiation (Figure 7). A dictionary definition may be viewed as a form of *mh*-precisiation, while a mathematical definition of a concept, e.g., stability, is *mm*-precisiation whose result is *mm*-precisiand of stability.

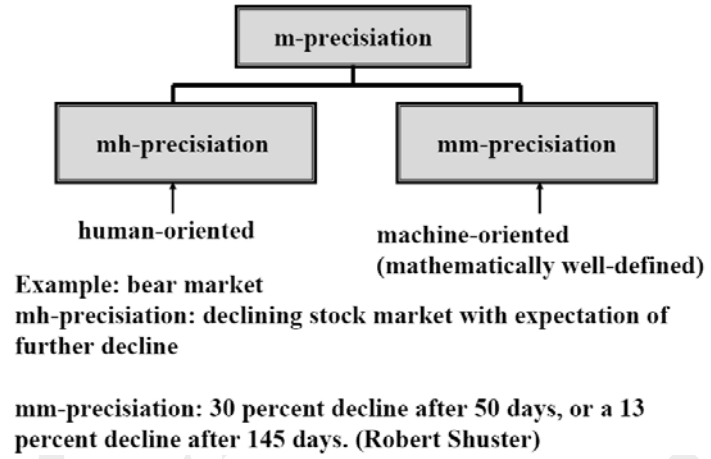


Figure 7. Modalities of m-precisiation

So far as imprecisiation is concerned, it may be forced or deliberate. Imprecisiation is forced when a precise value of a variable is not known. Imprecisiation is deliberate when a precise value is not needed and precision carries a cost. Familiar examples of deliberate imprecisiation are data compression and summarization. For convenience, the precisiand and imprecisiand of *p* are denoted as *p** and **p*, respectively.

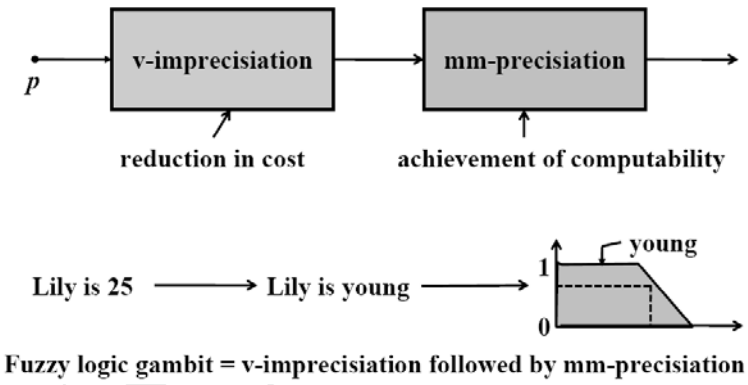


Figure 8. The fuzzy logic gambit

Note: Deliberate imprecisiation plays an important role in many applications of fuzzy logic especially in the realm of consumer products where cost is an important consideration. In such applications, what is employed is what is referred to as "The fuzzy logic gambit" (Figure 8). In the fuzzy logic gambit deliberate v-imprecisiation is followed by mm-precisiation.

A more general illustration of mm-precisiation relates to representation of a function as a collection of fuzzy if-then rules – a mode of representation which is widely used in practical applications of fuzzy logic (Yen and Langari 1998). More specifically, let *f* be a function from reals to reals which is represented as (Figure 9).

f: if *X* is small then *Y* is small

if *X* is medium than *Y* is large

if *X* is large than *Y* is small

where small, medium and large are labels of fuzzy sets. In this representation, the collection in question may be viewed as *mh*-precisiand of *f*. When the collection is interpreted as a fuzzy graph (Zadeh 1974, 1996) representation of *f* assumes the form.

*f**: small×small+medium×large+large×small

which is a disjunction of Cartesian products of small, medium and large. This representation is *mm*-precisiand of *f*.

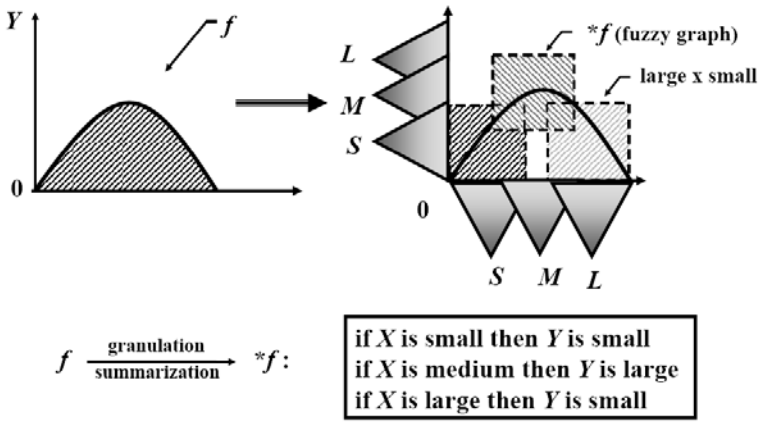


*Figure 9. Granulation of a function. S (small), M (medium) and L (large) are fuzzy sets. The granuland of f, *f, may be viewed as a summary of f*

In general, a precisiend may have many precisiands. As an illustration, consider the proposition "*X* is approximately *a*," or "*X* is **a*" for short, where *a* is a real number. How can "*X* is **a*" be precisiated?

The simplest precisiand of "*X* is **a*" is "*X* = *a*," (Figures 10a and 10b). This mode of precisiation is referred to as *s*-precisiation, with *s* standing for singular. This is a mode of precisiation which is widely used in science and especially in probability theory. In the latter case, most real-world probabilities are not known exactly but in practice are

frequently computed with as if they are exact numbers. For example, if the probability of an event is stated to be 0.7, then it should be understood that 0.7 is actually *0.7, that is, approximately 0.7. A standard practice is to treat *0.7 as 0.7000…, that is, as an exact number.

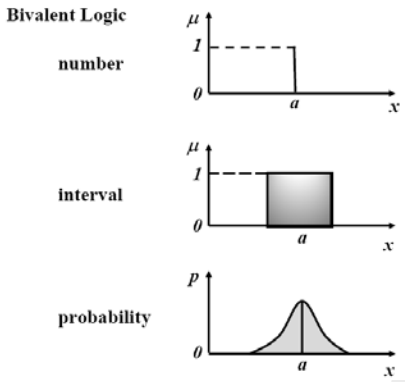


*Figure 10a. Alternative modes of mm-precisiation of "approximately a," *a, within the framework of bivalent logic*

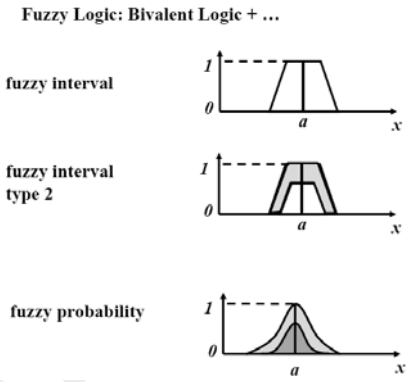


*Figure 10b. Alternative modes of mm-precisiation of "approximately a," *a, within the framework of fuzzy logic*

Next in simplicity is representation of *a is an interval centering on a. This mode of precisiation is referred to cg-precisiation, with cg standing for crisp-granular. Next is fg-precisiation of *a, with the precisiand being a fuzzy interval centering on a. Next is p-precisiation of *a, with the precisiand being a probability distribution centering on a. And so on.

An analogy is helpful in understanding the relationship between a precisiend and its precisiands. More specifically, a mm-precisiand, p*, may be viewed as a model of precisiend, p, in the same sense as a differential equation may be viewed as a model of a physical system.

In the context of modeling, an important characteristic of a model is its "closeness of fit." In the context of NL-Computation, an analogous concept is that of cointension. The concept is discussed in the following.

3. The Concept of Cointensive Precisiation

Precisiation is a prerequisite to computation with information described in natural language. To be useful, precisiation of a precisiend, p, should result in a precisiand, p*, whose meaning, in some specified sense, should be close to that of p. Basically, cointension of p* and p is the degree to which the meaning of p* fits the meaning of p.

In dealing with meaning, it is necessary to differentiate between this intension or, equivalently, the intensional meaning, i-meaning, of p, and the extension, or, equivalently, the extensional, e-meaning of p. The concepts of extension and intension are drawn from logic and, more particularly, from modal logic and possible world semantics (Cresswell 1973, Lambert 1970, Belohlavek 2006). Basically, e-meaning is attribute-free and i-meaning is attribute-based. As a simple illustration, if A is a finite set

in a universe of discourse, U, then the *e*-meaning of A, that is, its extension is the list of elements of A, $\{u_1, \ldots, u_n\}$, u_i being the name of *i*th element of A, with no attributes associated with the u_i. Let $a(u_i)$ be an attribute-vector associated with each u_i. Then the intension of A is a recognition algorithm which, given $a(u_i)$, recognizes whether u_i is or is not an element of A. If A is a fuzzy set with membership function μ_A then the *e*-meaning and *i*-meaning of A may be expressed compactly as

e-meaning of A: $A=\{\mu_A(u_i)/u_i\}$

where $\mu_A(u)/u$ means that $\mu_A(u)$ is the grade of membership of u_i in A; and

i-meaning of A: $A=\{\mu_A(a(u_i))/u_i\}$,

with the understanding that in the *i*-meaning of A the membership function, μ_A is defined on the attribute space. It should be noted that when A is defined through exemplification, it is said to be defined ostensively. Thus, *o*-meaning of A consists of exemplars of A. An ostensive definition may be viewed as a special case of extensional definition. A neural network may be viewed as a system which derives *i*-meaning from *o*-meaning.

Clearly, *i*-meaning is more informative than *e*-meaning. For this reason, cointension is defined in terms of intensions rather than extensions of precisiend and precisiand. Thus, meaning will be understood to be *i*-meaning, unless stated to the contrary. However, when the precisiend is a concept which plays the role of definiendum and we know its extension but not its intension, cointension has to involve the extension of the definiendum (precisiend) and the intension of the definiens (precisiand).

As an illustration, let p be the concept of bear market. A dictionary definition of p – which may be viewed as a *mh*-precisiand of p – reads "A prolonged period in which investment prices fall, accompanied by widespread pessimism." A widely accepted quantitative definition of bear market is: We classify a bear market as a 30 percent decline after 50 days, or a 13 percent decline after 145 days. (Shuster) This definition may be viewed as a *mm*-precisiand of bear market. Clearly, the quantitative definition, p^*, is not a good fit to the perception of the meaning of bear market which is the basis for the dictionary definition. In this sense, the quantitative definition of bear market is not cointensive.

Intensions are more informative than extensions in the sense that more can be inferred from propositions whose meaning is expressed intensionally rather than extensionally. The assertion will be precisiated at a later point. For the present, a simple example will suffice.

Consider the proposition p: Most Swedes are tall. Let U be a population of n Swedes, $U = (u_1, \ldots, u_n)$, u_1 = name of *i*th Swede.

A precisiand of p may be represented as

$$\frac{1}{n}\text{Count(tall.Swedes) is most}$$

where most is a fuzzy quantifier which is defined as a fuzzy subset of the unit interval (Zadeh 1983). Let $\mu_{tall}(u_i)$, i=1, …, n) be the grade of membership of u_i in the fuzzy set of tall Swedes. Then the *e*-meaning of tall Swedes may be expressed in symbolic form as

$$\text{tall.Swedes} = \mu_{tall}(u_1)/u_1+\cdots+\mu_{tall}(u_n)/u_n.$$

Accordingly, the *i*-precisiand of *p* may be expressed as

$$\frac{1}{n}\ (\mu_{tall}(u_1)+\cdots+\mu_{tall}(u_n) \text{ is most.}$$

Similarly, the *i*-precisiand of *p* may be represented as

$$\frac{1}{n}\ (\mu_{tall}(h_1)+\cdots+\mu_{tall}(h_n) \text{ is most.}$$

As will be seen later, given the *e*-precisiend of *p* we can compute the answer to the query: How many Swedes are not tall? The answer is 1-most. However, we cannot compute the answer to the query: How many Swedes are short? The same applies to the query: What is the average height of Swedes? As will be shown later, the answers to these queries can be computed given the *i*-precisiand of *p*.

The concept of cointensive precisiation has important implications for the way in which scientific concepts are defined. The standard practice is to define a concept within the conceptual structure of bivalent logic, leading to a bivalent definition under which the universe of discourse is partitioned into two classes: objects which fit the concept and those which do not, with no shades of gray allowed. Such definition is valid when the concept that is defined, the definiendum, is crisp, that is, bivalent. The problem is that in reality most scientific concepts are fuzzy, that is, are a matter of degree. Familiar examples are the concepts of causality, relevance, stability, independence and bear market. In general, when the definiendum (precisiend) is a fuzzy concept, the definiens (precisiand) is not cointensive, which is the case with the bivalent definition of bear market. More generally, bivalent definitions of fuzzy concepts are vulnerable to the Sorites (heap) paradox (Sainsbury 1995).

4. A Key Idea – the Meaning Postulate

In fuzzy logic, a proposition, *p*, is viewed as an answer to a question, *q*, of the form "What is the value of *X*?" Thus *p* is a carrier of information about *X*. In this perspective, the meaning of *p*, *M*(*p*), is the information which *p* carries about *X*. An important consequence of the fundamental thesis of fuzzy logic is what is referred to as the meaning postulate. In symbolic form, the postulate is expressed as $M(p) = GC(X(p))$, where $GC(X(p))$ is a generalized constraint on the variable which is constrained by *p*. In plain words, the meaning postulate assents that the meaning of a proposition may be represented as a generalized constraint. It is this postulate that makes the concept of a generalized constraint a cornerstone of fuzzy logic. By providing a mechanism for precisiating the meaning of a proposition, the concept of a generalized constraint opens the door to a wide-ranging enlargement of the role of natural languages in scientific

theories. What is referred to as PNL, Precisiated Natural Language, serves this purpose (Zadeh 2004).

A point which should be noted is that the question to which p is an answer is not uniquely determined by p; hence $X(p)$ is not uniquely defined by p. Generally, however, among the possible questions there is one which is most likely. For example, if p is "Monika is young," then the most likely question is "How old is Monika?" In this example, X is Age(Monika).

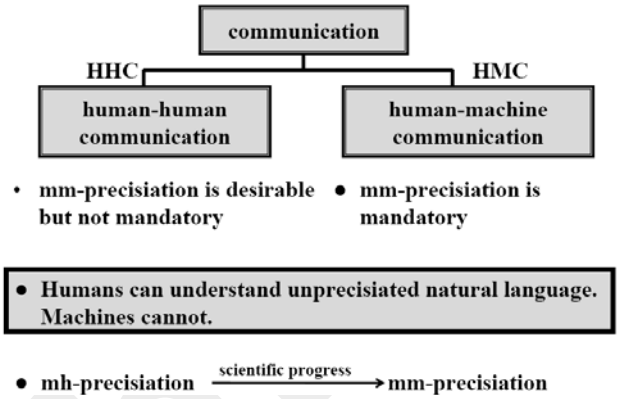


Figure 11. Precisiation in communication

The concept of precisiation has a direct bearing on communication in the context of human level machine intelligence (Figure 11). In communication between humans, HHC, precisiation of meaning is desirable but not mandatory. In communication between a human and a machine, HMC, precisiation is mandatory because a machine cannot understand unprecisiated natural language. In the case of HMC, an important issue relates to whether the precisiator is the sender (human, s-precisiation) or the recipient (machine, r-precisiation) (Figure 12). In most applications of fuzzy logic, the precisiator is the sender (human); an example is the Honda fuzzy logic transmission (Figure 13). In the case of s-precisiation, context-dependence is not a problem. As a consequence, precisiation is a much simpler function than it is in the case of r-precisiation. It should be noted that mechanization of natural language understanding involves for the most part r-precisiation.

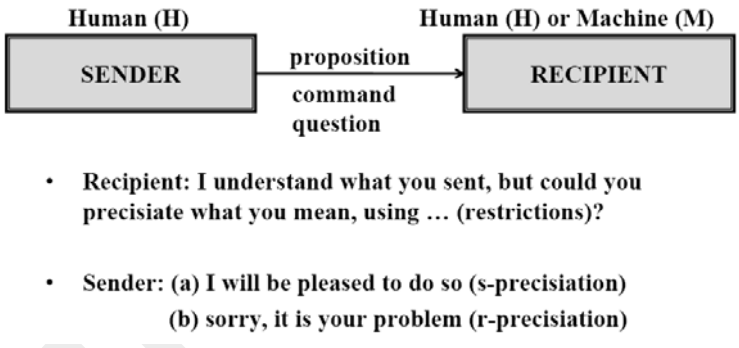


Figure 12. Precisiation in communication – Basic idea

In HHC, mm-precisiation is a major application area for generalized-constraint-based semantics (Zadeh 2008). Generalized-constraint-based semantics provides a basis for reformulation of bivalent-logic-based definitions of scientific concepts, associating Richter-like scales with concepts which are traditionally defined as bivalent concepts but in reality are fuzzy concepts. Examples: recession, civil war, arthritis, randomness, causality.

5. The Concept of a Generalized Constraint

Constraints are ubiquitous. A typical constraint is an expression of the form $X \in C$, where X is the constrained variable and C is the set of values which X is allowed to take. A typical constraint is hard (inelastic) in the sense that if u is a value of X then u satisfies the constraint if and only if $u \in C$.

The problem with hard constraints is that most real-world constraints are not hard, meaning that most real-world constraints have some degree of elasticity. For example, the constraints "check-out time is 1 pm," and "speed limit is 100 kmh," are, in reality, not hard. How can such constraints be defined? The concept of a generalized constraint is motivated by questions of this kind (Zadeh 2008).

Real-world constraints may assume a variety of forms. They may be simple in appearance and yet have a complex structure. Reflecting this reality, a generalized constraint, $GC(X)$, is defined as an expression of the form.

$$GC(X): \quad X \text{ isr } R,$$

where X is the constrained variable; R is a constraining relation which, in general, is non-bivalent; and r is an indexing variable which identifies the modality of the constraint, that is, its semantics.

The constrained variable, X, may assume a variety of forms. In particular,

- X is an n-ary variable, $X = (X_1, \ldots, X_n)$
- X is a proposition, e.g., X = Leslie is tall
- X is a function
- X is a function of another variable, $X = f(Y)$
- X is conditioned on another variable, X/Y
- X has a structure, e.g., X = Location(Residence(Carol))
- X is a group variable. In this case, there is a group, $G[A]$; with each member of the group, Name_i, $i = 1, \ldots, n$, associated with an attribute-value, A_i. A_i may be vector-valued. Symbolically

  $G[A]$: $\text{Name}_1/A_1 + \ldots + \text{Name}_n/A_n$.

  Basically, $G[A]$ is a relation.
- X is a generalized constraint, $X = Y$ isr R.

A generalized constraint is associated with a test-score function, $ts(u)$, (Zadeh 1982) which associates with each object, u, to which the constraint is applicable, the degree to which u satisfies the constraint. Usually, $ts(u)$ is a point in the unit interval. However, if necessary, the test-score may be a vector, an element of a semiring (Rossi 2003), an element of a lattice (Goguen 1969) or, more generally, an element of a partially ordered set, or a bimodal distribution – a constraint which will be described later. The test-score function defines the semantics of the constraint with which it is associated.

The constraining relation, R, is, or is allowed to be, non-bivalent (fuzzy). The principal modalities of generalized constraints are summarized in the following.

6. Principal Modalities of Generalized Constraints

(a) *Possibilistic* (r = blank)

X is R

with R playing the role of the possibility distribution of X. For example:

X is $[a, b]$

means that $[a, b]$ is the set of possible values of X. Another example:

X is small.

In this case, the fuzzy set labeled small is the possibility distribution of X (Zadeh 1978; Dubois and Prade 1988). If μ_{small} is the membership function of small, then the semantics of "X is small" is defined by

$$\text{Poss}\{X = u\} = \mu_{small}(u)$$

where u is a generic value of X.

(b) *Probabilistic* ($r = p$)

X isp R,

with R playing the role of the probability distribution of X. For example,

X isp $N(m, \sigma^2)$

means that X is a normally distributed random variable with mean m and variance σ^2.

If X is a random variable which takes values in a finite set $\{u_1, \ldots, u_n\}$ with respective probabilities $p_1, \ldots, p_n$, then X may be expressed symbolically as

$$X \text{ isp } (p_1 \backslash u_1 + \ldots + p_n \backslash u_n),$$

with the semantics

$$\text{Prob}(X = u_i) = p_i, \qquad i = 1, \ldots, n.$$

What is important to note is that in fuzzy logic a probabilistic constraint is viewed as an instance of a generalized constraint.

When X is a generalized constraint, the expression

X isp R

is interpreted as a probability qualification of X, with R being the probability of X, (Zadeh 1979a, 1981). For example.

(X is small) isp likely,

where small is a fuzzy subset of the real line, means that probability of the fuzzy event {X is small} is likely. More specifically, if X takes values in the interval [a, b] and g is the probability density function of X, then the probability of the fuzzy event "X is small" may be expressed as (Zadeh 1968)

$$\text{Prob}(X \text{ is small}) = \int_a^b \mu_{small}(u)\, g(u)\, du$$

Hence

$$ts(g) = \mu_{likely}\left(\int_b^a g(u)\, \mu_{small}(u)\, du\right).$$

This expression for test-score function defines the semantics of probability qualification of a possibilistic constraint.

(c) *Veristic* ($r = v$)

X isv R,

where R plays the role of a verity (truth) distribution of X. In particular, if X takes values in a finite set $\{u_1, \ldots, u_n\}$ with respective verity (truth) values $t_1, \ldots, t_n$, then X may be expressed as

X isv $(t_1|u_1 + \ldots + t_n|u_n)$,

meaning that $Ver(X = u_i) = t_i$, $i = 1, \ldots, n$.

For example, if Robert is half German, quarter French and quarter Italian, then

Ethnicity(Robert) isv 0.5|German+0.25|French+0.25|Italian

When X is a generalized constraint, the expression

X isv R

is interpreted as verity (truth) qualification of X. For example,

(X is small) isv very.true,

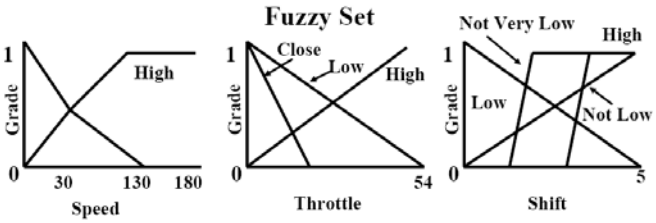
should be interpreted as "It is very true that X is small." The semantics of truth qualification is defined by (Zadeh 1979b)

$$\text{Ver}(X \text{ is } R) \text{ is } t \qquad X \text{ is } \mu_R^{-1}(t),$$

where μ_R^{-1} is inverse of the membership function of R, and t is a fuzzy truth value which is a subset of [0, 1], Figure 13.

(d) *Usuality* ($r = u$)

X isu R.



Control Rules:

1. **If (speed is low) and (shift is high) then (-3)**
2. **If (speed is high) and (shift is low) then (+3)**
3. **If (throt is low) and (speed is high) then (+3)**
4. **If (throt is low) and (speed is low) then (+1)**
5. **If (throt is high) and (speed is high) then (-1)**
6. **If (throt is high) and (speed is low) then (-3)**

Figure 13. Honda fuzzy logic transmission

The usuality constraint presupposes that X is a random variable, and that probability of the event $\{X \text{ isu } R\}$ is usually, where usually plays the role of a fuzzy probability which is a fuzzy number (Kaufman and Gupta 1985). For example.

X isu small

means that "usually X is small" or, equivalently,

Prob$\{X$ is small$\}$ is usually

In this expression, small may be interpreted as the usual value of X. The concept of a usual value has the potential of playing a significant role in decision analysis, since it is more informative than the concept of an expected value.

(e) *Random-set* $(r = vs)$

In

X isrs R

X is a fuzzy-set-valued random variable and R is a fuzzy random set

(f) *Fuzzy-graph* $(r = fq)$

In

X isfg R

X is a function, f, and R is a fuzzy graph (Zadeh 1974, 1996) which constrains f (Figure 14). A fuzzy graph is a disjunction of Cartesian granules expressed as

$$R = A_1 \times B_1 + \ldots + A_n \times B_n,$$

where the A_i and B_i, i=1, …, n, are fuzzy subsets of the real line, and × is the Cartesian product. A fuzzy graph is frequently described as a collection of fuzzy if-then rules (Zadeh 1973, 1996; Pedrycz and Gomide 1998; Bardossy and Duckstein 1995).

R: if X is A_i then Y is B_i, i=1, …, n.

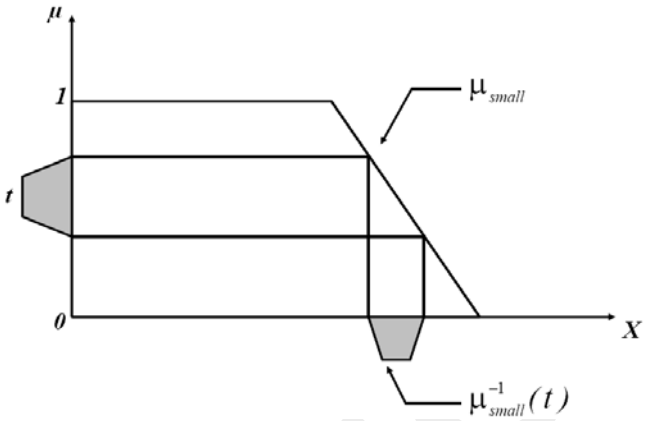


Figure 14. Truth-qualification: (X is small) is t

The concept of a fuzzy-graph constraint plays an important role in applications of fuzzy logic (Bardossy and Duckstein 1995; Filev and Yager 1994; Jamshidi, Titli, Zadeh and Boverie 1997; Yen, Langari and Zadeh 1995).

7. The Concept of Bimodal Constraint/Distribution

In the bimodal constraint,

X isbm R,

R is a bimodal distribution of the form

$$R: \quad \Sigma_i \; P_i \backslash A_i, \qquad i=1, \ldots, n.$$

with the understanding that Prob(X is A_i) is P_i. (Zadeh 2002), that is, P_i is a granular value of Prob (X is A_i), i=1, …, n. (See next section for definition of granular value).

To clarify the meaning of a bimodal distribution it is expedient to start with an example. I am considering buying Ford stock. I ask my stockbroker, "What is your perception of the near-term prospects for Ford stock?" He tells me, "A moderate decline is very likely; a steep decline is unlikely; and a moderate gain is not likely." My question is: What is the probability of a large gain?

Information provided by my stock broker may be represented as a collection of ordered pairs:

- Price: ((unlikely, steep.decline), (very.likely, moderate.decline), (not.likely, moderate.gain))

In this collection, the second element of an ordered pair is a fuzzy event or, generally, a possibility distribution, and the first element is a fuzzy probability. The expression for Price is an example of a bimodal distribution.

The importance of the concept of a bimodal distribution derives from the fact that in the context of human-centric systems, most probability distributions are bimodal. Bimodal distributions can assume a variety of forms. The principal types are Type 1, Type 2 and

Type 3 (Zadeh 1979a, 1981). Type 1, 2 and 3 bimodal distributions have a common framework but differ in important detail (Figure 15). A bimodal distribution may be viewed as an important generalization of standard probability distribution. For this reason, bimodal distributions of Type 1, 2, 3 are discussed in greater detail in the following.

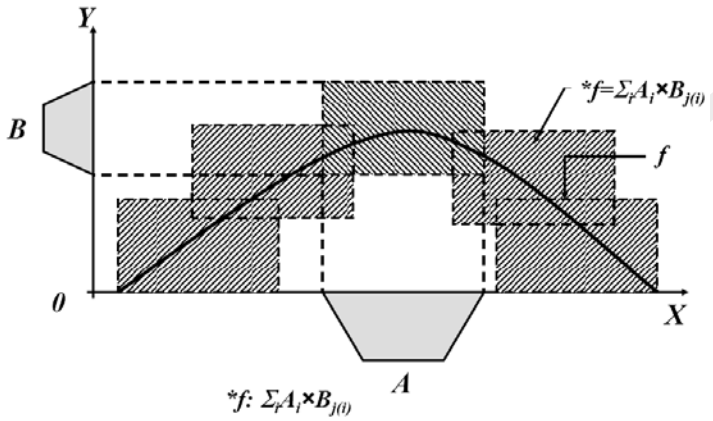


*Figure 15. Fuzzy-graph extension principle. B= *f (A)*

- Type 1 (default): X is a random variable taking values in U

  $A_1, \ldots, A_n, A$ are events (fuzzy sets)

  $p_i = \text{Prob}(X \text{ is } A_i)$, i = 1, …, n

  $\Sigma_i p_i$ is unconstrained

  P_i= granular value of P_i

  BD: bimodal distribution: $((P_1, A_1), \ldots, (P_n, A_n))$ or, equivalently,

  X isbm $(P_1\backslash A_1 + \cdots + P_n\backslash A_n)$

  Problem: What is the granular probability, P, of A? In general, this probability is fuzzy-set-valued.

- Type 2 (granule-valued distribution): X is a fuzzy-set-valued random variable with values

  $A_1, \ldots, A_n$ (fuzzy sets)

  $P_i = \text{Prob}(X = A_i)$, $i = 1, \ldots, n$

  P_i: granular value of P_i

  BD: X isrs $(P_1\backslash A_1 + \cdots + P_n\backslash A_n)$

  $\Sigma_i p_i = 1$

  Problem: What is the granular probability, P, of A? P is not definable. What are definable are (a) the expected value of the conditional possibility of A

given BD, and (b) the expected value of the conditional necessity of A given BD.

- Type 3 (Dempster-Shafer) (Dempster 1967, Shafer 1976): X is a random variable taking values $X_1, \ldots, X_n$ with probabilities $p_1, \ldots, p_n$

  X_i is a random variable taking values in A_i, $i = 1, \ldots, n$

  Probability distribution of X_i in A_i, $i = 1, \ldots, n$, is not specified

  X isp $(p_1\backslash X_1+\cdots+p_n\backslash X_n)$

  Problem: What is the probability, p, that X is in A? Because probability distributions of the X_i in the A_i are not specified, p is interval-valued. What is important to note is that the concepts of upper and lower probabilities break down when the A_i are fuzzy sets (Zadeh 1979a).

Note: In applying Dempster-Shafer theory it is important to check on whether the data fit Type 3 model. In many cases, the correct model is Type 1 rather than Type 3.

The importance of bimodal distributions derives from the fact that in many realistic settings a bimodal distribution is the best approximation to our state of knowledge. An example is assessment of degree of relevance, since relevance is generally not well defined. If I am asked to assess the degree of relevance of a book on knowledge representation to summarization, my state of knowledge about the book may not be sufficient to justify an answer such as 0.7. A better approximation to my state of knowledge may be "likely to be high." Such an answer is an instance of a bimodal distribution.

8. Primary Constraints, Composite Constraints and Standard Constraints

Among the principal generalized constraints there are three that play the role of primary generalized constraints. They are:

Possibilistic constraint: X is R

Probabilistic constraint: X isp R

and

Veristic constraint: X isv R

A special case of primary constraints is what may be called standard constraints: bivalent possibilistic, probabilistic and bivalent veristic. Standard constraints form the basis for the conceptual framework of bivalent logic and probability theory.

A generalized constraint is composite if it can be generated from other generalized constraints through conjunction, and/or projection and/or constraint propagation and/or qualification and/or possibly other operations. For example, a random-set constraint may be viewed as a conjunction of a probabilistic constraint and either a possibilistic or veristic constraint. The Dempster-Shafer theory of evidence is, in effect, a theory of possibilistic random-set constraints. The derivation graph of a composite constraint defines how it can be derived from primary constraints.

The three primary constraints – possibilistic, probabilistic and veristic – are closely related to a concept which has a position of centrality in human cognition – the concept

of partiality. In the sense used here, partial means: a matter of degree or, more or less equivalently, fuzzy. In this sense, almost all human concepts are partial (fuzzy). Familiar examples of fuzzy concepts are: knowledge, understanding, friendship, love, beauty, intelligence, belief, causality, relevance, honesty, mountain and, most important, truth, likelihood and possibility. Is a specified concept, *C*, fuzzy? A simple test is: If *C* can be hedged, then it is fuzzy. For example, in the case of relevance, we can say: very relevant, quite relevant, slightly relevant, etc. Consequently, relevance is a fuzzy concept.

The three primary constraints may be likened to the three primary colors: red, blue and green. In terms of this analogy, existing theories of uncertainty may be viewed as theories of different mixtures of primary constraints. For example, the Dempster-Shafer theory of evidence is a theory of a mixture of probabilistic and possibilistic constraints. The Generalized Theory of Uncertainty (GTU) (Zadeh 2006) embraces all possible mixtures.

9. The Generalized Constraint Language and Standard Constraint Language

A concept which has a position of centrality in GTU is that of Generalized Constraint Language (GCL). Informally, GCL is the set of all generalized constraints together with the rules governing syntax, semantics and generation. Simple examples of elements of GCL are:

(X is small) is likely

((X,Y) isp A) $\wedge$ (X is B)

(X isp A) $\wedge$ ((X,Y) isv B)

Proj_Y((X is A) $\wedge$ (X,Y) isp B)

where $\wedge$ is conjunction.

A very simple example of a semantic rule is:

(X is A) $\wedge$ (Y is B) $\longrightarrow$ Poss(X is A) $\wedge$ Poss(Y is B) = $\mu_A(u) \wedge \mu_B(v)$,

where u and v are generic values of X, Y; and μ_A and μ_B are the membership functions of A and B, respectively.

In principle, GCL is an infinite set. However, in most applications only a small subset of GCL is likely to be needed.

A key idea which underlies NL-Computation is embodied in the meaning postulate – a postulate which asserts that the meaning of a proposition, p, drawn from a natural language is representable as a generalized constraint. In the context of GCL, the meaning postulate asserts that p may be precisiated through translation into GCL. Transparency of translation may be enhanced through annotation. Simple example of annotation,

Monika is young $\longrightarrow$ X/Age (Monika) is R/young

In fuzzy logic, the set of all standard constraints together with the rules governing syntax, semantics and generation constitute the Standard Constraint Language (SCL). SCL is a subset of GCL.

In previous sections, we have employed the concept of a granular value in an informal fashion, without formulating a definition. The concept of a generalized constraint makes it possible to define a granular value more precisely. Let X be a variable taking values in a universe of discourse U, $U = \{u\}$. If a is an element of U, and it is known that the value of X is a, then a is referred to as a singular value of X. If there is some uncertainty about the value of X, the available information induces a restriction on the possible values of X which may be represented as a generalized constraint $GC(X)$, X isr R. Thus a generalized constraint defines a granule which is referred to as a granular value of X, $Gr(X)$ (Figure 16). For example, if X is known to lie in the interval $[a, h]$, then $[a, h]$ is a granular value of X. Similarly, if X isp $N(m, \sigma^2)$, then $N(m, \sigma^2)$ is a granular value of X. What is important to note is that defining a granular value in terms of a generalized constraint makes a granular value *mm*-precise. It is this characteristic of granular values that underlies the concept of a linguistic variable (Zadeh 1973). Symbolically, representing a granular value as a generalized constraint may be expressed as $Gr(X) = GC(X)$. It should be noted that, in general, perception-based information is granular (Figure 17).

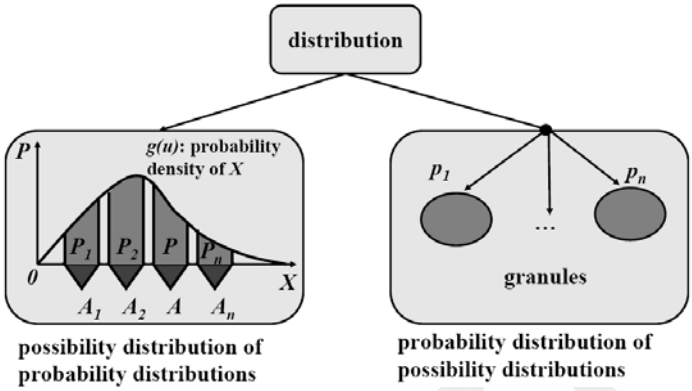


Figure 16. Bimodal distributions. Granular vs. granule-valued distributions

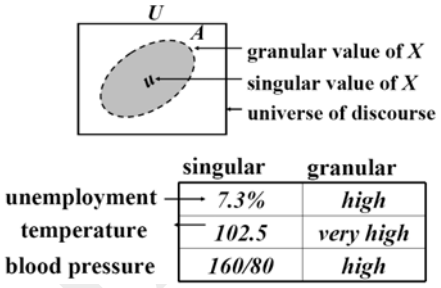


Figure 17. Singular and granular values

The importance of the concept of a granular value derives from the fact that it plays a central role in computation with information described in natural language. More specifically, when a proposition expressed in a natural language is represented as a system of generalized constraints, it is, in effect, a system of granular values. Thus, computation with information described in natural language ultimately reduces to computation with granular values. Such computation is the province of Granular Computing (Zadeh 1979a, 1998; Bargiela and Pedrycz 2002).

10. Concluding Remarks

There are many reasons why achievement of human level machine intelligence is a challenge that is hard to meet. One of the principal reasons is the need for mechanization of two remarkable human capabilities. First, the capability to converse, communicate, reason and make rational decisions in an environment of imprecision, uncertainty, incompleteness of information, partiality of truth and partiality of possibility. And second, the capability to perform a wide variety of physical and mental tasks – such as driving a car in heavy city traffic – without any measurements and any computations. What is well understood is that a prerequisite to mechanization of these capabilities is mechanization of natural language understanding. But what is widely unrecognized is that mechanization of natural language understanding is beyond the reach of methods based on bivalent logic and bivalent-logic-based probability theory. In addition, what is widely unrecognized is that mechanization of natural language understanding is contingent on precisiation of meaning.

Humans can understand unprecisiated natural language but machines cannot. Natural languages are intrinsically imprecise. Basically, a natural language is a system for describing perceptions. Perceptions are intrinsically imprecise. Imprecision of natural languages is rooted in imprecision of perceptions.

The principal thesis of this paper is that to address the problem of precisiation of meaning it is necessary to employ the machinery of fuzzy logic. In addition, the machinery of fuzzy logic is needed for mechanization of human reasoning. In this perspective, fuzzy logic is of direct relevance to achievement of human level machine intelligence. The cornerstones of fuzzy logic are the concepts of graduation, granulation, precisiation and generalized constraint.

Acknowledgements

To Ronald Yager and Bernadette Bouchon-Meunier. Research supported in part by ONR N00014-02-1-0294, BT Grant CT1080028046, Omron Grant, Tekes Grant, Chevron Texaco Grant, Ministry of Communications and Information Technology of Azerbaijan and the BISC Program of UC Berkeley.

References and related papers

[1] Bargiela, A., Pedrycz, W.: *Granular Computing,* Kluwer Academic Publishers, (2002).

[2] Bardossy, A., Duckstein, L.: *Fuzzy Rule-Based Modelling with Application to Geophysical, Biological and Engineering Systems*, CRC Press, (1995).

[3] Belohlavek, R., Vychodil, V.: *Attribute Implications in a Fuzzy Setting,* B. Ganter and L. Kwuida (Eds.): ICFCA 2006, Lecture Notes in Artificial Intelligence 3874, Springer-Verlag, Heidelberg, (2006), pp. 45-60.
[4] Carnap, R.: *The Logical Foundations of Probability*, University of Chicago Press, (1950).
[5] Cresswell, M. J.: *Logic and Languages,* Methuen, London, UK, (1973).
[6] Dempster, A. P.: *Upper and Lower Probabilities Induced by a Multivalued Mapping,* Ann. Math Statist. 38, (1967), pp. 325-329.
[7] Dreyfus, H., Dreyfus, S.: *Mind and Machine,* Free Press, New York, (2000).
[8] Dubois, D., Prade, H.: *Representation and combination of uncertainty with belief functions and possibility measures,* Computational Intelligence 4, (1988), pp. 244-264.
[9] Filev, D., Yager, R. R.: *Essentials of Fuzzy Modeling and Control,* Wiley-Interscience, (1994).
[10] Goguen, J. A.: *The logic of inexact concepts*, Synthese, vol. 19, (1969), pp. 325-373.
[11] Guha, R. V.: Douglas B.: *Lenat, Enabling Agents to Work Together,* Commun. ACM 37(7), (1994), pp. 126-142.
[12] Hibbard, W.: *Super Intelligent Machines,* Springer, New York, (2002).
[13] Jamshidi, M., Titli, A., Zadeh, L. A., Boverie, S. (eds): *Applications of Fuzzy Logic – Towards High Machine Intelligence Quotient Systems*, Environmental and Intelligent Manufacturing Systems Series, Prentice Hall, Upper Saddle River, NJ, Vol. 9, (1997).
[14] Kaufmann, A., Gupta, M. M.: *Introduction to Fuzzy Arithmetic: Theory and Applications,* Von Nostrand, New York, (1985).
[15] Kurzweil, R.: *Singularity is Near,* Viking Press, New York, (2005).
[16] Lambert, K., Van Fraassen, B. C.: *Meaning Relations, Possible Objects and Possible Worlds,* Philosophical Problems in Logic, (1970), pp. 1-19.
[17] McCarthy, J.: *From here to human-level AI*, Artif. Intell. 171(18), (2007), pp. 1174-1182.
[18] Minsky, M., Singh, P., Sloman, A.: *The St. Thomas Common Sense Symposium: Designing Architectures for Human-Level Intelligence,* AI Magazine 25(2), (2004), pp. 113-124.
[19] Pedrycz, W., Gomide, F.: *Introduction to Fuzzy Sets,* MIT Press, Cambridge, MA, (1998).
[20] Rossi, F., Codognet, P.: *Soft Constraints, Special issue on Constraints,* Kluwer, vol. 8, no. 1., (2003).
[21] Sainsbury, R. M.: *Paradoxes,* Cambridge University Press, (1995).
[22] Shafer, G.: *A Mathematical Theory of Evidence,* Princeton University Press, Princeton, NJ, (1976).
[23] Yen, J., Langari, R., Zadeh, L. A. (Ed.): *Industrial Applications of Fuzzy Logic and Intelligent Systems,* IEEE, (1995).
[24] Yen, J. Langari, R.: *Fuzzy Logic: Intelligence, Control and Information,* Prentice Hall, 1st edition, (1998).
[25] Zadeh, L. A.: *Probability measures of fuzzy events,* Jour. Math. Analysis and Appl. 23, (1968), pp. 421-427.

[26] Zadeh, L. A.: *Outline of a new approach to the analysis of complex systems and decision processes,* IEEE Trans. on Systems, Man and Cybernetics SMC-3, (1973), pp. 28-44.
[27] Zadeh, L. A.: *On the analysis of large scale systems,* Systems Approaches and Environment Problems, H. Gottinger (ed.), Vandenhoeck and Ruprecht, Gottingen, (1974), pp. 23-37.
[28] Zadeh, L. A.: *Fuzzy sets and information granularity,* Advances in Fuzzy Set Theory and Applications, M. Gupta, R. Ragade and R. Yager (eds.), North-Holland Publishing Co., Amsterdam, (1979), pp. 3-18.
[29] Zadeh, L. A.: *A theory of approximate reasoning,* Machine Intelligence 9, Hayes, J., Michie, D., Mikulich, L. I. (eds.), New York: Halstead Press, (1979), pp. 149-194.
[30] Zadeh, L. A.: *Possibility theory and soft data analysis,* Mathematical Frontiers of the Social and Policy Sciences, Cobb, L. Thrall, R. M. (eds.), Boulder, Westview Press, CO, (1981), pp. 69-129.
[31] Zadeh, L. A.: *Test-score semantics for natural languages and meaning representation via PRUF,* Empirical Semantics, Rieger, B. (ed), Brockmeyer, Bochum, W. Germany, (1982). pp. 281-349.
[32] Zadeh, L. A.: *A computational approach to fuzzy quantifiers in natural languages,* Computers and Mathematics 9, (1983), pp. 149-184.
[33] Zadeh, L. A.: *Outline of a computational approach to meaning and knowledge representation based on the concept of a generalized assignment statement,* Proceedings of the International Seminar on Artificial Intelligence and Man-Machine Systems, Thoma, M., Wyner, A. (eds.), Springer-Verlag, Heidelberg, (1986), pp. 198-211.
[34] Zadeh, L. A.: *Fuzzy logic and the calculi of fuzzy rules and fuzzy graphs,* Multiple-Valued Logic 1, (1996), pp. 1-38.
[35] Zadeh, L. A.: *Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems,* Soft Computing 2, (1998), pp. 23-25.
[36] Zadeh, L. A.: *Toward a perception-based theory of probabilistic reasoning with imprecise probabilities***,** Journal of Statistical Planning and Inference, Elsevier Science, Vol. 105, (2002), pp. 233-264.
[37] Zadeh, L. A.: *Precisiated Natural Language (PNL),* AI Magazine, Vol. 25, No. 3, (2004), pp. 74-91.
[38] Zadeh, L. A.: *Generalized theory of uncertainty (GTU) – principal concepts and ideas,* Computational Statistics & Data Analysis 51, (2006) pp. 15-46.
[39] Zadeh, L. A.: *Is there a need for fuzzy logic?* Information Sciences, Vol. 178, No. 13, (2008), pp. 2751-2779.

A Neural Network Seismic Detector

Guilherme Madureira*, António E. Ruano**

*** Institute of Meteorology, Geophysical Center of S. Teotónio,**
7630-585 Portugal
e-mail: guilherme.madureira@meteo.pt
**** Centre for Intelligent Systems, University of Algarve,**
8005-139 Portugal
e-mail: aruano@ualg.pt

Abstract: This experimental study focuses on a detection system at the seismic station level that should have a similar role to the detection algorithms based on the ratio STA/LTA. We tested two types of neural network: Multi-Layer Perceptrons and Support Vector Machines, trained in supervised mode. The universe of data consisted of 2903 patterns extracted from records of the PVAQ station, of the seismography network of the Institute of Meteorology of Portugal. The spectral characteristics of the records and its variation in time were reflected in the input patterns, consisting in a set of values of power spectral density in selected frequencies, extracted from a spectrogram calculated over a segment of record of pre-determined duration. The universe of data was divided, with about 60% for the training and the remainder reserved for testing and validation. To ensure that all patterns in the universe of data were within the range of variation of the training set, we used an algorithm to separate the universe of data by hyper-convex polyhedrons, determining in this manner a set of patterns that have a mandatory part of the training set. Additionally, an active learning strategy was conducted, by iteratively incorporating poorly classified cases in the training set. The best results, in terms of sensitivity and selectivity in the whole data ranged between 98% and 100%. These results compare very favorably with the ones obtained by the existing detection system, 50%.

Keywords**:** *Seismic detector, neural networks, support vector machines, spectrogram.*

1. Introduction

There is a growing interest in seismology for increasing the speed and the reliability of the automatic processing of seismic data acquired by the monitoring system. The application of Artificial Neural Networks (ANN) in this field, and more specifically, the automatic detection of seismic events, has been tested for some years and is a promising path of current research.

We propose a seismic detection system, to be implemented at the seismic station, using ANN. This system should be able to distinguish segments of seismic records containing

signal caused by local and regional events, from all other situations. The aim is to build a classifier that assigns one of two class periods of the seismic record of pre-determined fixed duration, Class 1, local and regional natural earthquakes, and class 2, all the other possibilities.

In the last two decades several researchers worked in the field of automatic seismic detection with neural networks. Some of those studies are presented below.

(Masotti et al., 2006) applied a Support Vector Machine (SVM) to classify volcanic tremor data at Etna volcano, Italy. Trained in a supervised way, the classifier should recognize patterns belonging to four classes; pre-eruptive, lava fountains, eruptive, and post-eruptive. 425 spectrogram based feature vectors were used for training. The system correctly classified 94.7 ± 2.4% of the data in validation.

In (Abu-Elsoud et al., 2004) an automatic system is proposed to discriminate between local earthquakes and local explosions in the Suez Gulf area, Egypt. The system is ANN-based and is composed of two modules; a feature extractor that quantifies the seismogram signatures using a Linear Prediction Code and a classifier to discriminate the seismic events. The data used is a set of 320 seismic events recorded by the Egyptian National Seismic Network; 142 records are explosions and 178 are local earthquakes. Validation results achieved 93.7% of correct classifications.

To detect distant seismic events automatically, (Tiira, 1999), proposes a Multi-Layer Perceptron (MLP) trained with the Error-Back-Propagation algorithm. The entries in this network are instantaneous values of STA/LTA (see Section 2.2) calculated with 4 different windows of STA, in 7 frequency bands. 193 distant seismic events were used in the training process. Comparing with the Murdock-Hutt detector (Murdock and Hutt, 1983), this system detected 25% more events, and produced 50% less false alarms.

(Dai and MacBeth, 1997) proposed a Back-Propagation Neural Network (BPNN) to identify P (Primary) and S (Secondary) arrivals (Udías, 2000) from three-component recordings of local earthquake data. The BPNN was trained by selecting trace segments of P and S waves and noise bursts, converted into an attribute space based on the Degree of Polarization (DOP). 1363 seismic records were used for training and validation. Compared with a manual analysis, the trained system can correctly identify between 76.6% and 82.3% of the P arrivals, and between 60.5% and 62.6% of the S arrivals.

The detection of seismic events was the objective of the study presented in (Wang and Teng, 1995). Two ANN were trained in supervised mode with different types of inputs: In one case the ratio STA/LTA was used, the other used spectrogram as input feature. Experiments have shown that these systems performed better than those algorithms based on a threshold of the STA/LTA ratio.

In this work we used data collected from the seismic station PVAQ[1], located in Vaqueiros, Algarve, in southern Portugal.

[1] In general, Portuguese seismic stations begin with a "P", that stands for Portugal, followed by an abbreviation of the location name, in this case "VAQ" stands for Vaqueiros.

The structure of the paper is as follows. In section 2, the procedures used for data collection and feature extraction are described. The training methods used in the experiment are also indicated in this section. In section 3 the experiments are described and the results analyzed. Conclusions and future work are expressed in section 4.

2. Data and Training Methods

2.1. Input Data

Non-stationary signals occur naturally in many real-world applications: Examples include speech, music, biomedical signals, radar, sonar and seismic waves. Time-frequency representations such as the spectrograms are important tools for processing such time-varying signals. In this work, the spectrogram is used as the first stage of earthquake detection.

The Power Spectrum Density (PSD) is estimated using periodogram averaging (Welch, 1967). Only positive frequencies are taken into account (the so-called one-sided PSD). PSD values are slightly smoothed by taking the average of PSD values in a constant relative bandwidth of 1/10 of a decade. The procedure to achieve that smoothness was as follows: Let $P(f)$ be the PSD values in some set of discrete frequencies F. Starting with the lowest frequency of F, (f_{min}), we created a sequence of frequencies separated by 1/10 of a decade,

$$f_k = f_{\min} 10^{\frac{k-1}{10}}, \quad k = 1, 2, \ldots \tag{1}$$

We then split F into disjoint subsets D_k,

$$D_k = \{ f \} : f_k \le f \le f_{k+1}, f \in F, k = 1, 2, \ldots \tag{2}$$

each set D_k is associated with a frequency f_k as defined above. The smoothed PSD, $P_s(f_k)$, is given by,

$$P_S(f_k) = \frac{1}{\# D_k} \sum_{f \in D_k} P(f) \tag{3}$$

We have divided segments of 120 seconds into 5 non-overlapping intervals. For each one of them we computed the PSD. This is done with standard Matlab functions. We then picked the power at 6 frequencies 1, 2, 4, 8, 10 and 15 Hz. This means that 30 different features will be used for the classifier. This was a constraint that we imposed, in order to limit the classifiers complexity. Fig. 1 illustrates a seismic-record and its spectrogram, highlighting the frequencies selected.

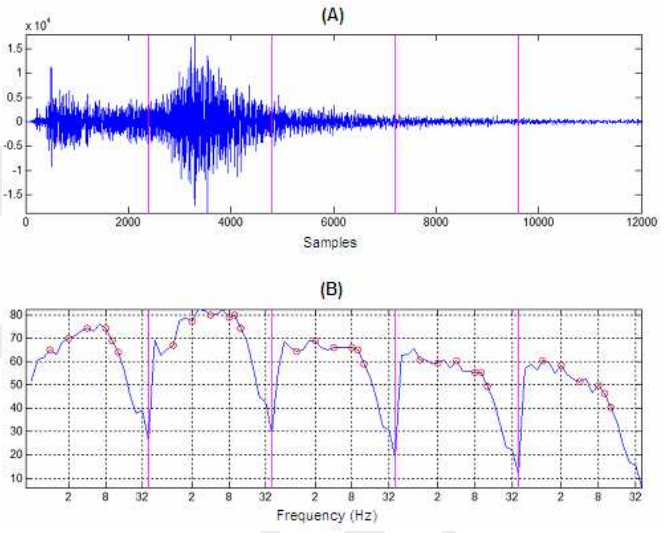


Figure 1. (A) 120 sec of seismic record (B) Spectrogram

In most experiments, a Butterworth digital high-pass filter was applied to the signal previous to PSD computation. The cut-off frequency was 0.5 Hz and the order of the filter was 5. This procedure intended to remove low frequency content from the spectrum, since for local and regional seismic events those frequencies are out of the main bandwidth of interest.

2.2. Target Data

Seismic data, previously classified was collected from the PVAQ station of the seismic monitoring system of the Institute of Meteorology of Portugal (IM). Seismic data was classified by seismologists of the National Data Center (NDC) at IM. The seismic detector used at a station level is a standard STA/LTA ratio based detector (Stewart, 1977). Fig. 2 outlines the operation of such a detector.

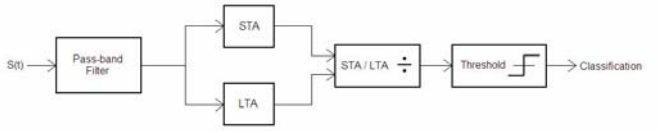


Figure 2. Block diagram of a typical STA/LTA detector

The input data is band-pass filtered to maximise sensitivity within a specific frequency band of interest, and to reject noise outside this band. Averages of the modulus of signal amplitude are computed over two user-defined time periods, a short time average (STA) and a long time average (LTA), and the ratio of the two, (STA/LTA), at each sample point is computed. If this ratio exceeds a user-defined threshold, then a trigger is declared, and the system remains in a triggered state until the ratio falls below the defined threshold.

These detectors based on the ratio STA/LTA at the seismic station show in general very modest performance, i.e., large numbers of non detected seismic events and several false alarms. However, a seismic network can drastically improve the overall performance considering clusters of stations. The likelihood of noise events occurring in a given time interval at various stations is very small, thus reducing the likelihood of making false alarms. In addition, an event that is not detected by a particular station is likely to be detected by other stations of the group, thereby increasing dramatically the ability of detection. However, the automatic system at the NDC is always supervised by seismologists.

2.3. Collected Data

From the year of 2007, 2903 examples were collected, 502 representing the positive class (classified as earthquake by the seismologists at NDC, and where seismic phases were identified in the PVAQ records), and the other 2401 classified as non-seism. In the former case, the station detection system miss-classified 50% of the events. In the latter class, 50% of the examples were randomly selected representing events that triggered the detection system, but that were not classified as seismic by the NDC, while the rest of the examples were selected randomly, neither coinciding with events detected by the system nor classified as earthquakes by the NDC. This way, the station automatic detected system achieved values of 50% of Sensitivity and Specificity (measures introduced latter) in the data collected.

2.4. Training Methods

In this work, MLPs were used as classifiers. We shall briefly describe here the training method employed. For more information, the reader is referred to, for instance (Ruano et al., 2005).

First of all we assign to each positive example the value of +1, and to each negative example, the value of -1. Input data is scaled and the classifier nonlinear parameters are initialized with a stochastic procedure which does not exacerbate the condition number of the Jacobean matrix of the model.

Parameter estimation is achieved by applying the Levenberg-Marquardt algorithm (Ruano et al., 1992) for the minimization of a criterion that exploits the separability of the classifier parameters, as linear parameters are used in the output layer (Ruano et al., 1991). This process is applied to the training data, and terminates whether a local minimum is found, or the performance in another set, denoted here as a test set, deteriorates. This is the well-known method of early stopping (Haykin, 1999).

As indirectly, the test set is used in the determination of the classifiers, their performance is assessed in a third data set, denoted here as the validation set.

3. Results

3.1. First Experiment

The first experiment was conducted by assigning, randomly, 60% of the data to the training set, 20% to the test set, and 20% to the validation set. It was only ensured that a

similar percentage of positive cases was assigned to each data set. The training set consisted of 1744 examples, with 307 positive cases; the test set had 582 examples, with 99 positive cases; the validation set consisted of 577 events, with 96 positive cases.

In this first experiment, 20 different topologies of MLPs were tried, each one with 20 different parameters initializations. Moreover, the use (or not) of the filter described above was tested, resulting in 800 different classifiers.

The results are presented in terms of the Sensitivity, or Recall (R) criterion, defined as:

$$R = \frac{TP}{TP + FN}, \tag{4}$$

and in terms of Specificity (S) criterion, defined as…

$$S = \frac{TN}{FP + TN}, \tag{5}$$

where *TP*, *TN*, *FP* and *FN* denote the number of True Positives, True Negatives, False Positives and False Negatives, respectively. Moreover, these criteria are applied to the training, test and validation sets, separately, and to all the data. As the classification is casted as a multi-objective problem, we do not have a single optimum; instead a set of Non-Dominated (ND) solutions is obtained, where the elements have the property that no one is better (larger in this case) in all objectives than the other solutions belonging to the set. The following tables show the ND solutions found, for the three data sets, individually considered.

A line in italic indicates that the same ND classifier is present in the training and in the validation sets, while an underlined line indicates that a common ND classifier is obtained in the test and in the validation sets. Please note that as 20 different initializations were conducted for the same topology, equal entries in the topology column is not an indication that the same classifier is used. A mark in the column labelled as *F* indicates if filtering of the input data has been applied. The columns labelled as *R(All)* and *S(All)* show the Recall and the Specificity values computed for the whole data (the union of the training, test and validation data sets). The topology column shows the number of neurons in the first and the second hidden layers.

Table 1. Training set

Topology	F	R	S	R(All)	S(All)
[7 2]		91.86	99.23	92.43	99.25
[5 7]		96.74	96.66	96.81	97.17
[4 16]		96.09	97.56	94.82	97.88
[5 11]		93.49	98.96	93.43	98.96
[7 2]		94.79	98.12	95.62	98.25
[6 5]		94.46	98.75	94.62	98.88
[6 3]	*	92.18	99.10	93.23	99.13
[5 7]	*	95.44	98.05	95.82	98.50
[7 2]	*	96.42	97.49	96.61	98.00
[4 15]	*	92.51	99.03	93.23	99.04

Table 2. Test set

Topology	F	R	S	R(All)	S(All)
[7 2]		96.97	99.38	94.22	98.71
[5 8]		98.99	98.96	94.82	97.33
[4 19]		95.96	99.79	93.82	98.04
[6 2]		*94.95*	*100.00*	*94.82*	*98.79*

Table 3. Validation set

Topology	F	R	S	R(All)	S(All)
[6 2]		*97.92*	*99.17*	*94.82*	*98.79*
[7 2]		98.96	98.13	95.62	98.25
[6 3]	*	95.83	99.38	94.82	98.96
[7 2]	*	90.63	99.79	90.44	98.67

If we perform the same analysis for the three data sets together (i.e., considering as criteria the Selectivity and the Specificity for the training, the test and the validation sets, and subsequently determining the ND solutions), we obtain the union of the ND solutions for the three data sets considered separately, plus a significant number of additional Pareto solutions. In the total, 51 ND solutions were obtained. If we select the classifier by the total number of misclassifications (both positive and negative) in the whole data, 3 models achieve the smallest number, 51, in the full 2903 examples. One of the three solutions is shown in the 3rd line of Table 3, and the other two belong to additional ND solutions.

The results can also be presented as a ROC (Receiver Operating Characteristics) curve (Swets, 1988). The next three figures present these results, where, in every case, the ND solutions obtained considering the corresponding data set are shown as a red circle, and the ND solutions, considered the 6 criteria, are shown as blue diamonds.

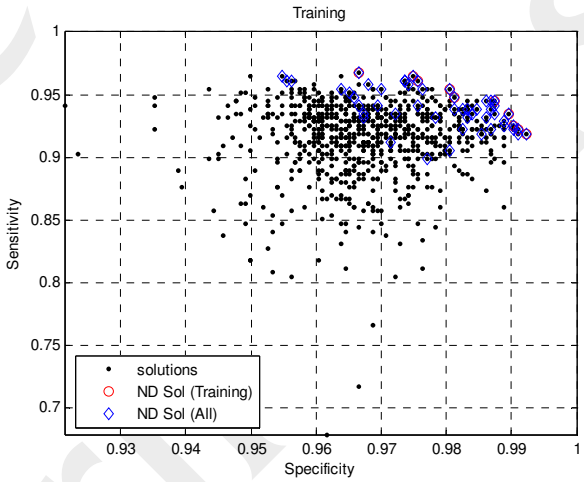


Figure 3. ROC for the training set

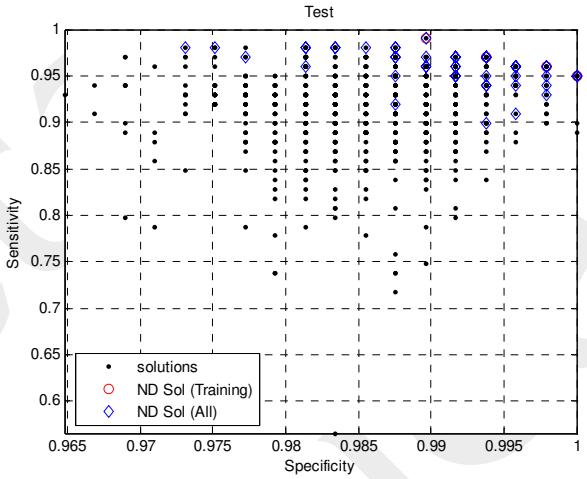


Figure 4. ROC for the test set

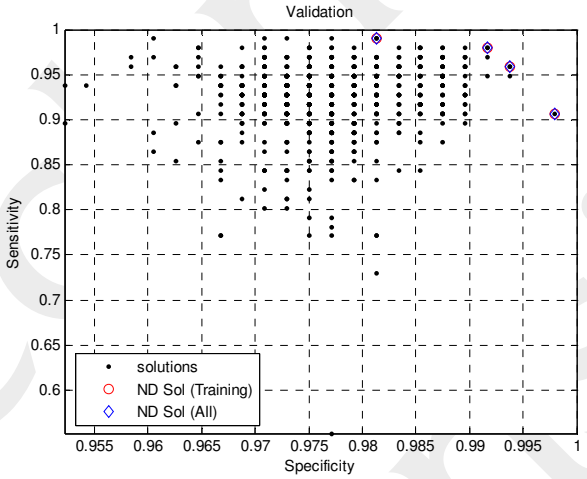


Figure 5. ROC for the validation set

We were therefore able to obtain classifiers with Recall and Specificity values above 95% (compared with the 50% values obtained by the existing detection system), and with a total number of misclassifications in the order of 50, compared with 1450, achieved by the existing system.

These results are also able to highlight that the use or not of the filter did not produce any significant difference. Filtered data will be used from now on.

3.2. Support Vector Machines

Another set of experiences regarding different partitioning of data between the training, test and validation sets was conducted. First of all, an approximate convex hull of the input data has been obtained, and the examples that lie in the hull were integrated in the

training set. In order to maintain an approximate distribution of 60%, 20% and 20% of the data to the three sets, examples of the original training set were moved to the other two sets. With this data partitioning, a Support Vector Machine (SVM) classifier, with a Gaussian kernel, was experimented. The implementation described in (Frieß et al., 1998) was used.

In this case the examples in the test and validation set were used as a single validation set. With a spread value of 0.237, the following results were obtained:

Table 4. SVM performance

SVs	R	S	R(All)	S(All)
583	100.00	100.00	99.62	99.35

Subsquently, a form of active learning (Cohn et al., 1994) was applied. The examples badly classified were incorporated in the training set, and randomly removed the same number of examples to the validation set, provided they were not in the approximate convex hull previously determined. This procedure was repeated three times. The results are presented in Table 5.

Table 5. SVM performance with active learning

SVs	R	S	R(All)	S(All)
609	100.00	100.00	99.72	99.66
626	100.00	100.00	99.76	99.72
640	100.00	100.00	100.00	99.93

This represents an almost perfect performance (only 2 misclassifications in the whole data). The major problem is the large complexity of the classifier, consisting of 640 support vectors. We therefore tried, with this new partitioning of data, to improve the performance of the MLP classifiers.

3.3. Further experiments with MLPs

We used 20 different topologies, each one with 10 different initializations. The non-dominated solutions obtained are shown below.

Table 6. Training set

Topology	R	S	R(All)	S(All)
[6 4]	**97.48**	**99.30**	**96.61**	**99.50**
[6 2]	99.37	99.09	96.81	98.82
[5 11]	96.21	99.44	92.23	99.29
[5 12]	95.27	99.79	92.43	98.83

Table 7. Test set

Topology	R	S	R(All)	S(All)
[4 20]	98.90	99.38	97.21	98.46
[5 8]	100.00	98.97	98.41	97.83
[6 4]	**94.51**	**99.79**	**96.61**	**99.50**
[4 19]	96.70	95.59	86.65	96.50

Table 8. Validation set

Topology	R	S	R(All)	S(All)
[6 2]	100.00	99.80	97.81	98.83

A line in bold indicates that the same ND classifier is obtained, considering the training set and the test set. The number of ND solution achieved, considering the three data sets, is 32. The best solution, in terms of the total number of miss-classifications, has a topology of [5 9], and it is not present in tables 6-8.

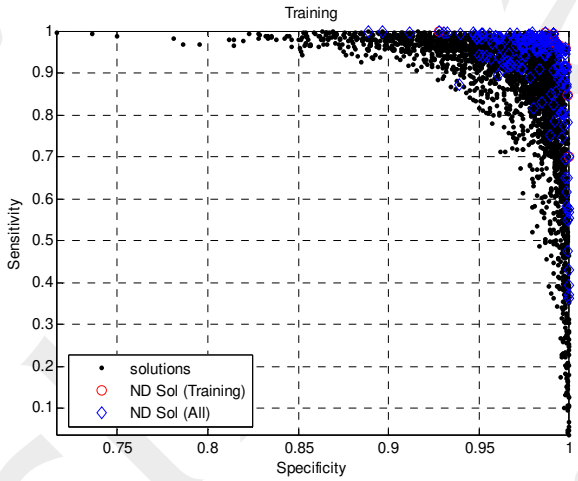


Figure 6. ROC for the training set

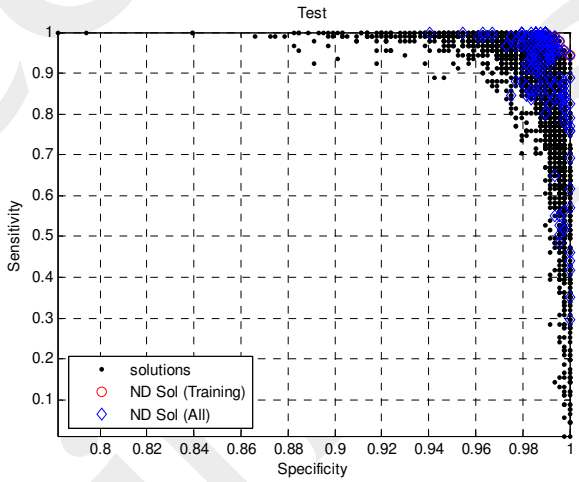


Figure 7. ROC for the test set

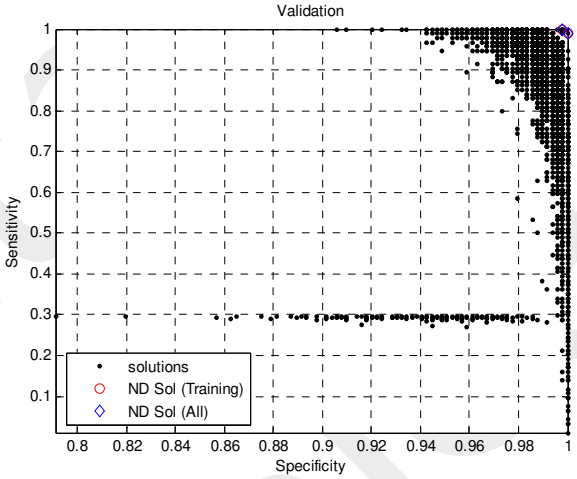


Figure 8. ROC for the validation set

We were able, with just a different data partitioning, to reduce the number of miss-classifications from 51 (please see Section 3.1) down to 29. This classifier presents a complexity, in terms of the number of parameters, of 219, compared with the solution obtained in Section 3.2, with 640 support vectors.

Further experiments were conducted, varying the decision threshold of the classifiers. No improvements, however, were obtained. The following figures show the ROC curves, for the training, test and validation sets.

Conclusions

With the same data that produced 50% Sensitivity and Selectivity values in an existing detection system, based on the LTA/STA ratio, we were able to obtain, in a first step, values greater than 95% for the two criteria. Using an active learning technique, we were able to improve the performance of our MLP classifiers to 98%. An SVM classifier was able to achieve almost perfect classification, albeit at the expense of a large complexity.

Although the results are encouraging, the work described in this paper must be considered as preliminary. At present the performance of the neural classifiers is being assessed in the whole 2007 record of the station employed. The analysis of the results will enable to construct a better off-line classifier. Then, our attention will be focused in on-line learning methods, so that the classifier learns with its on-line performance. Additional features can also be considered and a search for the best to use, together with the classifier topology, can be conducted by meta-heuristics. Finally, the use of data from different stations will be considered.

References

[1] Abu-Elsoud, M. A., Abou-Chadi, F. E. Z., Amin, A. E. M., Mahana, M.: *Classification of Seismic Events in Suez Gulf area, Egypt Using Artificial Neural Network,* ICEEE'04: 2004 International Conference on Electrical, Electronic and Computer Engineering, Proceedings, (2004), pp. 337-340.

[2] Cohn, D., Atlas, L., R, L.: *Improving Generalization with Active Learning,* Machine Learning, 15, (1994), pp. 201-221.

[3] Dai, H. C., MacBeth, C.: *The Application of Back-Propagation Neural Network to Automatic Picking Seismic Arrivals from Single-Component Recordings,* Journal of Geophysical Research-Solid Earth, 102, (1997), pp. 15105-15113.

[4] Frieß, T., Cristianini, N., Campbel, C.: *The Kernel Adatron Algorithm: A Fast and Simple Learning Procedure for Support Vector Machines,* 15th Intl. Conf. Machine Learning. Morgan Kaufmann Publishers, (1998).

[5] Haykin, S.: *Neural Networks: A Comprehensive Foundation,* Prentice Hall, (1999).

[6] Masotti, M., Falsaperla, S., Langer, H., Spampinato, S., Campanini, R.: *Application of Support Vector Machine to the Classification of Volcanic Tremor at Etna,* Italy. Geophysical Research Letters, 33. (2006).

[7] Murdock, J., Hutt, C.: *A New Event Detector Designed for the Seismic Research Observatories,* USGS Open-File-Report, (1983).

[8] Ruano, A. E., Ferreira, P. M., Fonseca, C. M.: *An Overview of Nonlinear Identification and Control with Neural Networks,* In Ruano, A. E. (Ed.) Intelligent Control using Intelligent Computational Techniques. IEE Control Series, (2005).

[9] Ruano, A. E. B., Fleming, P. J., Jones, D. I.: *A Connectionist Approach to PID Autotuning,* IEE Proceedings-D Control Theory and Applications, 139, (1992), pp. 279-285.

[10] Ruano, A. E. B., Jones, D. I., Fleming, P. J.: *A New Formulation of the Learning-Problem for a Neural Network Controller,* Proceedings of the 30th Ieee Conference on Decision and Control, Vols 1-3. (1991).

[11] Stewart, S. W.: *Real Time Detection and Location of Local Seismic Events in Central California,* Bulletin of Seismological Society of America 67, (1977), pp. 433-452.

[12] Swets, J.: *Measuring the Accuracy of Diagnostic Systems,* Science, 240, (1988), pp. 1285-1293.

[13] Tiira, T.: *Detecting Teleseismic Events using Artificial Neural Networks,* Computers & Geosciences, 25, (1999), pp. 929-938.

[14] Udías, A.: *Principles of Seismology,* Cambridge University Press. (2000).

[15] Wang, J., Teng, T. L.: *Artificial Neural-Network-Based Seismic Detector,* Bulletin of the Seismological Society of America, 85, (1995), pp. 308-319.

[16] Welch, P. D.: *Use of Fast Fourier Transform for Estimation of Power Spectra: A Method Based on Time Averaging Over Short Modified Periodograms,* IEEE Transactions on Audio and Electroacoustics, AU15, 70. (1967).

Development of Fuzzy System Models: Fuzzy Rulebases to Fuzzy Functions

I. Burhan Türkşen

Ph.D., P.Eng, Fellow: IFSA, IEEE, WIF
Hon. Doc.: Sakarya U., Azerbaycan Government U.
Head Department of Industrial Engineering
TOBB-Economics and Technology University
Sögütözü Cad. No:43, Sögütözü 06560 Ankara/Turkey
Tel:+90 312 292 4068, Cell: +90 533 501 8407
e-mail: bturksen@etu.edu.tr, www.etu.edu.tr
Director-Knowledge/Intelligence Systems Laboratory
Department of Mechanical & Industrial Engineering
University of Toronto,
Toronto, Ontario, M5S 3G8, Canada,
Ph/Fax: (416) 978-1278 (Direct)
e-mail: turksen@mie.utoronto.ca
www.mie.utoronto.ca/staff/profiles/turksen.html

Abstract We first review the development of Fuzzy System Models from "Fuzzy Rule bases" proposed by Zadeh (1965, 1975) and applied by Mamdani, et al. (1981) to "Fuzzy Functions" proposed by Turksen (2007-2008) and further developed by Celikyilmaz and Turksen (2007-2009) in a variety of versions. Furthermore, we also review a comlementary development of "Fuzzy C-Regression Model", (FCRM) proposed by Hathaway and Bezdek, (1993) and a "Combined FCM, and FCRM algorithms" proposed by Höppner and Klawonn (2003).

1. Introduction

"Fuzzy Functions" were defined by John Grinder and Richard Bandler in The Structure of Magic Volume II (1976) as a connecting or overlapping of our sensory representational systems. In their sense, "Fuzzy Functions" generate a representational system where either an input or the output channel is a different modality from the representational system with which it is being used. In traditional psychophysics, this term, "fuzzy function", is most closely translated by the term "synesthesia". Furthermore, "Fuzzy Functions" are investigated from a strictly mathematical perspective by Moritoshi Sasaki (1993) and M. Demirci (1999), etc. In these mathematical studies "Fuzzy Function" structures contain only membership values as input variables. Where as in "Fuzzy Functions" proposed by Turksen (2007-2008) and further developed by Celikyilmaz and Turksen (2007-2009) in a variety of versions, "Fuzzy Function" structures contain both membership values as well as their suitable trasformations in addition to original input variables. On the other hand, alternate

"Fuzzy Function" structures proposed by Hathaway and Bezdek, (1993) together with a "Combined FCM, and FCRM algorithms" proposed by Höppner and Klawonn (2003) which generate linear and non-linear versions of "Fuzzy C-Regression Models" where membership values effect only the coefficients of regression equations.

In particular, Türksen (2007-2008) first introduced "Fuzzy Functions" unaware of the publications of John Grinder and Richard Bandler(1976), Moritoshi Sasaki(1993), and M. Demirci(1999) as well as Hathaway and Bezdek, (1993) and Höppner and Klawonn (2003). As indicated above, **"Fuzzy Functions with LSE" (2008)** are quite different in structure and intent from **Sasaki and Demirci** expositions as well as Hathaway and Bezdek, (1993) and Höppner and Klawonn (2003) approches.

2. Fuzzy Rule Bases

Fuzzy Rule Bases were originally proposed by Zadeh (1965, 1975) and later applied by Mamdani, et al. (1981):

$$R : \underset{i=1}{\overset{c^*}{\text{ALSO}}}\left(\text{IF } antecedent_i \text{ THEN } consequent_i\right)$$

Further developments and applications were introduced by Sugeno-Yasukawa (1993) as:

$$R : \underset{i=1}{\overset{c^*}{\text{ALSO}}}\left(\text{IF } antecedent_i \text{ THEN } consequent_i\right)$$

In the next stage of developments, Tagaki-Sugeno (1985) inroduced right hand side to be a regression equation while keeping the left hand side as a fuzzy rule:

$$R : \underset{i=1}{\overset{c^*}{ALSO}}\ (IF\ antecedent_i\ THEN\ y_i = a_i x^T + b_i)$$

In a historical context, we observe the introduction of Fuzzy Regression models as the further stage of developments. These are:

(a) Methods that were proposed by Tanaka (1991) and investigated by Tanaka, et al. (1982, 1991, 1995), Celmins (1987), Savic and Pedrycz (1991) in the literature, where the coefficients of input variables are assumed to be fuzzy numbers which are defined by experts.

(b) In contrast in the method proposed by Hathaway and Bezdek (1993), first fuzzy clusters are determined by FCM method to define how many ordinary regressions are to be constructed, i.e., one for each cluster, determined by "Fuzzy C-Regression Model" clustering algorithm (FCRM). Next each fuzzy cluster is used essentially for switching purposes to determine the most appropriate ordinary regression that has to be applied.

(c) Furthermore, the method proposed by Höppner and Klawonn combine FCM, Fuzzy C-Means, and FCRM algorithms in one clustering schema, to build a combined clustering structure. Their main goal was to update FCM fuzzy clustering algorithm so that they can prevent the effect of harmonics by modifying the objective function. It is to be noted that they not only deal with point-wise clustering algorithms such as "Fuzzy C-Means" (FCM) clustering algorithm, but as well, they also deal with "Fuzzy C-Regression Model" clustering algorithm (FCRM). It is also well-known that Hathaway

and Bezdek, (1993), proposed to build linear regression models. Whereas one can build non-linear regression models with Höppner and Klawonn, 2003, approach.

3. Fuzzy Functions with LSE

Let us now review the essential components of "Fuzzy Function" structuers originally proposed by Turksen (2007-2008) and further developed by Celikyilmaz and Turksen (2007-2009) in a variety of versions.

Let (X_k,Y_k), $k = 1,\ldots, nd$, be the set of observations in a training data set, such that $X_k = (x_{jk} \mid j = 1,\ldots, nv, k = 1,\ldots, nd)$. Determine the optimal (m^*, c^*) pair for a particular performance measure, i.e., a cluster validity index, with an iterative search with an application of FCM algorithm, or IFC algorithm Celikyilmaz and Turksen (2009) where m is the level of fuzziness (in our experiments we usually take $m = 1.1,\ldots,2.5$), and c is the number of clusters (in our experiments we usually take $c = 2,\ldots,10$).

Determine the optimal (m^*, c^*) pair for a particular performance measure, i.e., a cluster validity index, with an iterative search and an application of FCM algorithm, where m is the level of fuzziness (in our experiments we usually take $m = 1.1,\ldots,2.5$), and c is the number of clusters (in our experiments we usually take $c = 2,\ldots,10$).

4. FCM Algorithm

$$\min\ J(U,V) = \sum_{k=1}^{nd}\sum_{i=1}^{c}(u_{ik})^m(\|x_k - v_i\|)_A$$

$$s.t. \quad 0 \le u_{ik} \le 1, \forall i,k$$

$$\sum_{i=1}^{c} u_{ik} = 1, \forall k$$

$$0 \le \sum_{k=1}^{nd} u_{ik} \le nd, \forall i$$

where A = is the Euclidian Norm and A =C-1 is the Mahalonobis Norm, etc.

By running FCM algorithm one identifies the cluster centers for m = m and c=1,...,c* as:*

$$v_{\underset{m^*}{X|Y,j}} = (x_{1,j}{}^c, x_{2,j}{}^c, \cdots, x_{nv,j}{}^c, y_j{}^c)$$

$$v_{\underset{m^*}{X,j}} = (x_{1,j}{}^c, x_{2,j}{}^c, \cdots, x_{nv,j}{}^c)$$

Optimum Membership Values *and* Augmented Input Matrix are determined as follows:

$$u_{ik} = \left(\sum_{j=1}^{c} \left(\frac{\|x_k - v_{X,i}\|}{\|x_k - v_{X,j}\|} \right)^{\frac{2}{m-1}} \right)^{-1}, \qquad \mu_{ik} = \{u_{ik} \ge \alpha\},$$

Where with an alfa cut one eliminates unwanted harmonics and one gets the modified membership values as:

$$\gamma_{ij}(x_j) = \frac{\mu_{ij}(x_j)}{\sum_{i'=1}^{c} \mu_{i'j}(x_j)}$$

Thus one obtaines the matrix of membership values of an X data sample in the i-th cluster as :

$$\Gamma_i = (\gamma_{ij} \mid i = 1,\ldots,c^*;\ j = 1,\ldots,nd)$$

Examples of the possible augmented input matrices are:

$$X_i' = [1, \Gamma_i, X] \quad X_i''' = \left[1, \Gamma_i^2, \Gamma_i^m, \exp(\Gamma_i), X\right]$$

etc. For example, we get:

$$X'_{ij} = [1, \Gamma_i, X_{ij}] = \begin{bmatrix} 1 & \gamma_{i1} & x_{i1} \\ \vdots & \vdots & \vdots \\ 1 & \gamma_{ind} & x_{ind} \end{bmatrix}$$

Next one determines a Least Squared Estimation of output for each of the clusters made of both the membership values and the original input values together with their appropriate transformations.

Thus for the most simple case of one input variable and its associated membership values in the i-th cluster, obtain **Fuzzy (Regression) Functions with LSE (FF-LSE) as:**

$$Y_i = \beta_{i0} + \beta_{i1}\Gamma_i + \beta_{i2}X_{ij}$$

which represents the *ith* rule corresponding to the *ith* interactive (joint) cluster in space,

$$(Y_i, \Gamma_i, X_j)$$

which is estimated with FF-LSE approach as follows:

$$\beta_i^* = (X_{ij}'^T X'_{ij})^{-1} (X_{ij}'^T Y_i)$$

where

$$\beta_i^* = (\beta_{i0}^*, \beta_{i1}^*, \beta_{i2}^*)$$

are the estimates, provided that the inverse of covariance matrix, ( )$^{-1}$, exists. Therefore the estimate of Y_i would be obtained as:

$$Y^*_i = \beta^*_{i0} + \beta^*_{i1}\Gamma_i + \beta^*_{i2}X_{ij}$$

The overall output value is calculated as follows:

$$Y^*_i = \frac{\sum_{i=1}^{c^*} \gamma_i Y^*_i}{\sum_{i=1}^{c^*} \gamma_i}$$

In such a case of one dimensinal analyses, a graph of an idealized one cluster would be depicted as:

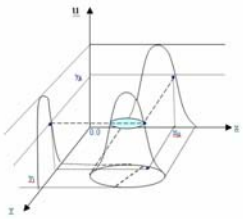


Figure 1. A Fuzzy cluster in [*UxXx*Y] space.

Further development on fuzzy functions investigated by **by Celikyilmaz and Turksen** can be found in the following published articles:

a) "Comparison of Fuzzy Functions with Fuzzy Rule Bases" (2006).

b) "Fuzzy functions with support vector machines" (*2007).*
c) "Enhanced Fuzzy System Models with Improved Fuzzy Clustering Algorithm", (2008).
d) "Uncertainty Modeling with Evolutionary Improved Fuzzy Functions Approach", (2008).
e) "Industrial Applications of Evolutionary Improved Fuzzy Functions", Journal of Computers, (2008).
f) "Increasing Accuracy of Two Class Pattern Recognition with Improved Fuzzy Functions", (2009).

5. Fuzzy C-Regression

Originally Fuzzy C-Regression Model (FCRM) [Hathaway and Bezdek, 1993] was introduced to classify objects into similar groups. FCRM yields simultaneous estimates of parameters for Fuzzy *C-R*egression models, while fuzzy partitioning a given dataset. It ought to be recalled that FCM is a point-wise clustering algorithms. Furthermore, FCM [Bezdek, 1981] clusters are hyper-sphere shaped. It should be noted that FCRM determines cluster prototypes as functions instead of geometrical objects. In particular, FCRM determines separate linear patterns, where each pattern can be identified by a linear function. It is to be noted that FCRM [Hathaway and Bezdek, 1993], clusters are hyper plane-shaped.

6. Differences of FCM and FCRM

It is well known that the representatives of clusters of FCM are cluster centers, υ_i.

Whereas the representatives of clusters in FCRM are hyper-planes, which are represented by:

$$y_i = \beta_i^0 + \beta_i^1 x_1 + \ldots + \beta_i^{nv} x_{nv}$$

where β_i are the regression coefficients of each function , *i=1…c.* FCM algorithm calculates cluster centers by averaging each data vector weighted with their membership

values. FCRM calculates cluster representative functions by weighted least squares regression algorithm as $y_k = f_i(\boldsymbol{x}_k, \beta_i)$ where $\boldsymbol{x}_k=[x_{1,k},\ldots,x_{nv,k}]^T \in \Re^{nv}$ denotes kth data object and $\beta_i \in \Re^{nv}$, $i=1,\ldots,c$. Performance of these functions is generally measured by:

$$E_{ik}(\beta_i) = (y_k - f_i(\boldsymbol{x}_k, \beta_i))^2$$

The objective function is to minimize the total error of these approximated functions:

$$E(U, \beta_i) = \sum_{i=1}^{c} \sum_{k=1}^{n} (\mu_{ik})^m E_{ik}(\beta_i)$$

In FCRM, μ_{ik},`s represent how close are the extent values predicted by $f_i(\boldsymbol{x}_k,\beta_i)$, to y_k. It should be recalled that for FCM:

$$\mu_{ik}^{(t)} = \left[\sum_{j=1}^{c} \left(\frac{d(x_k, \upsilon_i^{(t-1)})}{d(x_k, \upsilon_j^{(t-1)})} \right)^{\frac{2}{m-1}} \right]^{-1}$$

Where as From FCRM one gets:

$$\mu_{ik} = \left[\sum_{j=1}^{c} \left(E_{ik} / E_{jk} \right)^{\frac{1}{m-1}} \right]^{-1}, \forall\, i, j = 1, \ldots, c < n$$

CRM is formulated to find hidden structures in a given dataset.

Possible extensions of FCRM implement non-linear functions to find hidden patterns.

It is developed with

Min : $E(U,\beta_i) = \sum_{i=1}^{c} \sum_{k=1}^{n} (\mu_{ik})^m E_{ik}(\beta_i)$

Where $\beta_i = [X^T U_i X]^{-1} X^T U_i y$

$$X_i = \begin{bmatrix} x_{i,1}^T \\ x_{i,2}^T \\ \vdots \\ x_{i,n}^T \end{bmatrix}, y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, U_i = \begin{bmatrix} \mu_{i1} & 0 & \cdots & 0 \\ 0 & \mu_{i2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mu_{i,n} \end{bmatrix}$$

7. Non-Linear Fuzzy Regression

Höppner and Klawonn [2003] combine FCM [Bezdek, 1981] and FCRM [Hathaway and Bezdek, 1993] algorithms in one clustering schema, to build combined clustering structure. Their aim is to eliminate the counterintuitive membership values. They modified the objective function of FCM by combining it with FCRM.

$$\mu_{ik} = \left[\sum_{j=1}^{c} \frac{d_{ik}^2 - \left(\min_{i=1\ldots c} d_{ik}^2 - \eta \right)}{d_{jk}^2 - \left(\min_{i=1\ldots c} d_{ik}^2 - \eta \right)} \right]^{-1}, 0 < \eta$$

Where $\eta > 0$ is a user defined constant.

In [Höppner and Klawonn, 2003], each function, $\hat{y}_i = \hat{\beta}_i^T \boldsymbol{x}_i$

is interpreted as a rule in a Takagi-Sugeno [1985] model.

Höppner and Klawonn [2003] introduced a new combined distance function, which is the combination of both methods as follows:

$$d_{ik}^2((\boldsymbol{x}_k, y_k), (v_i, \hat{\beta}_i)) = \underbrace{\left\|\boldsymbol{x}_k - \upsilon_i(x)\right\|^2}_{FCM\ distance} + \underbrace{(y_k - \hat{\beta}_i^T \hat{\boldsymbol{x}}_k)^2}_{FCRM\ distance}$$

Where $\hat{x}$ represents a user defined polynomial, for instance , a two dimensional polynomial can be formed with the following vector: $(x_1, x_2) = (1, x_1, x_2, x_1x_2, x_1^2, x_2^2)$

The coefficients are obtained as:

$$\hat{\beta}_i = \left(\sum_{k=1}^{n} \left(\mu_{ik}\right)^m (y_k \hat{x}_k) \right) \Big/ \sum_{k=1}^{n} \left(\mu_{ik}\right)^m (\hat{x}_k \hat{x}_k^T), \forall\, i = 1, \ldots, c$$

8. Improved Fuzzy Clustering Algorithm (IFC)

We propose a new fuzzy clustering method by modifying standard FCM algorithm, called "***Improved Fuzzy Clustering (IFC)***" [Celikyilmaz, Turksen, 2007].

New objective function carries out two purposes:

(i) To find a good representation of the partition matrix; and

(ii) To find membership values which minimize the error of the ***Fuzzy Function*** models.

MİN $J_m^{IFC} = \sum_{i=1}^{c}\sum_{k=1}^{n}\left(\mu_{ik}^{imp}\right)^m d_{ik}^2 + \sum_{i=1}^{c}\sum_{k=1}^{n}\left(\mu_{ik}^{imp}\right)^m \left(y_k - h_i(\boldsymbol{\tau}_{ik}, \hat{w}_i)\right)^2$ $d^2 = \|x_k y_k - \upsilon_i(\boldsymbol{xy})\|^2$,

controls the precision of each input-output data vector.

9. Interim Fuzzy Functions

One can introduce a variety of interim fuzzy functions made of various powers and combinations of membership values:

As an Example: $\boldsymbol{\tau}_i = [\mu_i \quad \log((1-\mu_i^{imp})/\mu_i^{\text{imp}})]$

The set of planes in $\Re^2$ of each ith cluster is defined as :

$h_i = \hat{w}_{0i} + \hat{w}_{1i}\mu_i^{\text{imp}} + \hat{w}_{2i}\, log((1-\mu_i^{imp})/\mu_i^{imp})$, or $h_i = {}_i^T \hat{w}_i$, where $\hat{w}_i^T = [\hat{w}_{0i} \quad \hat{w}_{1i} \quad \hat{w}_{2i}]$ are the coefficients of the Interim Fuzzy Functions.

For example a particular set of fuzzy functions in $\Re^2$ is defined as:

$$\hat{y}_i = h_i(\boldsymbol{\tau}_i, \hat{w}_i) = \hat{w}_{0i} + \hat{w}_{1i}\mu_i^{imp} + \hat{w}_{2i} \log\left(\frac{1-\mu_i^{imp}}{\mu_i^{imp}}\right) = \hat{w}_{0i} + \sum_{j=1}^{2} \hat{w}_{ji}\tau_{ji}$$

For this purpose the distance function of the IFC algorithm is denoted by:

$$d_{ik}^{IFC} = \left\|\boldsymbol{z}_k - \upsilon_i(z)\right\|^2 + \left(y_k - h(\tau_{ik}, \hat{w}_i)\right)^2$$

For this purpose a solution can be found by:

$$\underset{m}{\text{Min}}\, L = \sum_{i=1}^{c}\sum_{k=1}^{n}\left(\mu_{ik}^{imp}\right)^{m} d_{ik}^{2} + \sum_{i=1}^{c}\sum_{k=1}^{n}\left(\mu_{ik}^{imp}\right)^{m}\left(y_k - h_i(\boldsymbol{\tau}_{ik}, \hat{w}_i)\right)^2 - \lambda\left(\left(\sum_{i=1}^{c}\left(\mu_{ik}^{imp}\right)\right) - 1\right)$$

Where λ is the Lagrange Multiplier. One takes the derivative of this objective function with respect to the cluster center and the membership values, and one obtains the optimum membership values:

$$\left(\mu_{ik}^{imp}\right)^{(t)}_{\substack{1<i,j\le c\\ 1\le k\le n}} = \left(\sum_{j=1}^{c}\left(\frac{\left(d_{ik}^{(t-1)}\right)^2 + \left(y_k - h_i(\tau_{ik}^{(t-1)}, \hat{w}_i)\right)^2}{\left(d_{jk}^{(t-1)}\right)^2 + \left(y_k - h_j(\tau_{jk}^{(t-1)}, \hat{w}_j)\right)^2}\right)^{1/(m-1)}\right)^{-1}$$

10. Experiments

We have experimented with a variety of system models shown in theTable 1 below.

Table 1. Overview of Datasets used in the experiments

No	Dataset	Type*	OBS**	#Var§	OBS used in three-way cross validation			Performs. Measure Used∂
					Training	Validation	Testing	
1	Friedmann Artificial	R	9,791	5	500	250	9,000	➢ RMSE ➢ MAPE ➢ Robust Simulated Trading Benchmark (RSTB) ➢ R^2 ➢ Ranking
2	Auto-Mileage–UCI	R	398	8	125	45	100	
3	Stock Price Predict. ➢ TID ➢ EMO ➢ Enbridge ➢ Loblaws ➢ Sun Life	R R R R R	389 445 445 445 445	16 16 16 16 16	120 200 200 200 200	90 144 144 144 144	100 100 100 100 100	
4	Dough Inoculation Process ➢ Reagent1 ➢ Reagent2	R R	10,000 10,000	11 11	250 250	750 750	8000 8000	
5	Liver Disorder–UCI	C	345	6	175	75	50	➢ Accuracy ➢ ROC Curve/ AUC ➢ Several Ranking Methods
6	Ionosphere-UCI	C	349	34	150	120	80	
7	Breast Cancer – UCI	C	277	9	130	70	50	
8	Diabetes – UCI	C	768	8	125	75	50	
9	Credit Scoring UCI	C	690	15	150	75	50	
10	California Housing	C	20,640	9	500	500	12,600	

* R: Regression, C: Classification Type Datasets.

** OBS: Total number of cases (i.e., instances, objects, data points, observations)

§ Var: Total Number of attributes/features/variables exists in the dataset.

∂ Performance measures used to evaluate each model performance in comparative analysis.

UCI: University of California, Irvine, Real Dataset Repository

11. Conclusions

We have drawn a number of conclutions out of our investigations.

A) We have implemented two clustering techniques in order to discover hidden structures embedded in a data set!!! That is we have shown that fuzzy clustering algorithms can identify hidden structures in a given data domain with the application of:

I) Fuzzy C-Means (FCM) clustering method [Bezdek, 1981]
II) Fuzzy C-Regression Clustering Method (FCRM) proposed by Hathaway and Bezdek [1993].
III) Improved Fuzzy Clustering (IFC) Method [Celikyilmaz, Turksen, 2007]

B) We have experimented with both:

I) Type-1 Fuzzy Functions and Improved Fuzzy Functions, and
II) Type-2 Fuzzy Functions and Improved Fuzzy Functions (with Further Developments)

It ought to be pointed out clearly that unique properties of the **Improved Fuzzy Functions** are:

a) The membership values obtained from improved fuzzy clustering algorithm and their transformations are used as additional predictors in identifying the local functions.
b) In addition to the two parameters, m^* and c^*, the proposed structure identification also requires improved fuzzy function types and structures to be defined.

Furthermore we have implemented two different strategies to identify system parameters.

I) The first one is the Type-I improved fuzzy functions (TIIFF) methods using an exhaustive search to identify the inference model parameters.
II) The second method is the evolutionary Type-I improved fuzzy functions (ETIIFF), which uses genetic algorithms to optimize the system parameters. The ETIIFF is computationally inexpensive since it requires less optimization steps compared to TIIFF.
III) Thus, one can easily reduce the exponentially growing search space to a manageable size with ET1IFF methods. With ET1IFF, the inference parameters are identified automatically, given the boundaries of the parameters.

References

[1] Aliev, R. A.: *Modeling and stability analysis in fuzzy economics*, Applied and Computational Mathematics. 7(1), (2008), pp. 31-53.
[2] Celikyilmaz, A., Thint, M.: *Semantic Approach to Textual Entailment for Question Answering – A New Domain for Uncertainty Modeling,* 7th IEEE Int. Conf. Cognitive Informatics (ICCI'08), Stanford University, CA, IEEE CS Press, (2008).
[3] Celikyilmaz, A., Turksen I. B.: *Type-2 Fuzzy Classifier Ensembles for Text Entailment,* Joint Conf. Information Sciences– Fuzzy Theory and Technology, Shenzhen, China, (2008).
[4] Celikyilmaz, A., Turksen, I. B.: *Enhanced Fuzzy System Models With Improved Fuzzy Clustering Algorithm,* IEEE Transactions on Fuzzy Systems 16(3), (2008), pp. 779-794.
[5] Celikyilmaz, A., Turksen, I. B.: *Fuzzy Decision Making With Imprecise Parameters,* Invited Paper, International Journal of Approximate Reasoning, (2009), accepted.
[6] Celikyilmaz, A., Turksen, I. B.: *Kernel Based Hybrid Fuzzy Clustering for Non-Linear Fuzzy Classifiers,* Proc. 27th Intern. Conf. of the NAFIPS, In IEEE Proceedings, (2009).

[7] Celikyilmaz, A., Turksen, I. B.: *Modeling Uncertainty with Fuzzy Logic,* Elsevier, (2009).
[8] Celikyilmaz, A., Turksen, I. B.: *Spectral Learning with Type-2 Fuzzy Numbers for Question/Answering System,* 2009 IFSA World Congress& EUSFLAT Conference, Portugal, (2009).
[9] Celikyilmaz, A., Turksen, I. B.: *Uncertainty Modeling of Improved Fuzzy Functions With Evolutionary Systems,* IEEE Trans. on Sys., Man & Cybern. 38(4), (2008), pp. 1098-1110.
[10] Celikyilmaz, A., Turksen, I. B.: *Validation Criteria for Enhanced Fuzzy Clustering,* Pattern Recognition Letters 29, (2008), pp. 97-108.
[11] Demirci, M.: *Fuzzy functions and their fundamental properties,* Fuzzy Sets and Systems 106, (1999), pp. 239-246.
[12] Dyson, R. G.: *Maxmin programming, fuzzy linear programming and multicriteria decision making,* Journal of Operating Research Society 31, (1980), pp. 263-267.
[13] Kacprzyk, J., Zadeh, L. A. (eds): *Computing with Words in Information/Intelligent Systems Part 2.,* Applications. Physica-Verlag, Heidelberg and New York, (1999).
[14] Mendel, J. M.: *An Architecture for Making Judgments Using Computing With Words,* Int. J. Appl. Math. Comput. Sci. 12(3), (2002), pp. 325-335.
[15] Mendel, J. M.: *Computing With Words and Its Relationships With Fuzzistics,* Information Sciences 177, (2007), pp. 988-1006.
[16] Ozkan, I., Turksen, I. B., Naci Canpolat: *A currency crisis and its perception with fuzzy C-means,* Information Sciences 178(8), (2008), pp. 1923-1934.
[17] Sasaki, M.: *Fuzzy Functions*, Fuzzy Sets and Systems, 55, (1993), pp. 295-301.
[18] Turksen, I. B.: *An Ontological and Epistemological Perspective of Fuzzy Set Theory,* Elsevier B. V., (2006).
[19] Turksen, I. B.: *Belief, plausibility, and probability measures on interval-valued type 2 fuzzy sets,* Int. Journal of Intelligent Systems. 19(7), (2004), pp. 681-699.
[20] Turksen, I. B.: *Fuzzy functions with LSE,* Appl. Soft Comput. 8(3) (2008), pp. 1178-1188.
[21] Turksen, I. B.: *Fuzzy System Models,* Encyclopedia of Complexity and Systems Science, (2009), pp. 4080-4094.
[22] Turksen, I. B.: *Meta-linguistic axioms as a foundation for computing with words,* Information Sciences 177(2), (2007), pp. 332-359
[23] Uncu, O., Turksen, I. B.: *Discrete Interval Type 2 Fuzzy System Models Using Uncertainty in Learning Parameters,* IEEE T. Fuzzy Systems 15(1), (2007), pp. 90-106.
[24] Wang, P.: *Computing with Words,* Albus J., et al. (eds), John Wiley & Sons, (2001).
[25] Zadeh, L. A.: *A new direction in AI – toward a computational theory of perceptions,* AI Magazine, Vol. 22, No. 1, (2001), pp. 73-84
[26] Zadeh, L. A.: *From computing with numbers to computing with words,* International Journal of Computer Science, Vol 12, No. 3, (2002), pp. 307-324.
[27] Zadeh, L. A.: Is there a need for fuzzy logic? Information Sciences 178(13), (2008), pp. 2751-2779.
[28] Zarandi, M. H. F., Turksen, I. B., Torabi Kasbi, O.: *Type-2 fuzzy modeling for desulphurization of steel process,* Expert Systems with Applications 32(1), (2007), pp. 157-171.

Evidence Based Approach for Sentence Extraction from Single Documents

Sukanya Manna, Tom Gedeon, B. Sumudu U. Mendis, Richard L. Jones

School of Computer Science, The Australian National University, ACT 0200, Canberra, Australia
{sukanya.manna, tom, sumudu, richard.jones}@cs.anu.edu.au

Abstract: We present an evidence based sentence extraction model which is an application of subjective logic in a document computing scenario, to rank sentences according to their importance in a document. Elements from the Dempster-Shafer belief theory are used by this model to measure the subjective belief or opinion about a sentence. The important sentences extracted by this model can be seen to summarize a document partially. For qualitative analysis, this method is compared with two different open source summarizers along with human extracted sentences which are used as benchmarks for this purpose. This model also improves the effect of signal to noise ratio on sentence rank by applying the whole evidence based model on a reduced data set to evaluate its stability and accuracy. Since evidence based models are computationally very expensive, here we show that one third of the words of a document are sufficient to rank sentences similarly to human judgements, but if reduced further, the accuracy drops. The results show that our evidence based model outperforms standard summarizers when evaluated with human ranked sentences.

Keywords: subjective logic, evidence theory, summarization, sentence extraction, uncertain probability, summarization

1. Introduction

Intelligence analysis is a complicated time critical task which requires attention and a high degree of analytical judgement under considerable uncertainty. It generally requires that analysts choose from several alternative hypotheses in order to present the most plausible of these as likely explanations or outcomes for the evidence being analyzed [18]. In a wider context, people make their decisions based on subjective information which is rarely completely certain and reliable. To handle such a scenario, such as analyzing a

single document, we require some form of subjective data analysis as there is no volume information available about the source data. In this paper, our main motivation is to show how subjective belief works on single documents for sentence classification and also to show the effect of reduced available information on the analysis. Our approach is to deduce relative sentence importance based on frequency plus inter-sentence interaction which is determined by subjective logic.

Standard logic deals with propositions which are either true or false. This is very unlikely to be useful in a human situation where a condition cannot be determined with absolute certainty whether that proposition is true or false. There are other alternative logics which handle uncertainty and ignorance and have been applied practically to solve problems where there is insufficient evidence [5], [12]. Probabilistic logic was defined by Nillson [13] with the aim of combining the capacity of deductive logic to exploit the structure and relationship of arguments and events, with the capacity of probability theory to express degrees of truth about those arguments and events. Belief theory represents the extension of classical probability by making explicit the expression of ignorance i.e., lack of information, by assigning belief mass to the whole state space [17]. Classical belief representation is quite general, and allows complex belief structures to be expressed on arbitrary large state spaces as seen in Dempster-Shafer theory, which addresses interaction based on the evidence [17]. The main idea behind belief theory is to abandon the additivity principle of probability theory. Instead, belief theory gives observers the ability to assign so-called belief mass to any subset of state spaces. A limitation of this model lies in the combination of evidence which may lead to counterintuitive conclusions after applying normalization [19]. To overcome this limitation, we use Jøsang's [6] model of subjective belief, as it has a simpler representation of belief functions called *'opinions'*, which can be easily mapped to probability density function.

Subjective logic [6] operates on subjective beliefs about the world and uses *opinion* to denote the representation of a subjective belief. An opinion can be interpreted as a probability measure containing secondary uncertainty, and thus subjective logic can be seen as an extension to both probability calculus and binary logic [7]. It can be seen that real world situations are more realistically interpreted and analyzed using this subjective logic when applied manually. An aim of our paper is to apply this subjective logic automatically to rank sentences from a document according to their importance. The concepts of belief and disbelief [6] have been incorporated to measure the uncertainty. The probability expectation of the sentences form the scores that rank the sentences by their importance with respect to their context.

In [6], subjective logic presented by Jøsang is used to model and analyze real world situations realistically; these real world situations are instances of open environments which have no specific limitations on evidence to be gathered for giving an opinion about

a hypothesis. On the other hand, in a document computing environment, the evidence collected is from the document which is a closed environment. A document consists of sentences, and each sentence consists of words. These form the basis for the evidence which we either get directly or by deriving from them. If there are n words in a document, then we have 2^n possible states (or combinations of words) which might exist in the whole document. The co-occurrence of words found in a sentence represents evidence which we derive from existing words. These represent non-atomic events belonging to 2^n. Now, we can consider belief of a sentence to refer to the total number of states present in that sentence i.e, words or combinations of words existing in the sentence represent evidence. This presents how much information we get from a sentence about the whole document. Disbelief of a sentence in this context can be stated as the words which are not present in that sentence as well as are not present in other sentences with which it has some words in common. This is 'ignorance' in Dempster-Shafer theory; so disbelief does not have any role in supporting a sentence or a hypothesis. Uncertainty of a sentence is the evidence which is plausible to support it. This means, if the sentence has some words in common with another, then the interaction of the words in the other sentence will have some contribution to the meaning of this sentence. In this context, interaction means all possible combinations of words which are present, either in the sentence being considered, or any other sentences in the document.

Arguments in subjective logic are called "subjective opinions" or "opinions" for short. An opinion can contain degrees of uncertainty in the sense of "uncertainty about probability estimates". The uncertainty of an opinion can be interpreted as ignorance about the truth of the relevant states, or as second order probability about the first order probabilities. Thus, opinion about a sentence presents the importance of that sentence in a given context containing degrees of uncertainty.

In this paper, besides finding the importance of sentences, we simultaneously reduce the complexity of the model by reducing information for analysis without significant loss of accuracy. It is known that most evidence based models are computationally expensive with the increase in size of the input space. We investigate the reduction (or purification) of the available information and its effect on sentence ranking using a reduced word set for the whole analysis. The quality of the top ranked sentence extracted is evaluated by comparing them with human ranked sentences of the same documents and also with open source summarizers.

The detailed implementation, modification and assumption of this application of the subjective logic based model are presented in sec. 2, followed by a detailed evaluation.

2. Modeling uncertain probabilities for sentence ranking

In this section we present the model for uncertain probabilities [6] for sentence ranking. The parameters of this model are defined using sentences in the form of hypotheses. Words (or terms) occurring in a sentence in a document are facts or evidence available to support or weaken the hypothesis. So the truth of evidence of the sentences is formulated using the given words or co-occurrence of words. Here *three basic assumptions* are made to proceed with this model:

1. All the words or terms (removing the stop words) in the document are atomic.
2. The sentences are unique, i.e., each of them occur only once in the given document.
3. It is a closed system where the evidence is confined within a single document.

A document consists of sentences. In this paper, a sentence is considered to be a set of words separated by a stop mark (".", "!", "?"). Non stop words are extracted and the frequencies (i.e. number of occurrences) of the words in each sentence are calculated.

Let us now define the notations which we will be using in the rest of the equations and explanations. Θ is the frame of discernment. We represent a document as a collection of words, which is

$$\Theta = D_w = \{w_1, w_2, ..., w_n\} \tag{1}$$

where, D_w is a document consisting of words w_1, $w_2 ... w_n$ and $|D_w| = n$. Now,

$$\rho(\Theta) = \{\{w_1\}, \{w_2\}, ..., \{w_1, w_2, w_3, ..., w_n\}\} \equiv 2^{\Theta} \tag{2}$$

$$|\rho(\Theta)| = 2^n, \tag{3}$$

where ρ represents the power set of the elements of Θ. We can also represent a document as a collection of sentences,

$$D_s = \{s_1, s_2, ..., s_m\} \tag{4}$$

where m is a finite integer and each s_i is an element of $\rho(\Theta)$. Each sentence is comprised of words, which belong to the whole word collection of the document D_w. We thus represent each sentence S_l by,

$$S_l = \{w_i, w_k, ..., w_r\} \in \Theta \tag{5}$$

where, $1 \leq i, k, r \leq n$ and $S_l \in \rho(\Theta)$.

The Belief Model The representation of uncertain probabilities [6] is based on a belief model similar to the one used in Dempster-Shafer theory of evidence. Initially a set of possible situations, frame of discernment are defined as in (1). It is assumed that the system cannot be in more than one elementary state at the same time. The elementary states in the frame of discernment Θ will be called atomic states because they do not contain substates.

Here, all the non stop words of the document are considered to be atomic and they are the elements of frame of discernment. The powerset of Θ, denoted by 2^{Θ}, contains the atomic states and all possible unions of the atomic states including Θ; this is the pattern of the words' occurrence or co-occurrence of words in the document. Sentences are events with non-atomic states. Similarly, co-occurrence of words represent sub-events (represented by non-atomic states as well). In this work, we actually support events using atomic and non-atomic states considering them to be sources of evidence within the document.

Suppose, we have a document D (fig.1) with 4 sentences, s_1, s_2, s_3, and s_4 and 5 words, w_1, w_2, w_3, w_4, and w_5 respectively. So the all possible states in the document will be 2^5 which are as follows:
$\{\emptyset, \{w_1\}, \{w_2\},..., \{w_1,w_2\}, \{w_2,w_3\},..., \{w_1,w_2,w_3\}, \{w_2,w_3,w_4\},..., \{w_1,w_2,w_3,w_4\},$
$... \{w_1,w_2,w_3,w_4,w_5\}\}$.
We consider only the ones which occur at least once in the document.

Now, the events with countable evidence are only considered for the calculations and a belief mass is assigned to each event.

Definition 1 (Belief Mass Assignment) *Let Θ be a frame of discernment. If with each substate $x \in 2^{\Theta}$ a number $m_{\Theta}(x)$ is associated such that:*
1. $m_{\Theta}(x) \geq 0$
2. $m_{\Theta}(\emptyset) = 0$
3. $\sum_{x \in 2^{\Theta}} m_{\Theta}(x) = 1$
then m_{Θ} is called a belief mass assignment in Θ, or BMA for short. For each substate $x \in 2^{\Theta}$, the number $m_{\Theta}(x)$ is called the belief mass of x.

Belief Mass Assignment (BMA) is defined here in the same way as modeled by Jøsang [6]. We also call this probability of evidence. Let Θ be a frame of discernment. If with each substate $x \in 2^{\Theta}$ a number m_{Θ} is associated such that:

1. $m_{\Theta}(x) \geq 0$
2. $m_{\Theta}(\Phi) = 0$
3. $\sum_{x \in 2^{\Theta}} m_{\Theta}(x) = 1$

In fig. 1, we depict the above example. The state space contains events that form evidence for the problem. Each of these words w_1, w_2, w_3, w_4, and w_5 are atomic states, and each of these sentences s_1, s_2, s_3 and s_4 are non atomic states (events). All these states are from power set of the frame of discernment. In this case, the possible states we get from the

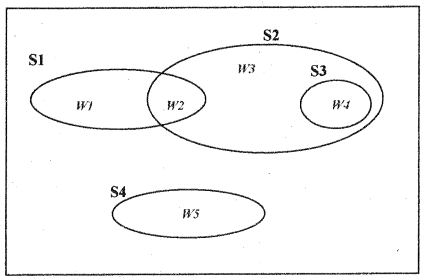


Figure 1: Example showing the occurrence of words in the sentences

example are:
$\{\{w_1\}, \{w_2\}, \{w_3\}, \{w_4\}, \{w_5\}, \{w_1,w_2\}, \{w_2,w_3\}, \{w_3,w_4\}, \{w_2,w_4\}, \{w_2,w_3,w_4\}\}$.

We calculate BMA for each event by,

$$m(x) = \frac{F(x)}{Z}, \tag{6}$$

where $F(x) = \sum_{k=1}^{N} f_{x_k}$, where N is the total number of sentences in the document, $x \in 2^{\Theta}$, and f_{x_k} is the frequency of occurrence of event x in sentence k. In words, it is the total frequency of that event in all the sentences (or the whole document).

$$Z = \sum_{\substack{\forall x \neq \emptyset \\ f_x \neq 0}} F(x), \quad x \in 2^{\Theta} \tag{7}$$

Z is the total frequency of the all the events which has valid evidence of truth (whose frequency is non zero). In this example, as shown in 1, let us assume that frequency of each of these words be 1 in each sentence. So, $Z = 12$. Now, we can see that s_1 has two words, w_1 and w_2. We have three different sub-states with non zero evidence; $m(w_1)$, $m(w_2)$, and $m(w_1,w_2) = m(s_1)$.

Definition 2 (Belief Function) *Let Θ be a frame of discernment, and let m_Θ be a BMA on Θ. Then the belief function corresponding with m_Θ is the function $b : 2^{\Theta} \rightarrow [0,1]$ defined by:*

$$b(x) = \sum_{y \subseteq x} m_\Theta(y), \quad x,y \in 2^{\Theta} \tag{8}$$

Now, in context to the example, we calculate the belief of a sentence, $b(s_1) = m(w_1) + m(w_2) + m(w_1,w_2)$. Similarly, an observer's disbelief must be interpreted as the total belief that a state is not true.

Definition 3 (Disbelief Function) *Let Θ be a frame of discernment, and let m_Θ be a BMA on Θ. Then the disbelief function corresponding with m_Θ is the function $d: 2^\Theta \rightarrow [0,1]$ defined by:*

$$d(x) = \sum_{y \cap x = \emptyset} m_\Theta(y), \quad x,y \in 2^\Theta. \tag{9}$$

If we now consider the example, we calculate disbelief of s_1 by $d(s_1) = m(w_3) + m(w_4) + m(w_3,w_4) + m(w_5)$.

Definition 4 (Uncertainty Function) *Let Θ be a frame of discernment, and let m_Θ be a BMA on Θ. Then the uncertainty function corresponding with m_Θ is the function $u: 2^\Theta \rightarrow [0,1]$ defined by:*

$$u(x) = \sum_{\substack{y \cap x \neq \emptyset \\ y \not\subseteq x}} m_\Theta(y), \quad x,y \in 2^\Theta. \tag{10}$$

From Josang's research concept, we can get **Belief Function Additivity** which is expressed as:

$$b(x) + d(x) + u(x) = 1, \quad x \in 2^\Theta, x \neq \emptyset. \tag{11}$$

One can simply calculate the uncertainty of a sentence by using (11), i.e., $u(s_1) = 1 - (b(s_1) + d(s_1))$.

Definition 5 (Relative Atomicity) *Let Θ be a frame of discernment and let $x,y \in 2^\Theta$. Then for any given $y \neq \emptyset$ the relative atomicity of x to y is the function $a: 2^\Theta \rightarrow [0,1]$ defined by:*

$$a(x/y) = \frac{|x \cap y|}{|y|}, \quad x,y \in 2^\Theta, y \neq \emptyset. \tag{12}$$

In this case, we get the following relative atomicity for sentence s_1 as:

$$a(s_1/w_1) = \frac{|s_1| \cap |w_1|}{|w_1|} = \frac{1}{1} = 1$$
$$a(s_1/w_2) = \frac{|s_1| \cap |w_2|}{|w_2|} = \frac{1}{1} = 1$$
$$a(s_1/\{w_1,w_2\}) = a(s_1,s_1) = \frac{|s_1| \cap |\{w_1,w_2\}|}{|\{w_1,w_2\}|} = \frac{2}{2} = 1$$
$$a(s_1/w_3) = \frac{|s_1| \cap |w_3|}{|w_3|} = \frac{0}{1} = 0$$
$$a(s_1/w_4) = a(s_1/s_3) = \frac{|s_1| \cap |w_4|}{|w_4|} = \frac{0}{1} = 0$$

$a(s_1/\{w_2,w_3\}) = \frac{|s_1|\cap|\{w_2,w_3\}|}{|\{w_2,w_3\}|} = \frac{1}{2}$
$a(s_1/\{w_2,w_3,w_4\}) = a(s_1/s_2) = \frac{|s_1|\cap|\{w_2,w_3,w_4\}|}{|\{w_2,w_3,w_4\}|} = \frac{1}{3}$
$a(s_1/w_5) = a(s_1/s_4) = \frac{|s_1|\cap|w_5|}{|w_5|} = \frac{0}{1} = 0$
Likewise, we calculate the atomicity for other sentences.

Definition 6 (Probability Expectation) *Let Θ be a frame of discernment with BMA m_Θ then the probability expectation function corresponding with m_Θ is the function $E: 2^\Theta \rightarrow [0,1]$ defined by:*

$$E(x) = \sum_y m_\Theta(y)a(x/y), \quad y \in 2^\Theta. \tag{13}$$

So, for the given example, we calculate *ProbExp* for sentence s_1 as follows:
$E(s_1) = m(w_1)a(s_1/w_1) + m(w_2)a(s_1/w_2) + m(\{w_1,w_2\})a(s_1/\{w_1,w_2\}) + ... + m(w_5)a(s_1/w_5)$

We calculate PE of each sentence using (13). We consider sentences to be important if they have higher probability expectation and lower uncertainty. By doing this, we have seen that sentences with higher PE and lower uncertainty, have more words interacting with other sentences.

Definition 7 (Opinion) *Let Θ be a binary frame of discernment with 2 atomic states x and $\neg x$, and let m_Θ be a BMA on Θ where $b(x)$, $d(x)$, $u(x)$, and $a(x)$ represent the belief, disbelief, uncertainty and relative atomicity functions on x in 2^Θ respectively. Then the opinion about x, denoted by w_x is the tuple defined by:*

$$w(x) \equiv (b(x), d(x), u(x), a(x)). \tag{14}$$

For compactness and simplicity of notation we will in the following denote belief, disbelief, uncertainty and relative atomicity functions as b_x, d_x, u_x and a_x respectively. Thus opinion about a sentence s_1 can be expressed using these four parameters as, $w(s_1) = (b(s_1), d(s_1), u(s_1), a(x))$.

3. Experiment

In this paper, Jøsang's subjective logic [6] is implemented in a document computing scenario to classify sentences in a document according to their importance. The motivation and context of the paper is different from that of [6], where the idea was to find opinion about an unknown event from the known ones at hand. But here, we are building evidence from a document to support an event based on its importance. The concept of evidence

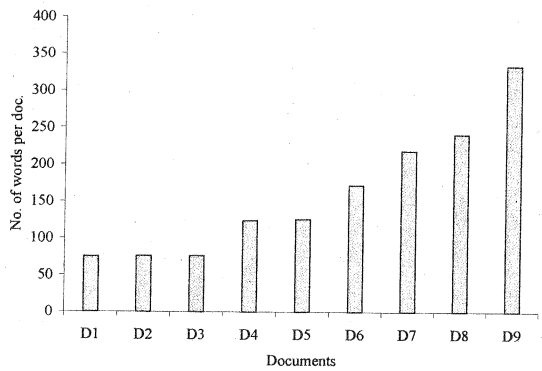


Figure 2: Number of words per document

is used in the form of word occurrence and its interaction with other words in a sentence and probability expectation of a sentence is calculated using (13) and (10) to calculate uncertainty. Sentences are then arranged according to descending probability expectation and ascending uncertainty to rank then according to their importance.

3.1. Data processing

The experiment is carried out using different Cross Document Structure Theory (CST) data sets [15]. Each data set consists of documents related to a specific topic such as plane crash, space shuttle mishap, and so on; consisting of fewer documents ranging from nine to ten or eleven. Our main aim is to see how this model works on single documents for content analysis purposes, so we focussed on this kind of data set unlike other information retrieval areas. Among the results, we present here are documents related to a Milan plane crash which consists of nine single documents. These documents were parsed, tokenized, cleaned, and stemmed. The cleaning is done by removing the stop words. The term list is generated from each of the documents. The documents are comparatively of moderate length as shown in fig.2.

3.2. Method

The subjective logic implementation of sentence classification has exponential time complexity. So, we prepared a reduced set of data to see the effects of the algorithms to extract sentences with them accurately. Two data sets were created for each documents; one with top 25 words and the other with every third word, together 25 in number. First, we found the frequency of occurrence of the words (excluding the stop-words) from a document. Then we arranged them in descending order of their occurrence and chose the top 25 words; as well as choosing every third element from the list keeping 25 here also. According to Gedeon *et al.*, [3] choosing every third word is appropriate as we suspect there is significant noise in the document so eliminating 2 out of 3 words can improve the model by increasing the signal to noise ratio so long as enough points remain to detect the underlying trend.

For shorter length documents, we kept approximately one third of the total number of words (approximately 25), then we gradually decreased to one sixth and then one ninth for the longer documents in order to maintain the count approximately to 25. It is seen that word count after the top 20 words have frequency of 1 in the word list for each documents. Statistically these words with count one will have similar contribution in the documents. So, this is another reason to analyze the effect of reduced word list for determining the sentence ranking.

We computed the probability expectation of each sentence and uncertainty using (13) and (10) respectively. We ranked each sentence with increasing probability expectation (PE) and decreasing uncertainty; the higher the PE and the lower the uncertainty, the greater is the importance of the sentence in the documents thus assigning higher rank to it.

Our main motivation is to extract important sentences from a document and use them for content analysis. To analyze the quality of the sentences extracted, we need some methods to proceed with the evaluation. So in the next phase, we perform the evaluation of this model where human accessors were involved to mark the important sentences which are used as a benchmark. This is then compared with two different open source summarizers MEAD [14] and OTS [1] respectively as also used by [1] for their evaluation.

3.3. Generation of Summaries

Summaries are broadly classified into text extraction and text abstraction [10], [8]. For text extraction, sentences from the documents are used as summaries and for text abstraction important pieces of information are extracted and then stitched together to form summaries

[1] `http://libots.sourceforge.net/`

following some linguistic rules. This evidence based model can be used as a text extraction as we use the original sentences like MEAD [14], an open source summarizer and OTS. Both these methods are used as a benchmark [1] for evaluation of summaries before.

Extraction of top ranked sentences play partially the role of summarization, so for qualitative analysis of our work, we compared the sentences extracted with open source tools such as MEAD and OTS.

Available summarizers *MEAD* [14] is a publicly available toolkit for multi-lingual summarization and evaluation. The toolkit implements multiple summarization algorithms (at arbitrary compression rates) such as position-based, Centroid, TF*IDF, and query-based methods. Methods for evaluating the quality of the summaries include co-selection (precision/recall, kappa, and relative utility) and content-based measures (cosine, word overlap, bigram overlap). We used single documents to summarize using MEAD.

The Open Text Summarizer (OTS) is an open source tool for summarizing texts. The program takes a text and decides which sentences are important and which are not. It ships with Ubuntu, Fedora and other linux distributions. OTS supports many (25+) languages which are configured in XML files. There is published researche on summarization, where OTS is used as a benchmark to evaluate the performance of summaries generated by their method [1], [16]. So we also used it for our comparative study for performance evaluation.

Human ranked sentences For the evaluation of our method, we involved two human assessors RJ and BP. We gave each of them the sets of documents. We asked them to rank 30% [2] of the important sentences as they read through the text. We then collected those sentences, and then formed extractive summaries maintaining the ranks assigned by them to the sentences.

Evidence based model (ProbExp): In sec.2, we described the methods of sentence ranking; subjective logic based where we ranked the sentences based on the probability expectation (ProbExp) and uncertainty of that sentence in that document. We ranked the sentences according to their importance by descending PE value and ascending uncertainty value. We took 30% [2] of the top ranked sentences. We consider these to represent the summary of the whole document. We then compared this top 30% with the summaries generated by the above summarizers (human as well as automated) to perform qualitative evaluation.

3.4. Evaluation

ROUGE evaluation ROUGE [9] stands for Recall-Oriented Understudy for Gisting Evaluation. It includes measures to automatically determine the quality of a summary by comparing it to other (ideal) summaries created by humans. ROUGE is a recall based metric for fixed length summaries. The measures count the number of over lapping units such as n-gram, word sequences, and word pairs between the computer-generated summary to be evaluated and the ideal summaries created by humans.

For this experiment, we used both machine generated summaries as well as human generated summaries to compare to our evidence based approach.

In this experiment, we present the result with ROUGE-1 (n-gram, where n=1) at 95% confidence level. ROUGE is sensitive to the length of the summaries [11]. The results vary when the length of the summaries of the peer and the model differs. We have shown here both kinds of results:
1. the usual convention of ROUGE by fixing the peer and model summary length to 100;
2. varying the length of the summaries, simply taking 30% of the top ranked sentences of a document as summary.

The figures 4 to 8 present the recall curve whereas the tables 1 to 4 present the average recall, precision and F-measure [9] of the comparisons of all documents.

Results In this part, summarization evaluation results are discussed. In the figures 3 to 8 and tables 1 to 4 some abbreviations are used, which are as follows:
$ProbExp_{top}$: Probability Expectation with top words (reduced word set)
$ProbExp_{\frac{1}{3}}$: Probability Expectation with every third word (reduced word set)
$Human_{BP}$: Refers to the human assessor BP
$Human_{RJ}$: Refers to the human assessor RJ

We mentioned that ROUGE is sensitive to the length of the summaries being compared. So, two different forms of ROUGE evaluation are presented: by limiting the word length of peer and model summaries to 100 (subsec.3.4.1); and without any word limitation (subsec.3.4.2). In both cases it is seen that the performance of our evidence based model is as good as human ranked sentences than automated summarizers: OTS and MEAD in particular. The results for $ProbExp_{\frac{1}{3}}$ show more consistent performance than $ProbExp_{top}$. When these two are compared with other automated summarizers like MEAD and OTS, $ProbExp_{top}$ results are more similar to these than $ProbExp_{\frac{1}{3}}$. This suggests that MEAD and OTS are more focussed on important words in a document. The tables 1 to 4 present the overall performance of each summarization method with human assessors' one. $ProbExp_{\frac{1}{3}}$ shows consistent performance in all the cases.

3.4.1 Evaluation by limiting the length of summaries to hundred words

Two different sets of results are shown in this section. The first is the comparison of the machine generated summaries with two different human assessors $Human_{BP}$ (fig.3) and $Human_{RJ}$ (fig.4). The second is the comparison of our evidence based method (with top words, $ProbExp_{top}$ and with every third, $ProbExp_{\frac{1}{3}}$) with MEAD and OTS (fig.5).

In this case, our evidence based model $ProbExp$ performs very similar to human generated summaries, $Human_{BP}$ and $Human_{RJ}$ than other automated summarizers, MEAD and OTS. The average performance $ProbExp_{\frac{1}{3}}$ is better than $ProbExp_{top}$, showing improvement of signal to noise ratio with reduction of words. Degradation of performance in our model is also noticed with excess removal of words. When $ProbExp$ is compared with MEAD and OTS, $ProbExp_{top}$ has higher similarity with them than $ProbExp_{\frac{1}{3}}$.

Comparison between evidence based model with human assessors We can see in fig.3 that $ProbExp_{\frac{1}{3}}$ and $ProbExp_{top}$ are more similar to human judgement given by $Human_{BP}$. Of the other two automated ones, MEAD and OTS, OTS performs better. It is closer to $ProbExp_{\frac{1}{3}}$ and $ProbExp_{top}$. Here the result for $ProbExp_{\frac{1}{3}}$ is better than $ProbExp_{top}$, since taking every third word increases the signal to noise ratio and purifies the ranking. Now, if we look at the length of the documents in fig.2, we find the number of words increasing with the documents. The performance of $ProbExp_{top}$ and $ProbExp_{\frac{1}{3}}$ initially outperformed the other two methods for the first three documents. The performance degraded from document 4 onwards. This is because the number of words per document started increasing from document 4 onwards (see fig.2), so in these documents approximately one sixth and one ninth of the words are considered for sentence ranking using our evidence based model. In this situation there is higher loss of useful information with greater reduction of words. But still it outperforms MEAD except for the last document, where we lost the maximum amount of information while reducing words to one ninth.

Fig.4 shows a similar effect to that seen in fig.3. $ProbExp_{top}$ as well $ProbExp_{\frac{1}{3}}$ is very close to human $Human_{RJ}$. Here again the drop in performance by our method is noticed for the documents with higher number of words. From these two figures we can say that if we consider one third of the total number of words, this evidence based model can give us close results to human judgement.

Comparison between evidence based model with automated summarizers In fig.5, $ProbExp_{top}$ and OTS have higher overlap than $ProbExp_{\frac{1}{3}}$ and OTS. $ProbExp_{top}$ and MEAD comes next to $ProbExp_{top}$ and OTS in terms of performance. Here too the performance degradation is observed as seen in figures 3 and 4.

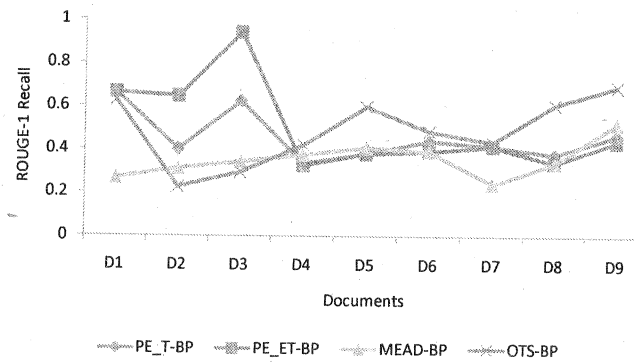


Figure 3: ROUGE-1 recall for Assessor BP with different automated methods

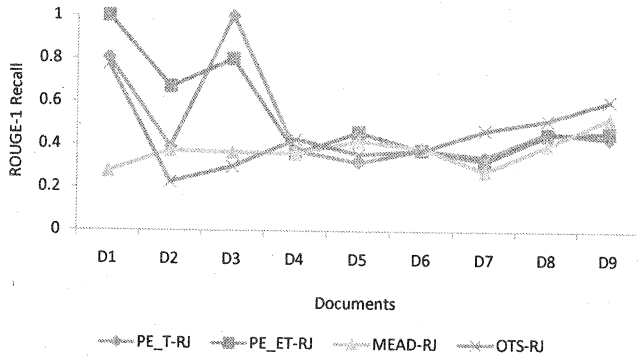


Figure 4: ROUGE-1 recall for Assessor RJ with different automated methods

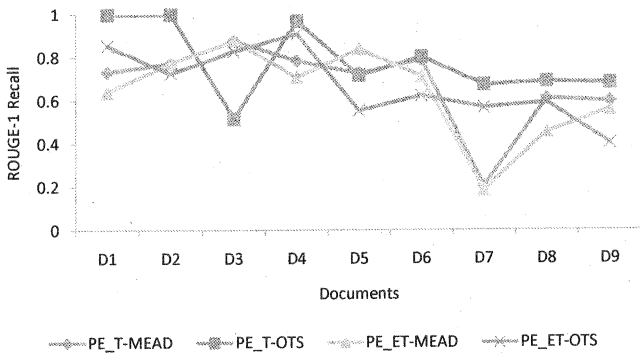


Figure 5: ROUGE-1 recall for evidence based model (PE) with automated summarizers

Table 1: Average ROUGE-1 (word limit=100) Recall(R), Precision(P) and F-measure(F) of all documents by comparing $Human_{BP}$ with different automatic summarization methods

	Avg R	Avg P	Avg F
$ProbExp_{top}$	0.46	0.50	0.47
$ProbExp_{\frac{1}{3}}$	0.51	0.51	0.51
MEAD	0.36	0.40	0.37
OTS	0.49	0.60	0.53

It should be noticed that using only one third of the words, our evidence based model works better than standard summarizers when compared with human assessors. The performance is boosted when we increase the signal to noise ratio by taking every third word from the document word list. But, summarizers like OTS and MEAD are more significant keyword focussed, so fig.5 shows our $ProbExp_{top}$ has higher similarity with them than $ProbExp_{\frac{1}{3}}$.

Tables 1 and 2 present the average recall, precision, and F-measure of all the documents for both human assessors' summaries with automated ones when ROUGE parameter is

Table 2: Average ROUGE-1 (word limit=100) Recall(R), Precision(P) and F-measure(F) of all documents by comparing $Human_{RJ}$ with different automatic summarization methods

	Avg R	Avg P	Avg F
$ProbExp_{top}$	0.50	0.49	0.49
$ProbExp_{\frac{1}{3}}$	0.55	0.48	0.51
MEAD	0.38	0.39	0.38
OTS	0.45	0.50	0.47

fixed to 100 words for evaluation. In both the tables similar results are noticed. As shown in the figures 3 and 4, here too in the tables, summaries generated by $ProbExp_{\frac{1}{3}}$ are most similar to humans than $ProbExp_{top}$ and then followed by OTS and MEAD.

3.4.2 Evaluation without any specific word limit

In this part, we present ROUGE evaluation results without limiting the lengths of model and peer summaries. The summaries are are 30% of the total length of a document. We found that ROUGE score tends to increase with increases in the length of the summaries. Like the previous evaluations for fixed length summaries, here the same comparisons are presented without fixing the length. The comparison results show that our evidence based models behave more similarly with human assessors' than MEAD and OTS. $ProbExp_{\frac{1}{3}}$ is even better than $ProbExp_{top}$ (like subsec.3.4.1). Similar degradation of performance is noticed here like the fixed summary length evaluation results (see subsec.3.4.1). In the tables 3 and 4, similar results are observed when averaged over all documents.

Comparison between evidence based model with human assessors In fig,6, $ProbExp_{\frac{1}{3}}$ and $ProbExp_{top}$ are more similar to human assessor BP in terms of overlap than the other two automated standard summarizers. Though there is performance degradation due to reduced data size to one sixth and then to one ninth for the documents with ascending word length, still $ProbExp_{top}$ and $ProbExp_{\frac{1}{3}}$ outperforms others. $ProbExp_{\frac{1}{3}}$ is best among all. The reason for this case is the same as in the other figures 3 to 5, due to purification of words increasing the signal to noise ratio.

In fig.7 similar behaviour is observed like fig.6. For majority of the documents, $ProbExp_{\frac{1}{3}}$ is higher than the other models and has maximum overlap with $Human_{RJ}$.

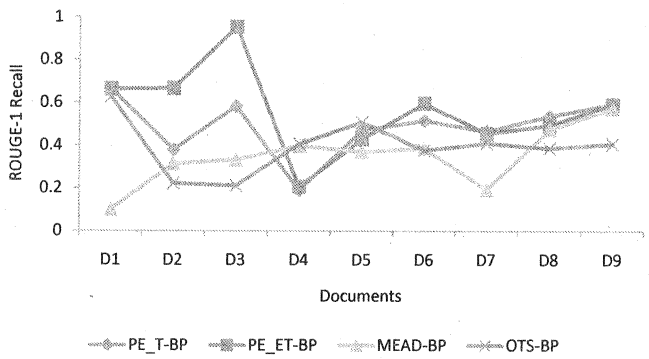


Figure 6: ROUGE-1 recall for Assessor BP with different automated methods

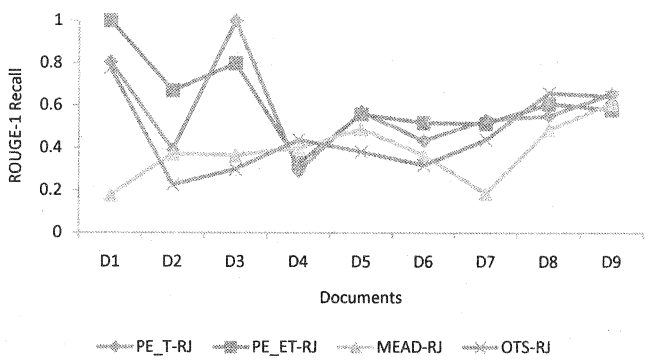


Figure 7: ROUGE-1 recall for Assessor RJ with different automated methods

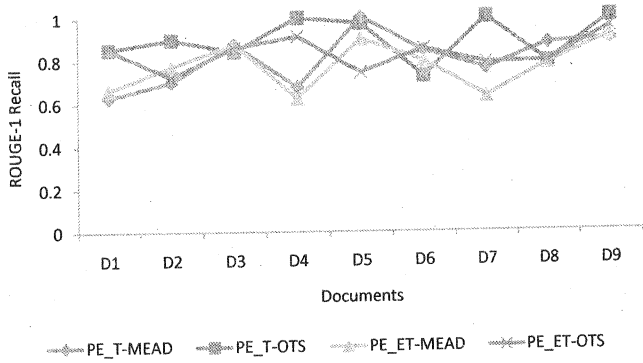


Figure 8: ROUGE-1 recall for evidence based model (PE) with automated summarizers

Table 3: Average ROUGE-1 (without specific word limit) Recall(R), Precision(P) and F-measure(F) of all documents by comparing $Human_{BP}$ with different automatic summarization methods

	Avg R	Avg P	Avg F
$ProbExp_{top}$	0.49	0.50	0.47
$ProbExp_{\frac{1}{3}}$	0.56	0.53	0.53
$MEAD$	0.35	0.46	0.38
OTS	0.40	0.64	0.48

Comparison between evidence based model with automated summarizers In fig.8, the picture is a bit different. $ProbExp_{top}$ has maximum overlap with OTS than MEAD. $ProbExp_{\frac{1}{3}}$ comes next in terms of overlap with OTS than MEAD. Here the performance degradation of $ProbExp_{top}$ and $ProbExp_{\frac{1}{3}}$ is not obvious.

Like tables 1 and 2, tables 3 and 4 present the average recall, precision and F-measure of all the documents for both human assessors' ($Human_{BP}$ and $Human_{RJ}$) generated summaries with automated ones without limiting word limits on ROUGE parameter. In both the tables, $ProbExp_{\frac{1}{3}}$ results are consistent for and similar to human assessors. $ProbExp_{top}$ is next to this. But overall performance of our evidence based model is higher than OTS and MEAD.

Table 4: Average ROUGE-1 (without specific word limit) Recall(R), Precision(P) and F-measure(F) of all documents by comparing $Human_{RJ}$ with different automatic summarization methods

	Avg R	Avg P	Avg F
$ProbExp_{top}$	0.64	0.45	0.51
$ProbExp_{\frac{1}{3}}$	0.63	0.44	0.50
MEAD	0.39	0.38	0.36
OTS	0.47	0.54	0.49

4. Conclusions

In this work we presented a subjective belief model for ranking sentences according to their importance from a single document. This evidence based model uses interaction and word occurrence among sentences. We presented the effect of a reduced word set for the evidence based model on sentence extraction. One of our hypotheses is supported by the results which show that a reduced filtered (or purified) data set can increases the signal to noise ratio, and can be used for extraction of significant sentences for summarization which are almost as good as human analysis. Another observation from this experiment is that the summaries generated by the top word set closely resemble the standard summarizers rather than the word set having every third word; but the word sets with every third word resembles more closely the human annotated results. This suggests that machine summarizers are too focussed on the *important* words, while human summarizers may be focusing elsewhere, which is likely to be the *content* or the *meaning*. The illustrations in the experimental result section also showed that ROUGE score for $ProbExp_{top}$ and $ProbExp_{\frac{1}{3}}$ is consistently higher for all the documents having fewer words, where we have considered one third of total words. But performance started degrading with the increase in the number of words in the documents where we increased the reduction ratio to one sixth and one ninth respectively. But it is interesting to notice that with only few words, it is still possible to rank the sentences meaningfully according to their importance closely matching with human judgements. Overall, it is clear from the experimental results that PE, the evidence based model, though having higher complexity, is effective and consistent in sentence extraction and summarization and outperforms the other standard summarizers with only one third of the words; thus reducing complexity to a greater extent.

The high complexity of belief based model is also the issue that we encountered. Though we have obtained good performance with our algorithm with a reduced word set, we are

working on further improvements. There are some methods to reduce the computational complexity of the state space which is similarly used in fuzzy measures, called the K-additivity method [4]. We also aim to improve the effectiveness (or accuracy) of the belief based model by data filtration (or reduction), some initial results have already been shown here in the form of reduced data sets in our experiments on significant sentence extraction and summarization.

References

[1] O. Boydell and B. Smyth. From social bookmarking to social summarization: an experiment in community-based summary generation. In *Proceedings of the 12th international conference on Intelligent user interfaces*, page 51. ACM, 2007.

[2] H. Dalianis. SweSum-A Text Summarizer for Swedish http://www. dsv. su. se/% 7Ehercules/papers. *Textsumsummary. html*, 2000.

[3] T. D. Gedeon and T. G. Bowden. Heuristic pattern reduction. *International Joint Conference on Neural Networks*, pages 449–453, 1992.

[4] M. Grabisch. k-order additive discrete fuzzy measures and their representation. *Fuzzy Sets and Systems*, 92(2):167–189, 1997.

[5] A. Hunter. *Uncertainty in information systems*. Mc-Graw Hill, London, 1996.

[6] A. Jøsang. A Logic for Uncertain Probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–311, 2001.

[7] A. Jøsang. Probabilistic logic under uncertainty. *Proceedings of the thirteenth Australasian symposium on Theory of computing-Volume 65*, pages 101–110, 2007.

[8] Elizabeth DuRoss Liddy. The discourse-level structure of empirical abstracts: an exploratory study. *Inf. Process. Manage.*, 27(1):55–81, 1991.

[9] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[10] C.Y. Lin and E. Hovy. Identifying topics by position. In *Proceedings of the fifth conference on Applied natural language processing*, pages 283–290. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1997.

[11] C.Y. Lin and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, page 78. Association for Computational Linguistics, 2003.

[12] A. Motro and P. Smets. *Uncertainty Management in Information Systems: From Needs to Solutions*. Kluwer Academic Pub, 1997.

[13] N. J. Nilsson. Probabilistic Logic. *Artificial Intelligence*, 28(1):71–87, 1986.

[14] D. Radev, T. Allison, S. Blair-Goldensohn, J. Blitzer, A. Çelebi, S. Dimitrov, E. Drabek, A. Hakim, W. Lam, D. Liu, et al. MEAD-a platform for multidocument multilingual text summarization. *Proceedings of LREC*, 2004, 2004.

[15] D. Radev, J. Otterbacher, and Z. Zhang. CST Bank: A Corpus for the Study of Cross-document Structural Relationships. In *Proceedings of LREC 2004*, 2004.

[16] Lawrence H. Reeve, Hyoil Han, and Ari D. Brooks. The use of domain-specific concepts in biomedical text summarization. *Inf. Process. Manage.*, 43(6):1765–1776, 2007.

[17] G. Shafer. *A mathematical theory of evidence*. Princeton University Press Princeton, NJ, 1976.

[18] M. Warner. Wanted: A Definition of Intelligence. *Studies in Intelligence*, 46(3):15–22, 2002.

[19] L. A. Zadeh. Reviews of Books. *AI Magazine*, 5(3):81–83.

Using Neural Gas Networks in Traffic Navigation

Ján Vaščák

Centre for Intelligent Technologies, Technical University in Košice,
Letná 9, 042 00 Košice, Slovakia,
e-mail: jan.vascak@tuke.sk

Abstract: The quality of navigation methods for mobile means depends first of all on description accuracy of their movement in a given area. This paper deals with Neural Gas (NG) networks whose role is creating of topologies for complex objects as for instance road networks of municipal communications where it is necessary to determine relations among individual elements (in our case a network of mutually interconnected communication nodes), too. The proposed approach combines besides NG networks also tree search algorithms, namely A*, hereby enabling to consider also various restraints, e.g. traffic rules, too. The performance of the proposed algorithm is shown on the road network of the city Košice in Slovakia.

*Keywords: navigation, Neural Gas networks, tree search algorithms, algorithm A**

1. Introduction

At present there are three basic groups of navigation methods for mobile means: heuristic, grid and exact algorithms [1]. A typical representative of the first group there are Bug algorithms. They are simple but suitable only for avoiding only a smaller number of obstacles, mainly in tasks for preventing immediate collisions. Potential fields [15, 18] are the most known approaches of the grid algorithms. However, they require not only creating a potential field of the whole area in advance but in the case of some changes (e.g. a new obstacle) it will be necessary again newly to create the whole field and computational efforts are considerably high. The exact algorithms as visibility graphs or Voronoi diagrams enable to find the best (in our case the shortest) trajectory. If there is no solution they will be even able to terminate the computation and so to prevent timeless sub-cycling. Opposite to the potential fields in the case of changes it will be necessary to do modifications only in the given area. However, their drawback is that they require very accurate data about obstacles that means a serious problem in practical applications.

The NG networks are basically graphs, which enable modelling the form of given patterns, similarly as in Kohonen networks (Self-organizing Maps) but opposite to them NG networks do not have any definite topology of connections in the output layer. This fact seems to be very advantageous mainly in the case of non-homogenous areas. Their ability of accurate description for a given pattern was compared to other kinds of neural networks and confirmed e.g. in [6, 12, 21]. Therefore it seems to be very suitable to

utilize them just for the description of road networks with various types and forms of communications. Besides, the use of some modified search algorithms enables us to incorporate into the description various restraints, e.g. traffic rules or traffic density, too. In addition, convenient graph structure of NG networks enables very efficient searching.

This paper deals with description of NG networks principles, followed by description of the traffic networks problem and modification of the A* algorithm. In concluding parts some experiments will be analysed, which were done on a real road network of the city Košice in Slovakia.

2. Principles of NG Networks

The NG networks are basically derived from Kohonen networks where for the position change determination of output neurons also the neighbourhood principle is used. However, opposite to them the neighbourhood is changed in each adaptation step according to given input [9]. This enhanced adaptation measure enables a relatively free movement of neurons in the area, which should be described just by their suitable deployment and it represents analogy to gas spreading in a closed space. Learning of NG networks mutually combines two learning paradigms – *own NG learning* and *competitive Hebbian learning*.

Let us suppose that we have a predetermined number of points N with help of them we can describe a given space. They will be denoted as *reference vectors w* whose elements are in general space coordinates – in our case they will be two-dimensional. Reference vectors w divide the area (space) into N parts and in each of them they represent centres of these parts. Such an approach is known as *vector quantization* that represents certain form of coding with help of which we can describe given area and which is the own task of NG networks. The success of this task is dependent just on suitable deployment of reference vectors. Therefore, similarly as in Kohonen networks, in individual cycles we will select points ξ belonging to this area and with their help we will adapt positions of reference vectors, which define positions of neurons c_i of the network output layer A, i.e. $A=\{c_1, c_2, \ldots, c_N\}$.

The object of the *own learning* is adaptation of reference vectors, i.e. calculation of their change $\Delta w_i(t)$ in the time t. It is a kind of the multiple-winners learning where the most change is at the winner vector w_{s1} but in accordance with the *neighbourhood range h(t,k)* also other k neighbours are changed although in a lower measure. The vector w_{s1} is assigned to such a point, which is the nearest to the input ξ, i.e. $\arg(\min(\xi - w_{S1}))$. The ambition is to reach bigger changes in a broader range at the beginning of the learning process and continuing with time this trend would decrease till the adaptation end time t_{max}. From this reason parameters λ_p and λ_z are defined – starting and final where $\lambda_p \square \lambda_z$. The parameter λ for the given time t is defined as:

$$\lambda(t) = \lambda_p (\lambda_z / \lambda_p)^{t/t_{\max}} \tag{1}$$

and the *neighbourhood range h(t, k)* for the k^{th} neighbour is computed as:

$$h(t,k) = \exp\left(\frac{1-k}{\lambda(t)}\right). \tag{2}$$

The adaptation process is also influenced by the learning parameter γ whose value is time-dependent according to γ_p and γ_z similarly as for λ and the calculation is like in (1).

The entire adaptation process in individual adaptation steps till t_{max} is following:

1. Sequencing all reference vectors by their distance to ξ into series:

$$\|\xi - w_{S1}\| \leq \|\xi - w_{S2}\| \leq \ldots \leq \|\xi - w_{SN}\|. \tag{3}$$

2. Adapting all reference vectors by:

$$\Delta w_{Si}(t) = y(t) \cdot h(t,k) \cdot (\xi - w_{Si}). \tag{4}$$

3. Setting up the time $t = t+1$ and until $t < t_{max}$ return to the step 1 else finish the adaptation.

It is possible to prove that this kind of adaptation principally corresponds to the optimization of a cost function according to the gradient descent method [10], which does not exist in Kohonen networks. In other words, the adaptation in NG networks is usually quicker.

The *competitive Hebbian learning* [9] serves for constructing a topological structure among neurons of the output layer. This kind of learning comes from the basic idea of Hebbian learning, i.e. that the connections whose neurons are activated at the same time (synchronously) are strengthened. Their change corresponds to the product of the activation values of these neurons. However, at the same time a competition element is embedded, too. This means in one adaptation step only one connection is created, namely between the two closest network points to the input ξ, i.e. between the reference vectors w_{s1} and w_{s2} of the points *S1* and *S2* (neurons c_1 and c_2). The distance among points is usually calculated by Euclidian norm. In [11] it was shown that the topology created in such a manner corresponds to Delaunay triangulation. Consequently, after transforming to Voronoi regions, these regions ensure finding an optimal path.

The simplest way for learning NG networks is using a two-stage process where at first suitable deployment of a given number of points is created and then they are mutually interconnected using competitive Hebbian learning. However, this approach is possible only in the case of predefined t_{max}, which is a considerably limiting circumstance. The other possibility is based on parallel processing of both learning kinds. However, there is a danger that Delaunay triangulation will be damaged due to continued changes of reference vectors. From this reason it is necessary to incorporate a mechanism of removing obsolete connections, too. For this purpose a process of *aging connections* is used where an age $v(c_1, c_2)$ is assigned to each connection between the neurons c_1 and c_2. At the moment of creating a new connection its age is set up to zero as well as in the case if the algorithm tries again to create this connection (the so-called *connection rejuvenating*). The age will be incremented by 1 in all connections among all direct neighbours of c_1 if it becomes again the winner. If the age of a connection reaches the value $T(t)$ it will be removed. In such a way a risk of invalid connections will be minimized. As at the adaptation start bigger changes are done it is necessary for $T(t)$ to be changed in time from smaller to bigger values. It is determined in a similar way as in (1) where $T_p \square T_z$.

A learning process by adaptation steps for obtaining a description of a ring (grey surface) [4] is depicted in Fig. 1. It is possible to observe the influence of parameters (λ_p= *10,* λ_z = *0.01,* γ_p = *0.5,* γ_z = *0.005,* t_{max} = *40000,* T_p = *20,* T_z = *200*) with advancing time, too. From intelligibility reasons in the depictions b – e the connections among neurons are missing.

In the first steps of learning a large neighborhood of the input ξ contains adapted points, which is characterised by a massive deployment of a large number of points in a form similar to the given space. Later, the influence of adaptation parameters *h(t, k)* and *γ(t)* will be weakened and thereby the adaptation will be performed only in close surroundings of the input ξ, i.e. on a local level individual reference vectors try to cover the given space uniformly. Such a phenomenon is typical for various forms of described space [3].

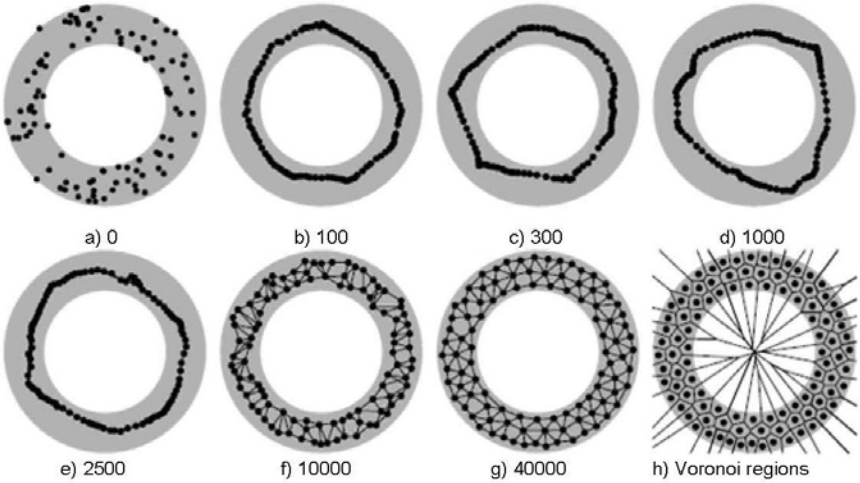


Figure 1. Learning process of a NG network with competitive Hebbian learning on a ring indicating the number of steps

2.1. Growing NG Networks

A considerable limitation of the previous approach is based on a fact that it is necessary to determine a fix number of output neurons in advance whose estimation can be difficult (if at all possible). Besides, further required property could be embedding a quality criterion directly as an ending condition for learning. These requirements led to a design of modified NG networks with incremental learning named as *Growing NG (GNG) networks* [2] using already mentioned algorithms.

A GNG network starts learning with two starting neurons. For growing their number as well as their deployment it uses heuristics and the so-called *quantization error* E_c of individual neurons c_i where squared errors of the vector w_s relating to the input ξ will be accumulated, i.e.:

$$E_c = \sum \|\xi - w_S\|^2 \tag{5}$$

if this neuron is a winner, i.e. a neuron with the biggest error (further point *S1*). The learning task is to perform the space quantization in such a way to obtain the minimum total quantization error *E*, i.e. $\sum E_c$ for $i = 1, \ldots, N$. From experimental experience (heuristics) it can be mostly supposed that inserting further neurons will reduce the error *E* and the best result will be obtained if the new point *r* is 'close' to *S1*. For 'closeness' determination there is another heuristic based on using point *S2*, which is a direct neighbour having the biggest E_c comparing to another ones. A new reference vector w_r will be then placed between *S1* and *S2*:

$$w_r = \left(w_{S1} + w_{S2}\right)/2. \tag{6}$$

Consequently, the connection between *S1* and *S2* will be removed and new two ones will be created between *S1* and *r* as well as *S2* and *r*. Simultaneously, the errors of points *S1* and *S2* will be reduced by values $\alpha.E_{c1}$ and $\alpha.E_{c2}$ where α is the quantization error reduction parameter. As it is clear the point *r* will not fully reduce the total quantization error analogically to (6) a certain amount of starting error will be assigned to *r*, i.e. $E_r = (E_{c1} + E_{c2})/2$.

Unlike NG networks only the reference vectors of *S1* and its direct neighbours are adapted. We can consider only local adaptation with uniform growing of output neurons as it is visible in Fig. 2 and it seems to be also a certain advantage from the point of view of possible simpler learning process analysis. As the adaptation of reference vectors is an iterative process inserting new points will be done only in adaptation steps being a natural product of the parameter τ. Further differences to NG networks are timely constant learning parameters for *S1* and its direct neighbours S_i, i.e. γ_{S1} and γ_{Si} as well as the maximum age of a connection *T*. It is supposed in each adaptation step certain improvement will occur and therefore the quantization error E_{ci} will be always reduced by the value $\beta.E_{ci}$.

The whole learning process of GNG networks is as follows:

1. During initialization two starting neurons are selected arbitrarily. The connection set is empty.
2. An input signal ξ enters the system and the point *S1* with the biggest error E_{c1} from all $c_i \in A$ will be obtained. Consequently, the point *S2* as a direct neighbour of *S1* will be determined. The points *S1* and *S2* will be connected and the connection age will be set up to zero. If the connection already exists then its age will be set up again to zero.

For c_1 the equation (5) will be used and reference vectors for *S1* as well as its direct neighbours will be adapted by:

$$\Delta w_{Si} = \gamma_i \cdot \left(\xi - w_{Si}\right) \tag{7}$$

for $i = 1, 2, \ldots$ and the age of their connections will be incremented.

3. If a connection reaches the age *T* it will be removed as well as all points without any connections.

4. If the adaptation step reaches a natural product of τ (if no then next step 6) a new point *r* (6) will be inserted and connections and errors for *S1*, *S2* and *r* will be modified.
5. For each neuron c_i the quantization error E_{ci} is reduced by the value $\beta.E_{ci}$.
6. If the ending condition is not yet fulfilled (e.g. maximum network dimension or minimum error) then continue next adaptation step and go to the step 2.

In Fig. 2 there is depicted learning process of a GNG network by individual adaptation steps again on the same example of a ring (see Fig. 1) [4] ($\tau = 300$, $\gamma_{S1} = 0.05$, $\gamma_{Si} = 0.0006$, $\alpha = 0.5$, $\beta = 0.0005$, $T = 88$, $N = 100$). Comparing Fig. 1 and Fig. 2 it is possible to see differences of learning NG and GNG networks. However, the obtained results are almost identical (see Fig. h).

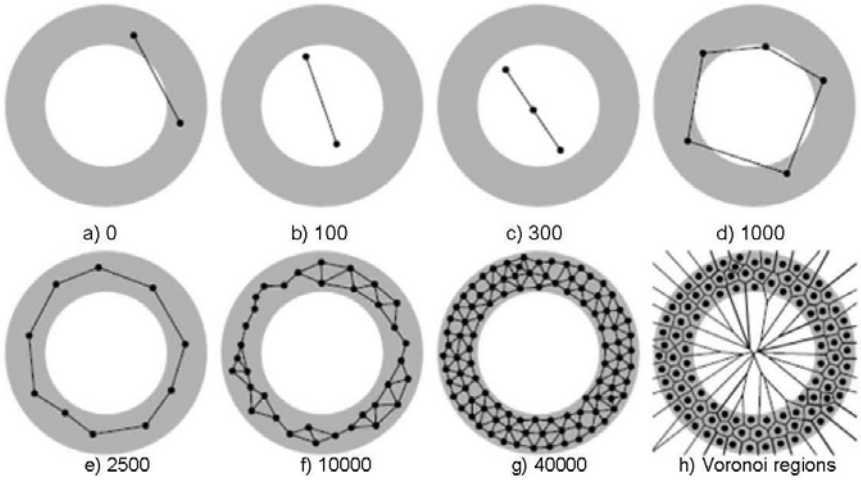


Figure 2. Learning process of a GNG network on a ring indicating the number of steps

3. Utilization and Modification of NG Networks for Purposes of Path Planning

Still nowadays data for traffic navigation systems are created by a complicated way with considerable portion of manual work either in preparing maps or directly on place in measuring orientation points using GPS. In our case it is possible to almost fully automate this preparation stage (besides inserting data about one-way roads and other entry restrictions). Using a colour filter only communications are extracted from the map (Fig. 3) and this kind of information will be a direct basis (training set) for learning. For our purposes we used a GNG network and experiments were done on the map of city Košice.

In Fig. 3 there is a part of a primarily learned network (blue colour) together with the real state of communications (black colour). We can see that this network partially:

- connects communications without any real connection, which is wrong,

- is redundant, i.e. it contains too many points, which can be omitted and thereby we can get a simpler network structure.

Figure 3. A part of a primarily learned GNG network

During removing wrong connections it is necessary to distinguish between really incorrect connections as e.g. the connections *a, b* in Fig. 4 and principally acceptable connections *c, d*. Acceptable connections represent a principal existence of a road only they are not able accurately to describe its form. From this reason additional points will be inserted better to form it. For separating these two cases a new modification of the Bug2 algorithm [8] was proposed [16] where a search oval is created with the radius ρ covering the investigated connection (see Fig. 4). If in this oval a continuous connection exists between the end points of the investigated connection then the connection is acceptable else it will be removed.

Further step is removing redundant connections. In Fig. 3 it can be seen straightforward roads are described not by only one connection but by several shorter connections, too. All intermediate connections are dispensable because they can be substituted by a longer one, which causes lower memory efforts and shortening the path search. Usually, using reduction mechanisms a considerably simpler network is obtained.

Since, in a traffic network there are both bidirectional and one-way roads the network from Fig. 3 will be doubled and connections will be given orientations. Consequently, for one-way communications (including rotaries) the prohibited directions will be removed. Only this stage requires manual activity. Finally, the connections will be given the information about their length, which is in other words the path cost. The last two kinds of information, i.e. orientation and path cost are fundamental for path search algorithms.

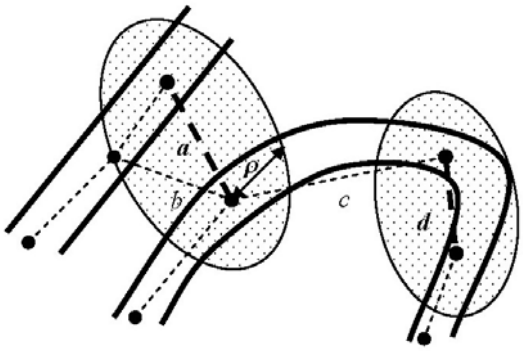


Figure 4. Identification of wrong connections

Creating a description of the road network in the form of a tree structure using NG networks represents only the first stage in the navigation task. Next stage is preformed by tree search algorithms utilizing the structure of a NG network as a convenient data source where they solve the task to find the best path between two points. For this reason the A^* algorithm was used being able to find the shortest path with a minimum number of browsing and in the case of its absence also being able to give a message [8].

However, it is necessary still to take some modifications of the A^* algorithm [19] to keep traffic rules, namely:

- prohibition of turning in a node,
- the so-called P-problem.

Since in the bidirectional communications there is each connection doubled with reverse orientations this would enable turning in next node (point), which means turning in a road. Therefore, being used the current connection its reverse orientation is temporarily cancelled.

As seen from Fig. 5 the so-called P-problem resembles to the character P. There is a problem of inability to find a path although it exists indeed. Let us suppose a car is on the connection between the points 1 and 2 and tries again to come back to the point 1. In such a case the algorithm will stop although there is a solution in the form $2\rightarrow3\rightarrow4\rightarrow5\rightarrow2\rightarrow1$ (principles of A^* as well as its modifications are more detailed described in [20]). To prevent this problem the coding of the whole NG network was changed in such a way the path will not be searched by the points but by their connections [19]. In other words, neurons of the NG network will not represent communication points (crossings and curves) but connections between these points. In our case for Fig. 5 the solution will look like $23\rightarrow34\rightarrow45\rightarrow52\rightarrow21$.

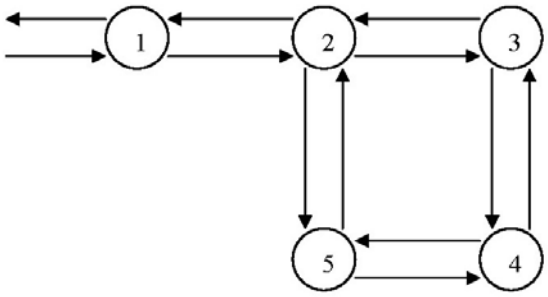


Figure 5. Description of the P-problem

4. Setting up Learning Parameters and Experiments

There are in total eight parameters whose setting up influences not only the learning process but also the entire quality of the resulting space description. In following we will summarize them and introduce some remarks concluded from experiments. Namely there are these parameters for GNG networks:

- number of adaptation steps given by the time t_{max},
- maximum number of neurons (points) of the output layer N,
- interval of inserting new points τ,
- maximum age of a connection T,
- learning parameters γ_{S1} and γ_{Si},
- parameters for reduction of the quantization error α and β.

By [7] the value for γ_{S1} is chosen considerably smaller than *0,3* – in our case $\gamma_{S1} = 0{,}05$ and for γ_{Si} the value will be approximately one tenth, i.e. $\gamma_{Si} = 0{,}006$. Further, for our experiments values of remaining parameters were: $T = 100$, $\alpha = 0{,}5$, $\beta = 0{,}0005$.

The experiments were mainly focused on observing parameters t_{max}, N and τ because just these parameters influence most the quality of the created network and interact mutually. Combining their various values and comparing obtained results following outcomes can be confirmed:

1. The optimum number of output neurons is approximately one hundredth of the number of training points.
2. The number of adaptation steps should be from two up to three times bigger as the number of training points.

The interval of inserting new points should enable inserting all points during the first 2/3 of the adaptation (having enough time for deploying new points to a correct position), i.e.:

$$\tau \approx \frac{2t_{\max}}{3N}. \tag{8}$$

During the experiments a roughly uniform deployment of training points was supposed. The bigger number of adaptation steps the more accurate the network but also the bigger computational efforts. Therefore, the proposed values for parameters express a compromise between descriptive precision and computational speed. Although the greatest influence on the position of a given neuron is caused by the initial step of inserting (6) but it will be influenced also by continuous adaptation of its position even if relating to the magnitudes of γ_{S1} and γ_{Si} in a considerably smaller measure. Therefore, it is necessary to enable a multiple adaptation of each neuron and from this reason t_{max} needs to have high values as well as an inserted neuron needs to have possibility of moderate position correction at least during the last third of the entire adaptation time t_{max}.

For needs of creating a GNG network describing the communication network of the city Košice a training set with *242 470* points was used and other parameters owned these values: $t_{max} = 900\,000$, $N = 2\,000$, $\tau = 300$, $T = 100$, $\alpha = 0{,}5$, $\beta = 0{,}0005$, $\gamma_{S1} = 0{,}05$, $\gamma_{Si} = 0{,}006$. The experiments showed the network was able to learn with the same quality on various types of road networks regardless the form and density as seen in the Fig. 6.

5. Conclusions

The proposed combination of NG networks and exact graph algorithms offers very advantageous properties for navigation either as an auxiliary for drivers [17] or as a direct means for navigation of mobile robots with the ability incrementally to modify space description. The absence of any definite topology of connections in the output layer (in comparison to Kohonen networks) enables NG networks to model whatever area of arbitrary complexity without any limitations regarding various restrains, e.g. forms of roads and traffic rules in the case of communications. This approach enables including further mechanisms like automatic removing of connections in the case of one-way roads or merging several independently created networks describing neighbourhood areas. It is possible to create an overview network with a less detailed description (like maps with different scales) for purposes of approximate navigation [13], too. Since the concept of NG networks is general for spaces with arbitrary dimensions it is also possible to incorporate for instance height data.

The advantage of this approach is based mainly on its two-stage processing. In the first stage a descriptive network is created although computationally demanding but necessary only ones, which will be later modified only occasionally using mentioned mechanisms. Anyway, also in this stage most of activities are automated (opposite to conventional approaches). In the second stage, which will be used many times, highly efficient algorithms are used for finding optimum paths whose time efforts do not exceed 3 seconds in the case of Košice.

The descriptive form of NG networks for contour manifold and heterogeneous spaces offers further utilization possibilities. It would be probably very perspective to use such a numerical knowledge representation form for knowledge extraction into rules using fuzzy logic (because of its nonlinear matter [14]) and its learning (adaptation) approaches [5] to obtain symbolic form of knowledge, which is necessary for more complex control and decision tasks.

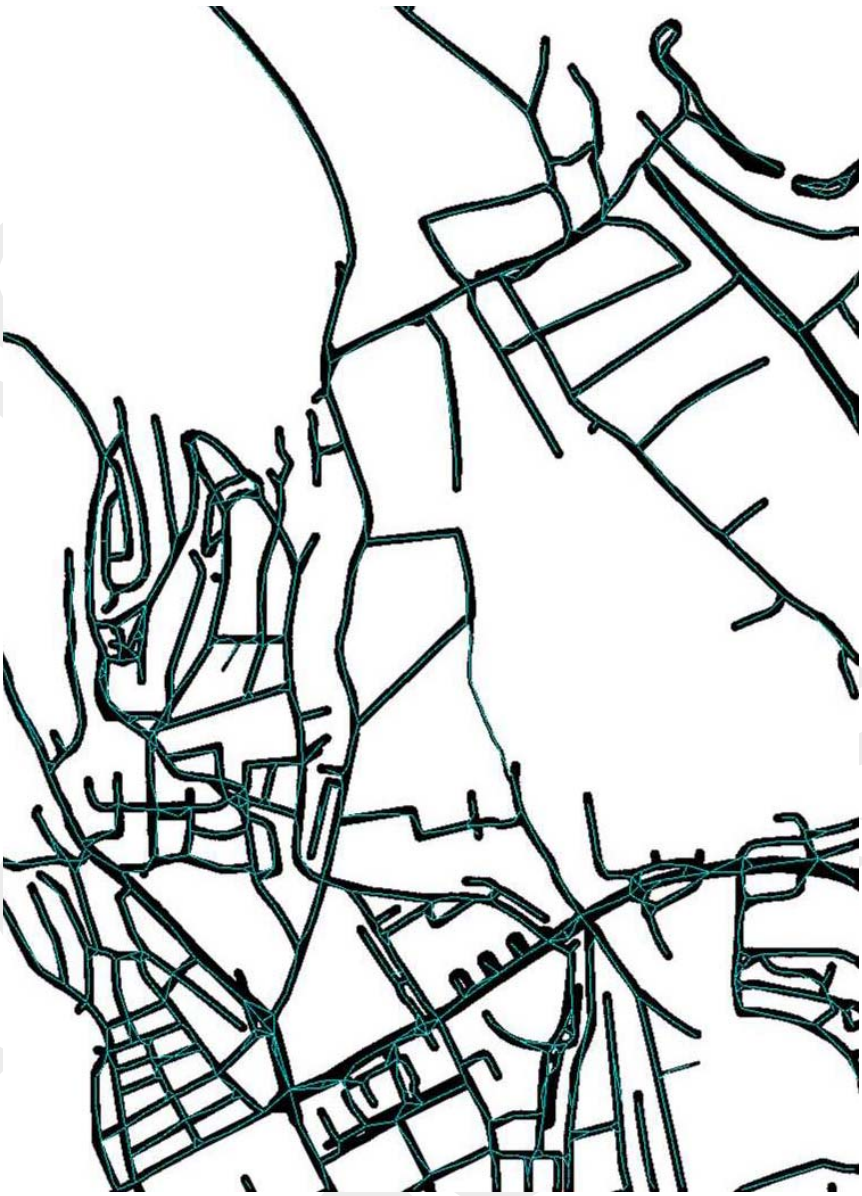


Figure 6. Description of the road network for urban part Košice – North

Acknowledgement

This publication is the result of the project implementation Centre of Information and Communication Technologies for Knowledge Systems (project number: 26220120020) supported by the Research & Development Operational Programme funded by the ERDF.

References

[1] Dudek, G., Jenkin M.: *Computational Principles of Mobile Robotics*, Cambridge University Press, Cambridge, (2000).

[2] Fritzke, B.: *A growing neural gas network learns topologies*, In: Advances in Neural Information Processing Systems 7; MIT Press Cambridge, USA, (1995), pp. 625-632.

[3] Fritzke, B.: *Some competitive learning methods*, (1997), pp. 45, <citeseer.ist.psu.edu/fritzke97some.html> [cit. 22.7. 2008].

[4] Fritzke, B.: *Vektorbasierte Neuronale Netze* (Prof. thesis in German), Shaker Verlag, (1998), pp. 157, <http://www.ki.inf.tu-dresden.de/~fritzke/> [cit. 22.7. 2008].

[5] Heinke, D., Hamker F.H.: *Comparing Neural Networks: A Benchmark on Growing Neural Gas*, *Growing Cell Structures, and Fuzzy ARTMAP*, IEEE Transactions on Neural Networks, Vol. 9, No. 6, (1998), pp. 1279-1291. [6]

[6] Holmström, J.: *Growing Neural Gas* (PhD. thesis), Uppsala University - Department of Information Technology, (2002), pp. 42. [7]

[7] Johanyák, Z. C., Kovács, Sz.: *A brief survey and comparison on various interpolation based fuzzy reasoning methods,* Acta Polytechnica Hungarica, Vol. 3, No. 1, (2006), pp. 91-105. [5]

[8] Lavalle, S. M.: *Planning algorithms*, Cambridge University, (2006), pp. 842, http://planning.cs.uiuc.edu/ [cit. 22.7. 2008].

[9] Martinetz, T. M., Schulte, K. J.: *A neural gas network learns topologies,* In: Artificial Neural Networks; North Holland Amsterdam, (1991), pp. 397-402.

[10] Martinetz, T. M.: *Competitive Hebbian learning rule forms perfectly topology preserving maps*, In: ICANN - International Conference on Artificial Neural Networks, Amsterdam, Springer, (1993), pp. 427-434. [11]

[11] Martinetz, T. M.: Selbstorganisierende neuronale Netzwerkmodelle zur Bewegungssteuerung (in German), Infix Verlag, (1992). [10]

[12] Milano, M., Koumoutsakos, P., Schmidhuber, J.: *Self-Organizing Nets for Optimization*, IEEE Transactions on Neural Networks, Vol. 15, No. 3, (2004), pp. 758-765.

[13] Mls. K.: *Implicit knowledge in Concept Maps and their Revealing by Spatial Analysis of Hand-Drawn Maps*, Proc. of the Second International Conference on Concept Mapping – Concept Maps: Theory, Methodology, Technology, Vol. 2, San José, Costa Rica, (2006).

[14] Oblak, S., Škrjanc, I., Blažič, S.: *If approximating nonlinear areas, then consider fuzzy systems,* IEEE Potentials, Vol. 25, No. 6, (2006), pp. 18-23.

[15] Pozna, C., Troester, F., Precup, R.-E., Tar, J. K., Preitl, St.: *On the Design of an Obstacle Avoiding Trajectory: Method and Simulation*, Mathematics and Computers in Simulation, Elsevier Science, Vol. 79, No. 7, (2009), pp. 2211-2226.

[16] Rutrich, M.: *Využitie sietí typu Neural Gas v navigácii* (MSc. thesis in Slovak), TU v Košiciach, (2007), pp. 67.

[17] Spalek, J., Dobrucký, B., Lusková, M., Pirník, R.: *Modeling of Traffic Flows of Suburban Agglomerations: Technical and Social Aspects, Applications*, The 2nd International Conference on Knowledge Generation, Communication and Management: KGCM 2008 ORLANDO, (2008).

[18] Vaščák, J., Rutrich, M.: *Path Planning in Dynamic Environment Using Fuzzy Cognitive Maps*, In: SAMI – 6th International Symposium on Applied Machine Intelligence and Informatics, Herľany, Slovakia, January 21-22 (2008), pp. 5-9. [19]
[19] Vaščák, J., Szászi, T.: *Navigácia mobilných robotov pomocou harmonických potenciálových polí (1) and (2)* (in Slovak), In: AT&P Journal, No. 2 and 3, (2007) pp. 58-60 and 74-75. [18]
[20] Vaščák, J.: *Fuzzy cognitive maps in path planning*, In: Acta Technica Jaurinensis: Series intelligentia computatorica, Vol. 1, No. 3, (2008), pp. 467-479.
[21] Yeh, M. F., Chang, K. Ch.: *A Self-Organizing CMAC Network With Gray Credit Assignment,* IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics, Vol. 36, No. 3, (2006), pp. 623-635.

Avoiding of Saturation and Resonance Effects via Simple Adaptive Control of an Electrically Driven Vehicle Using Omnidirectional Wheels

József K. Tar*, János F. Bitó†, Csaba Ráti‡ (student)

Budapest Tech Polytechnical Institution
H-1034 Budapest,Bécsi út 96/B., Hungary
***Transportation Informatics and Telematics Knowledge Centre**
†‡John von Neumann Faculty of Informatics,
Institute of Intelligent Engineering Systems
E-mail: *{tar.jozsef@nik.bmf.hu}, †{bito@bmf.hu}

Abstract: The task of solving the adaptive control of a partially and imprecisely modeled electrical vehicle driven by three omnidirectional wheels together with the torque and/or power limits of the drives is considered. Instead of the application of precise analytical models and sophisticated parameter estimation techniques the vehicle is modeled as a rigid body while the considerable part of the burden carried by it through elastic connection is completely neglected in the controller's model. Instead of that, a simple kinematically designed, PID–type trajectory tracking is formulated that is approximated by robust fixed point transformations. It is shown that if the nominal trajectory to be traced does not significantly excite the unknown degree of freedom of the vehicle–burden connection precise and stable control can be achieved by the adaptivity, while the pure PID–type control may excite this degree of freedom and can be corrupted by achieving either the torque or the power limits of the actual drives applied. This statement is substantiated by numerical simulations. The main advantage of the proposed control method is that it operates with local basin of attraction developed for convergent iterative Cauchy sequences that is easy to design by setting only a few parameters. Its disadvantage is that it cannot guarantee global stability therefore its application must be preceded by numerical tests. Its use may be especially useful in applications in industrial workshops when modeling the dynamics of the coupled burden technically is very difficult, e.g. when it is a tank partially containing liquid.

Keywords: Adaptive Control, Robust Fixed Point Transformations, Iterative Control, Omnidirectional Wheels

1. Introduction

Due to practical motivations various efforts has been exerted to plan and precisely trace trajectories for mobil robots navigating in a environment often crowded by various obstacles. The traditional algorithms [e.g. the BUG –a wall tracing algorithm used by insects– and its variant TBA –Tangent Bug Algorithm– (1) or the SFP –Shortest Feasible Path– Algorithm (2)] and the Visibility Graph Algorithm e.g. (3) plan the trajectory in the close vicinity of the obstacles that are augmented by some roughly estimated "*safety zones*" to evade accidental collision with them. This rough estimation may exclude useful but very narrow passages that otherwise would be traceable by the robot. For this purpose precise robot navigation techniques are available that –by the use of various markers– more precisely can determine the proper extent of augmentation of the obstacles [e.g. (4) and (5)]. For selecting the "best" route various weighting techniques were proposed as e.g. the original version (6) and its refinement (7) in which besides the edges of the graph the vertices are also weighted according to the appropriate turns the curvature of which can limit the maximum allowed velocity of the robot motion on the basis of dynamical considerations.

The subject area of the present paper is the precise realization of the routes proposed by various route planning techniques when the dynamical model of the cart is imprecise and incomplete, i.e. it is approximated as a rigid body while the burden it carries partly is connected to it by an elastic spring. The carried burden is supposed to have viscous friction in its contact with the floor that is supposed not to be exactly horizontal in the reality though the controller's rough model calculates with strictly "vertical" gravitational acceleration. The dynamical properties of this burden as well as of the connecting spring are completely abandoned in the controller's rough model. A three wheeled robot cart motored by omnidirectional wheels of electrical drives are considered in details. A detailed model and a purely kinematically formulated, desired PID-type control is elaborated in Section 2. In Section 3 the principles of a simple adaptive control using local basins of attraction instead of some more complicated Lyapunov function technique is briefed. Following that, in Section 4 comparative results are given for adaptive and non–adaptive path tracking for various trajectories containing sharper turns, too. Section 5 contains the conclusions drawn as the result of the investigations. It is shown that by the use of the simple adaptive technique proposed the excitation of the coupled degree of freedom not present in the controller's model can be evaded if the trajectory to be tracked does not excite it directly. It is shown, too, that the simple PID–type tracking policy without adaptation can generate considerable excitation that may lead to instability partly caused by reaching the torque and/or the power limits of the drives.

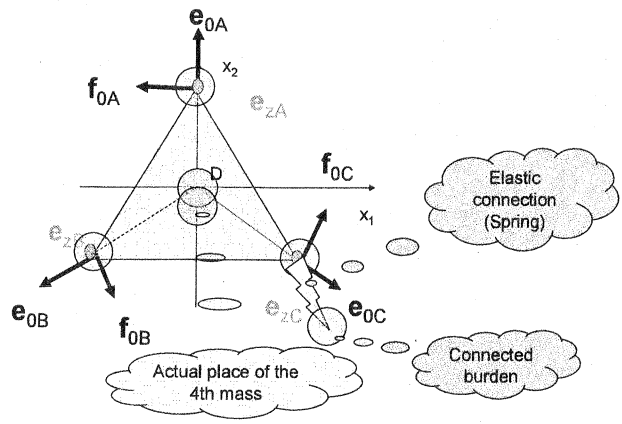


Figure 1: The rough description of the vehicle model considered

2. The Dynamic Model of the Vehicle and the PID-type Control

As is well known the conventional vehicles with Ackerman steering system suffer from significant limitation: due to geometric reasons it is impossible to simultaneously prescribe their precise location and orientation while tracking a given trajectory. For planning trajectories for such devices various practical techniques are available [e.g. (8), (9)]. In contrast to the Ackerman devices AGVs using omnidirectional wheels [e.g. (10)] can precisely track arbitrary trajectories at least from kinematic point of view. On this reason for our purposes a triangular structure similar to that in (10) was chosen as a paradigm (Fig. 1). The cart was supposed to have canonical triangular shape of side length $L = 2\ (m)$. The orientation of the active forces were supposed to be described by the orthogonal unit vectors ($\mathbf{e}_A$, $\mathbf{f}_A$, $\mathbf{e}_B$, $\mathbf{f}_B$ and $\mathbf{e}_C$, $\mathbf{f}_C$) at wheels A,B, and C in the (x,y) plane in which the direction of the appropriate $\mathbf{e}$ vectors was identical to that of the straight line connecting the geometric center of the triangle to the appropriate vertices. These vectors were assumed to rigidly rotate around the axis z with angle q_3. Each wheel had the common constant vector component in the z direction $\mathbf{e}_z$ along which the contact constraint forces originating from the ground acted. It was assumed that the plane of motion in the reality was not exactly horizontal, so the vector of the gravitational acceleration in the reality had components in the x, y, and z directions, too. However, the controller's rough model assumed the full vector in the $-z$ direction. At the vertices of the triangle three heavy wheels and drive systems were located, each of them had the mass $M = 30\ kg$. It was assumed that further $2 \cdot M\ kg$ mass was located over the geometric center of the triangle at the height of $h_D = 0.5\ m$. The vehicle was assumed to move on the (x,y) plane with

prescribed nominal location of the projection of its hypothetical mass center point $\mathbf{S}^{(m)^N}$ m and nominal rotational pose q_3^N rad around the axis z. According to Fig. 1 the *"not modeled degree of freedom"* was a mass-point connected to wheel C by an elastic connection, a spring. In this case it had the mass of $0.45 \cdot M$ attached to a spring of stiffness $k = 1000\ N/m$ and zero force length $L_0 = 1\ m$. It was assumed to move along the (x,y) plane with a viscous friction coefficient $\mu = 5\ Ns/m$.

Utilizing the well known fact that the acceleration of the mass center point of a rigid body multiplied by its full mass is equal to the sum of the external forces acting on that system, and that the time-derivative of momentum of the system computed with respect to the actual mass center point is equal to the momentum of the external forces (torque) with respect to this point the required active driving force components F_{Ae_A}, F_{Af_A}, F_{Be_B}, F_{Bf_B}, and F_{Ce_C}, F_{Cf_C}, as well as the hypothetical vertical constraint force components F_{A_z}, F_{B_z}, and F_{C_z} can be calculated. According to Fig. 1, since the small wheels do not have drives in the horizontal **e** directions no forces can be exerted. The rough dynamic model available for the controller is given in Eq.(1):

$$\begin{bmatrix} 5M^{(m)}(\ddot{\mathbf{S}}^{(m)} + \mathbf{g}^{(m)}) \\ \dot{\mathbf{P}}^{(m)} \end{bmatrix} = \begin{bmatrix} \mathbf{A}^{(m)} & \mathbf{B}^{(m)} & \mathbf{C}^{(m)} \\ \mathbf{D}^{(m)} & \mathbf{E}^{(m)} & \mathbf{F}^{(m)} \end{bmatrix} \begin{bmatrix} \mathbf{F}_e \\ \mathbf{F}_f \\ \mathbf{F}_z \end{bmatrix}, \qquad (1)$$

in which $\mathbf{P}^{(m)}$ denotes the *model momentum vector*, $\mathbf{g}^{(m)}$ is the vector of the gravitational acceleration having $\theta = 0.4\ rad$ tilting angle and $\phi = 0.5\ rad$ rotation with respect to the axis z (in the model it was assumed to have only a z component), and the 3×3 sized blocks of the big matrix are defined as follows: $\mathbf{A}^{(m)} = [\mathbf{e}_A,\mathbf{e}_B,\mathbf{e}_C]$, $\mathbf{B}^{(m)} = [\mathbf{f}_A,\mathbf{f}_B,\mathbf{f}_C]$, $\mathbf{C}^{(m)} = [0,0,0;0,0,0;1,1,1]$, $\mathbf{D}^{(m)} = [\mathbf{e}_A \times \mathbf{x}_A^{(m)},\mathbf{e}_B \times \mathbf{x}_B^{(m)},\mathbf{e}_C \times \mathbf{x}_C^{(m)}]$, $\mathbf{E}^{(m)} = [\mathbf{f}_A \times \mathbf{x}_A^{(m)},\mathbf{f}_B \times \mathbf{x}_B^{(m)},\mathbf{f}_C \times \mathbf{x}_C^{(m)}]$, and $\mathbf{F}^{(m)} = [\mathbf{e}_z \times \mathbf{x}_A^{(m)},\mathbf{e}_z \times \mathbf{x}_B^{(m)},\mathbf{e}_z \times \mathbf{x}_C^{(m)}]$, $\mathbf{F}_e = [F_{Ae_A},F_{Be_B},F_{Cf_C}]^T$, $\mathbf{F}_f = [F_{Af_A},F_{Bf_B},F_{Cf_C}]^T$, and $\mathbf{F}_z = [F_{A_z},F_{B_z},F_{C_z}]^T$. The $\mathbf{x}_A^{(m)}$, $\mathbf{x}_B^{(m)}$, and $\mathbf{x}_C^{(m)}$ vectors connect the assumed mass center point with the appropriate vertices at the wheels A,B, and C.

The "*actual system's*" equation of motion that can be used for calculating the "*realized accelerations*" and "*realized contact forces in the z direction*" is similar to Eq.(1), but it contains the acceleration of the actual mass center point **S** and the actual momentum calculated with respect to that (**P**). (Fortunately $\mathbf{S}^{(m)}$ and **S** have simple geometric connection.) Beside that it contains the The $\mathbf{x}_A$, $\mathbf{x}_B$, and $\mathbf{x}_C$ vectors that connect the *actual mass center point* with the appropriate vertices at the wheels A,B, and C. Furthermore, the equation has to be rearranged since in it in the "input side" we have $\mathbf{F}_e$ and $\mathbf{F}_f$, and the unknown quantities are $\mathbf{F}_z$, $\ddot{\mathbf{S}}$ and $\ddot{q}_3$. By expressing $\dot{\mathbf{P}}$ with q_3, $\dot{q}_3$ and $\ddot{q}_3$ it is obtained that

$$\begin{bmatrix} 5M\ddot{\mathbf{S}} \\ \dot{\mathbf{P}} \end{bmatrix} = -\begin{bmatrix} 5M\mathbf{g} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{D} & \mathbf{E} \end{bmatrix} \begin{bmatrix} \mathbf{F}_e \\ \mathbf{F}_f \end{bmatrix} + \begin{bmatrix} \mathbf{C} \\ \mathbf{F} \end{bmatrix} [\ \mathbf{F}_z\ ], \qquad (2)$$

in which $\dot{\mathbf{P}}$ contains certain elements of the actual inertia matrix Θ, and the arrays $\mathbf{H} = [\mathbf{e}_z \times \mathbf{x}_A, \mathbf{e}_z \times \mathbf{x}_B, \mathbf{e}_z \times \mathbf{x}_C, -\Theta^{(3)}; 1,1,1,0]$. From the fact that certain kinematic data can be exactly known it concludes that $\mathbf{A} = \mathbf{A}^{(m)}$, $\mathbf{B} = \mathbf{B}^{(m)}$, and $\mathbf{C} = \mathbf{C}^{(m)}$. In the calculations it was taken into account that the full momentum of the gravitational forces with respect to the mass center point is zero, and that no acceleration component may exist in the z direction (supposing that the vehicle does not turn up). So the appropriate component of the gravitational forces must be compensated by the contact forces in the direction z. In the solution of the "*actual system's equations*" it can be utilized that they are decoupled to some extent: $\ddot{\mathbf{S}}$ has only x and y components, as well as the arrays $\mathbf{A}$ and $\mathbf{B}$, while the array $\mathbf{C}$ does not have 1st and 2nd components. On this reason the two nontrivial nonzero components of $\ddot{\mathbf{S}}$ can be determined independently of the $\mathbf{F}_z$ values, while its zero 3rd component yields some restriction for the sum of the components of $\mathbf{F}_z$. This can be associated with the three equations pertaining to $\dot{\mathbf{P}}$, therefore we obtain 4 equations for 4 unknown quantities in Eq.(3):

$$\begin{bmatrix} -\dot{q}_3^2\Theta_{23} \\ \dot{q}_3^2\Theta_{13} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -\mathbf{F}_{ABCef} \\ 5Mg_3 \end{bmatrix} = \mathbf{H} \begin{bmatrix} F_{A_z}^{Real} \\ F_{B_z}^{Real} \\ F_{C_z}^{Real} \\ \ddot{q}_3^{Real} \end{bmatrix}. \qquad (3)$$

Equation (3) does not contain the model of the connected subsystem the existence of which can be taken into account by calculating the contact forces (and their momentum) that appear due to the dynamic coupling of these subsystems. (The equations of motion of the mass point were solved separately.)

In the simulations Eqs.(1) and (3) were used as follows. On purely kinematical basis a PID-type control was prescribed to obtain the "*desired accelerations*" as $\ddot{\xi}^D = \ddot{\xi}^N + P_\alpha(\xi^N - \xi) + D_\alpha(\dot{\xi}^N - \dot{\xi}) + I_\alpha \int_{t_0}^{t} [\xi(\tau)^N - \xi(\tau)]d\tau$ in which $\alpha = \{q_3, xy\}$, $P_{q_3} = P$, $D_{q_3} = D$, $I_{q_3} = I$, $P_{xy} = 4P$, $D_{xy} = 4D$, $I_{xy} = 4I$ with $P = 50\ s^{-2}$, $D = 10\ s^{-1}$, and $I = 5\ s^{-3}$. The superscript N refers to the *nominal accelerations* determined by the trajectory to be traced. In the charts displaying the results instead of the *realized acceleration of the actual mass center point* that of the hypothetical one, $\mathbf{S}^{Real}$ are given. The inverse of $\mathbf{H}$ was calculated to obtain the realized angular acceleration $\ddot{q}_3^{Real}$ and the realized vertical constraint force components at the wheels as $F_{A_z}^{Real}$, $F_{B_z}^{Real}$, and $F_{C_z}^{Real}$. If one of these components becomes negative the vehicle becomes apt to turn up. Besides this, the active driving forces in the (x,y) plane cannot be exerted in the lack of appropriate pressing forces in the z direction at the appropriate wheels. For safety

reasons these components must be over a safety limit to evade turning up of the vehicle. On the basis of the *approximate model* used by the controller and the *exact one* applied in the simulation investigations can be carried out for the proposed PID control for various nominal trajectories to be tracked.

As it can generally be expected in the case of a rigid vehicle model, proper increase in the feedback gains can reduce the tracking error even in the case of a very rough available dynamic model. The same expectation is not substantiated when the controlled system contains unknown internal degrees of freedom not taken into account in the controller's model. Certain PID settings can excite these hidden connections that may lead to instabilities. In connection with such excitations another important factor is the limited torque and power of the actual drives applied. In her PhD Thesis Zsuzsa Preitl (11) investigated various electrical driving systems for the use in hybrid vehicles. The so called Brushless DC Machines [e.g. (12), (13)] are very good potential solutions for such purposes. According to (14) such systems can work in *forward and backward motoring mode* when the rotational velocity of their axis and the exerted torque have the same signum, otherwise they work in the *forward and reverse braking mode*. In the present investigations it was supposed that a controlled variant of such motors can exert arbitrary torque between zero and a maximum torque limit when the speed of rotation is small. However, over a velocity limit power limitation becomes effective that means that the actual torque multiplied by the actual velocity cannot exceed a preset power limit. The so obtainable torque vs. velocity curve was used as an envelope in the calculations. [In considering the motion along the (x,y) plane the appropriate limits were "translated" to force limit of $F_{f_{max}} = 7000\ N$, and velocity limit of $v_{max} = 8\ m/s$.] For the simplicity energy recuperation in the braking mode (for which modeling of the operation of the battery would have been necessary) was not considered in the present simulation. Instead of that simple dissipative braking based on mechanical friction with maximal braking force $F_{f_{max}}$ was assumed. The limitations of this saturated system appeared in the simulations when no adaptive controller was used.

An alternative, more intelligent approach of improving tracking precision instead of increasing the feedback parameters is the application of adaptive control at reduced PID feedback gains that may introduce into the system less drastic corrective contributions than the simple PID controller. In the next section this adaptive controller is outlined.

3. Adaptive Control Based on the Excitation–Response Scheme

3.1. The underlying idea

The basic idea of the control approach was published e.g. in (15). Certain control task can be formulated by using the concepts of the appropriate *"excitation"* Q of the controlled system to which it is expected to respond by some prescribed or *"desired response"* r^d. The appropriate excitation can be computed by the use of some *inverse dynamic model* $Q = \varphi(r^d)$. Since normally this inverse model is neither complete nor exact, the actual response determined by the system's dynamics, ψ, results in a *realized response* r^r that differs from the desired one: $r^r \equiv \psi(\varphi(r^d)) \equiv f(r^d) \neq r^d$. It is worth noting that the functions $\varphi()$ and $\psi()$ may contain various hidden parameters that partly correspond to the dynamic model of the system, and partly pertain to unknown external dynamic forces acting on it. The controller normally can manipulate or "deform" the input value from r^d so that $r^r \equiv \psi(r^d_*)$. Such a situation can be maintained by the use of some local deformation that can properly "drag" the system's state in time meandering along some trajectory.

3.2. Design issues

To realize the above outlined local deformation fixed point transformation was introduced in (16) that is rather "robust" as far as the dependence of the resulting function on the behavior of $f(\bullet)$ is concerned. This robustness can approximately be investigated by the use of an affine approximation of $f(x)$ in the vicinity of $x_\star$ and it is the consequence of the strong nonlinear saturation of the sigmoid function $\tanh(x)$:

$$\begin{gathered} G(x|x^d) := (x+K)\left[1 + B\tanh(A[f(x) - x^d])\right] - K \\ G(x_\star|x^d) = x_\star if f(x_\star) = x^d, G(-K|x^d) = -K, \\ G(x|x^d)' = \frac{(x+K)ABf'(x)}{\cosh(A[f(x)-x^d])^2} + 1 + B\tanh(A[f(x) - x^d]), \\ G(x_\star|x^d)' = (x_\star + K)ABf'(x_\star) + 1. \end{gathered} \tag{4}$$

It is evident that the transformation defined in (4) has a proper ($x_\star$) and a false ($-K$) fixed point, but by properly manipulating the control parameters A, B, and K the good fixed point can be located within its basin of attraction, and the requirement of $|G'(x_\star|x^d)| < 1$ can be guaranteed. This means that the iteration can have considerable speed of convergence even nearby $x_\star$, and the strongly saturated tanh function can make it more robust in its vicinity, that is the properties of $f(x)$ have less influence on the behavior of G. It is not difficult to show that in the case of *Single Input – Single Output (SISO)* systems the $G(x|x^d)$ (or in a simpler notation the $G(x)$) functions can realize contractive

mapping around $x_\star$, i.e. the conditions of $|G'| \leq H < 1$ $[0 \leq H < 1]$ can be maintained by properly choosing the parameters of this function and if f' is finite and its sign and absolute value can be estimated in the vicinity of $x_\star$. Then the sequence of points $\{x_0, x_1 = G(x_0), ..., x_{n+1} = G(x_n), ...\}$ obtained via iteration form a *Cauchy Sequence* that is convergent $(x_n \to x_\star)$ in the real numbers and converge to the solution of the *Fixed Point Problem* $x_\star = G(x_\star)$:

$$\begin{aligned} |G(x_\star) - x_\star| &\leq |G(x_\star) - x_n| + |x_n - x_\star| = \\ &= |G(x_\star) - G(x_{n-1})| + |x_n - x_\star| \leq \\ &\leq H|x_\star - x_{n-1}| + |x_n - x_\star| \to 0, x_n \to x_\star. \end{aligned} \tag{5}$$

A possible way of applying this simple idea elaborated for SISO to *Multiple Input – Multiple Output (MIMO)* systems is its separate application for each component of a vector valued r, r^d. To mathematically substantiate this statement consider the fact that any contractive map acting in linear, normed, complete metric spaces (i.e. in Banach spaces) generally yields Cauchy sequences that must be convergent due to the completeness of the space. Therefore, all the above considerations can be applied for MIMO systems if instead of the absolute values the following norm is used for $\vec{x} \in \Re^n$: $||\vec{x}|| := \sum_{i=1}^n |x_i|$. If a multiple dimensional sigmoid function $\vec{\sigma} : \Re^n \to \Re^n$ defined as $y_i = \sigma^{(i)}(x_i)$ $\{i = 1,2,...,n\}$ in its each component is contractive, i.e. for $\forall i \; \exists 0 \leq M_i < 1$ so that $|\sigma^{(i)}(a) - \sigma^{(i)}(b)| \leq M_i|a - b|$ then it can be stated that $||\vec{\sigma}(\vec{a}) - \vec{\sigma}(\vec{b})|| := \sum_{i=1}^n |\sigma^{(i)}(a_i) - \sigma^{(i)}(b_i)| \leq [\max_{i=1}^n \{M_i\}] \sum_{j=1}^n |a_j - b_j| \equiv M||\vec{a} - \vec{b}||$, $0 \leq M < 1$, i.e. it is contractive in the normed $\Re^n$ space, too.

Regarding the way of designing the control parameters it can be noted that the PID-type feedback parameters can be prescribed independently of the adaptive parameters on the basis of purely kinematic consideration, e.g. by prescribing some exponentially decaying error $\vec{e}$ as e.g. $(d/dt + \Lambda)^m \vec{e} = \vec{0}$ where m depends on the order of the system. The A, B, and K adaptive parameters can be designed by using simulation according to their particular role in the control. Parameter A determines the *"range of response error"* monitored by the saturated sigmoid function: small A monitors a wide range, big A monitors a narrow interval. Parameter B normally can be taken to be 1. Parameter K can be determined by studying the "response error" on the non-adaptive controller. A few times the maximum of the so occurred absolute value normally works well. Normally it is expedient to start the simulations with very small (e.g. $A \approx 10^{-6}$) and increase it step by step. Following a few steps of computations stable regimes can be easily found.

In the adaptive control of the vehicle driven by three omnidirectional wheels this simple control idea was applied. This approach is far simpler than the conventional, Lyapunov function based techniques e.g. the *Slotine & Li adaptive controller* (17) [a detailed comparison was given e.g. in (18)]. Its other, very important advantage is that while Slotine

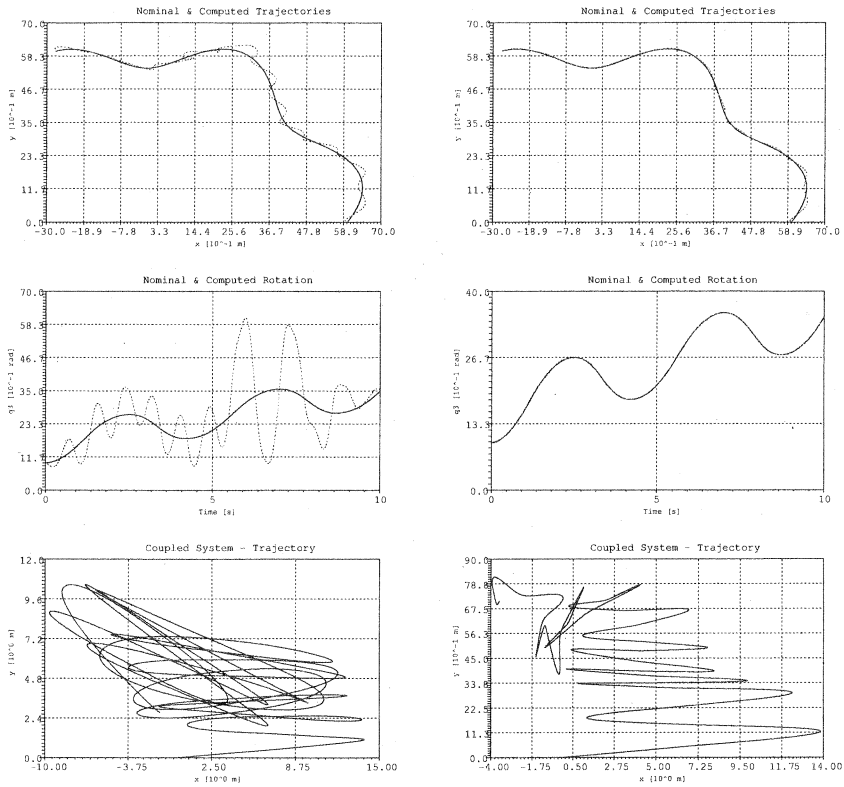


Figure 2: Trajectory and orientation tracking, and the trajectory of the connected subsystem [LHS: non–adaptive, RHS: adaptive control]

& Li's controller is unable to properly compensate the effects of unknown, additional external disturbances, the here proposed one is able for that. It is a significant advantage since the dynamically coupled hidden subsystem's effects appear just in this form in the equations of motion of the modeled subsystem. In the simulations instead of the tanh function another sigmoid, $\sigma(x) := x/(1 + |x|)$ was applied that has similar qualitative properties.

4. Computational Results

The adaptive loop worked with $\Delta t = 1\ (ms)$ time-resolution and $K_{Ctrl_{q3}} = -4000$, $B_{Ctrl_{q3}} = 1$, $A_{Ctrl_{q3}} = 10^{-4}$ settings for the rotational axis, and $K_{Ctrl_{xy}} = -1000$, $B_{Ctrl_{xy}} = 1$, and $A_{Ctrl_{xy}} = 5 \times 10^{-4}$ values for the translation in the (x,y) plane. According to Fig. 2 the task was tracking a complicated trajectory having small turns. It is evident from the charts that the adaptive control considerably improved the precision of the trajectory and orientation tracking and resulted in reduced swinging/excitation of the joined burden carried by the cart. For a better comparison the tracking errors are displayed in Fig. 3. It is evident that the adaptive control guaranteed relatively slow motion of the end of the spring attached to the cart at wheel C, therefore the initial excitation of the coupled subsystem relaxed in the adaptive case. In contrast to that the non–adaptive control resulted in permanent excitation of this degree of freedom. Fig. 4 reveals that the adaptive nature of the control manifested itself in better approximation of the *desired accelerations* prescribed by the simple, kinematically formulated PID–type control.

To study the saturation effects the active force components and the power consumption of the drives of the appropriate wheels are described in Fig. 5. In our case negative power consumption corresponds to braking, positive one to motoring. It is evident that in the non–adaptive case the driving force limit, the braking force limit, and the power limits of the drives were sometimes achieved. No such problems were observed in the adaptive case. Persistent saturation in general may result in divergence. Since the limitations of the drive system cannot be exceeded by the adaptive law, within the frames of the present approach the only possibility for evading saturation is the limitation of the PID feedback parameters if the problems appear even in the adaptive case, too.

To study the stability of the motion the velocity profiles and the repulsive contact force components acting from the tilted plane of the motion are described in Fig. 6. It is evident that the simple adaptive law considerably improved the stability of the controlled motion.

The better describe the effect of the adaptivity on the efficiency of the motion the sections belonging to the motoring and the braking phases are described in Fig. 7. The non–adaptive case evidently works very inefficiently by using alternating motoring/braking phases, while application of adaptivity reduces the frequency of this alternation and makes it better fitting to the required values originating from the properties of the nominal trajectory to be traced.

5. Conclusions

In this paper the results of a dynamic numerical simulation were presented for the control of a triangle-shaped vehicle driven by three, electrically motored omnidirectional wheels.

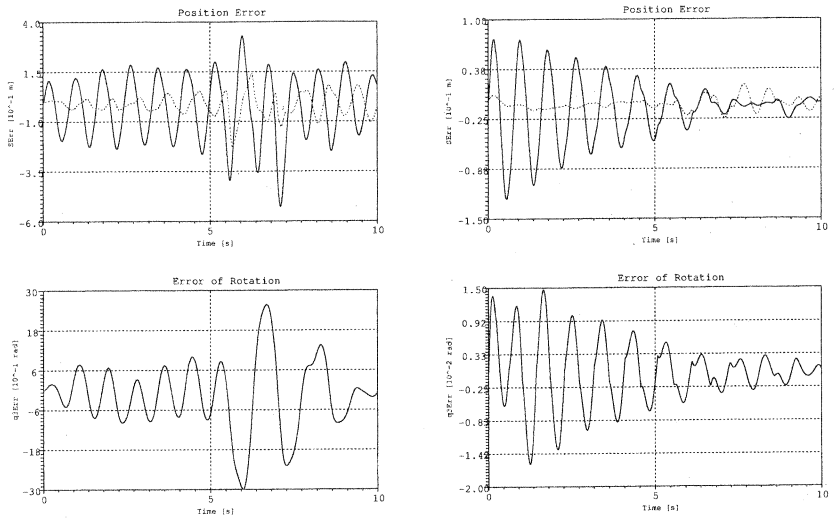


Figure 3: Trajectory and orientation tracking errors vs time [LHS: non–adaptive, RHS: adaptive control]

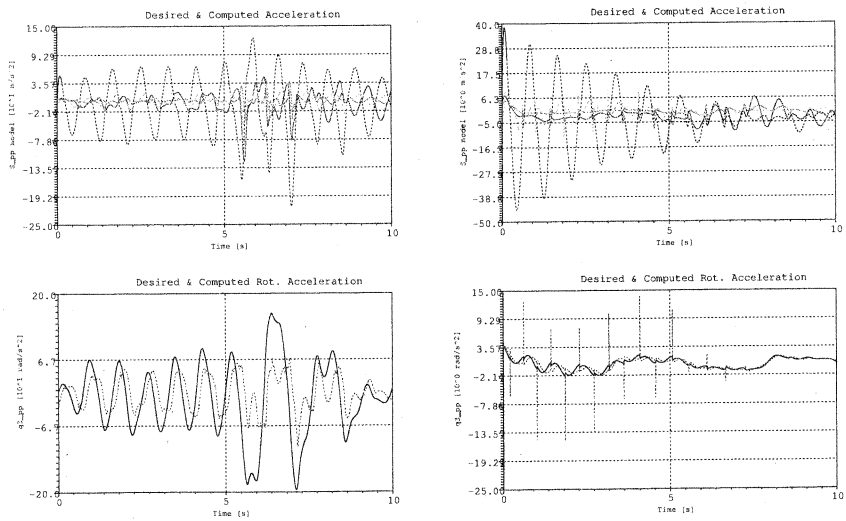


Figure 4: The *desired* and *simulated* accelerations vs time [LHS: non–adaptive, RHS: adaptive control]

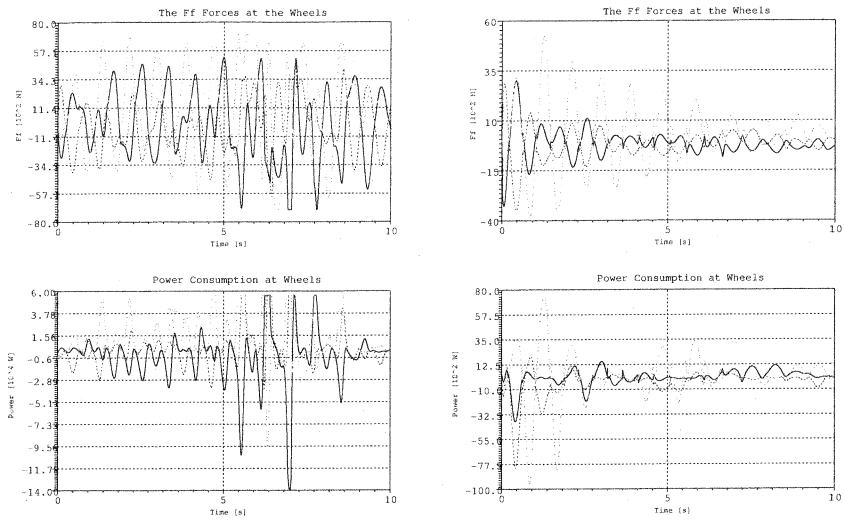


Figure 5: The active driving force components and the power consumption of the wheels vs time [LHS: non–adaptive, RHS: adaptive control]

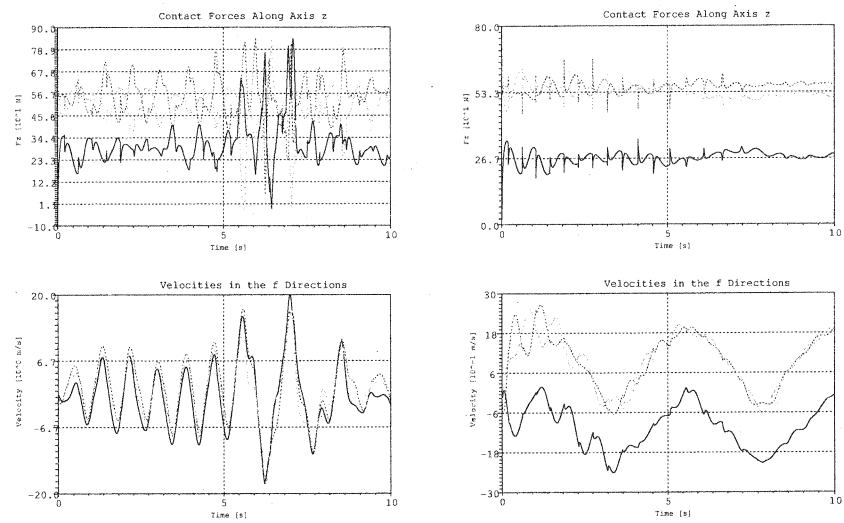


Figure 6: The contact repulsive forces in the z direction and the velocities of the wheels at the appropriate **f** directions vs time [LHS: non–adaptive, RHS: adaptive control]

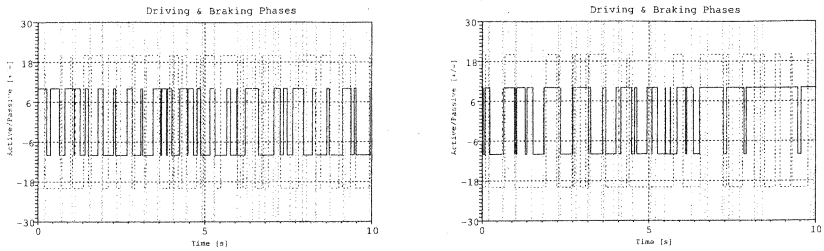


Figure 7: The variation of the motoring/braking phases of the motion vs time: motoring phase (coded by the values 1,2, and 3 for wheels A,B, and C, respectively), and the braking phases (denoted by the values $-1,-2$, and -3 for wheels A,B, and C); [LHS: non–adaptive, RHS: adaptive control]; (the sign of the appropriate coding value equals to that of the power consumption at the wheel under consideration; it is not easy to identify the appropriate signs in Fig. 5)

It was shown that a special, simple, fixed point transformations based adaptive control can cause quite precise trajectory and orientation tracking in spite of the modeling errors pertaining the modeled subsystem and the existence of the abandoned subsystem of the controlled physical system. The effect of the adaptive approach on the stability of the achieved motion and the efficiency of the use of the torque and power resources of the electrical driving system was studied, too. Saturation effects in the controlled drives were also taken unto account. It was definitely shown that the introduction of the adaptive approach definitely improved the quality of the controlled system.

The proposed approach has limited number of control parameters apart from the three coefficients of the PID controller. It far more easily can be constructed than e.g. a traditional Lyapunov function based solution. (Such systems in principle can be controlled by the use of some GPS installed in their environment.)

The present paper a simple Euler integration was applied. In the future we wish replace the presently applied simple Euler integration with a more sophisticated integrator offered by SCILAB's SCICOS.

Acknowledgment

The authors gratefully acknowledge the support by the *National Office for Research and Technology (NKTH)* using the resources of the *Research and Technology Innovation Fund* within the projects *OTKA: No. CNK-78168* and *"Pázmány Péter Project": No. RET-*

10/2006.

References

[1] T. Lozano-Perez. Spatial planning: a configuration space approach. *IEEE Trans. on Computers* Vol. 32, 1983.

[2] P. Moutarlier, B. Mirtich, and J. Canny. Shortest paths for a car-like robot to manifolds in configuration space. *The International Journal of Robotics Research*, Vol. 15, No. 1, pp. 36–60, 1996.

[3] K.T. Simsarian, T.J. Olson, N. Nandhakumar. View invariant regions and mobile robot self-localisation. *IEEE Trans. Rob. Autom.* Vol. 12 No. 5, 1996.

[4] I. Nagy, A.L. Bencsik. Model-based path planning algorithm in respect of the APF and AEF of the environment. *Proc. of the 7th International Conference on Intelligent Engineering Systems 2003 (INES'03), March 4-6, Assiut-Luxor, Egypt*, vol. 2., pp. 551–556, 2003.

[5] I. Nagy. Path planning algorithm based on user defined maximal localization error. *Periodica Polytechnica*, BUTE, Hungary, pp. 43–57, 2005.

[6] G. Tan, D. Mamady. Real-time global optimal path planning of mobile robots based on modified ant system algorithm. *ICNC 2006, Part II, LNCS 4222*, Springer-Verlag Berlin Heidelberg, pp. 204–214, 2006.

[7] I. Nagy, L. Vajta. Local trajectory optimization based on dynamical properties of mobile platform. *Proc. of the 2001 IEEE International Conference on Intelligent Engineering Systems (INES 2001), Helsinki, Finland, September 16-18, 2001*, pp. 285–290, 2001.

[8] R. M. Murray, S. Sastry. Nonholonomic motion planning. Steering using sinusoids. *IEEE Transactions on Automatic Control*, Vol. 38 No. 5, pp. 700–716, 1993.

[9] J.K. Tar, I.J. Rudas, J.F. Bitó. Constraints' resolution by optimal trajectory planning for anholonom devices. *Proc. of the 34th Annual Conference of the IEEE Industrial Electronics Society (IECON 2008), 10-13 November 2008, Orlando, FL, U.S.A.*, pp. 1597–1601, 2008.

[10] J.M. Holland. Basic Robotics Concepts. Howard W. Sams, Macmillan, Inc., Indianapolis, IN., 1983.

[11] Zs. Preitl. Control design methods for optimal energy consumption systems. *PhD Thesis*, Budapest University of Technology and Economics (BUTE), 2009.

[12] C. Mi, M. Filippa, J. Shen, N. Natarajan. Modeling and control of a variable–speed constant frequency synchronous generator with brushless exciter. *IEEE Transactions on Indsutry Applications*, Vol. 40, No. 2, March/April 2004.

[13] O.F. Bay, G. Bal, S. Demirbas. Fuzzy logic based control of a brushless DC servo motor drive. *Proc. of the 7th International Power Electronics & Motion Control Conference and Exhibition*, Budapest, Hungary, Vol. 3, pp. 448–452, 1996.

[14] T-C. Tsai, M-C. Tsai. Powe control of a brushless permanent magnet electric machine for exercise bikes. *Proc. of the IFAC 15th Triennial World Congress*, Barcelona, Spain (in electronic format), 2002.

[15] J.K. Tar, I.J. Rudas and K.R. Kozłowski. Fixed point transformations-based approach in adaptive control of smooth systems. *Lecture Notes in Control and Information Sciences 360 (Eds. M. Thoma and M. Morari), Robot Motion and Control 2007 (Ed. Krzysztof R. Kozłowski)*, Springer Verlag London Ltd., pp. 157–166, 2007.

[16] J.K. Tar, J.F. Bitó, I.J. Rudas, K.R. Kozłowski, and J.A. Tenreiro Machado. Possible adaptive control by tangent hyperbolic fixed point transformations used for controlling the Φ^6-type Van der Pol oscillator. *Proc. of the* 6^{th} *IEEE International Conference on Computational Cybernetics (ICCC 2008), November 27–29, 2008, Stará Lesná, Slovakia*, pp. 15–20, 2008.

[17] Jean-Jacques E. Slotine, W. Li. Applied Nonlinear Control. *Prentice Hall International, Inc., Englewood Cliffs, New Jersey*, 1991.

[18] J.K. Tar, I.J. Rudas, Gy. Hermann, J.F. Bitó, and J.A. Tenreiro Machado. On the robustness of the Slotine-Li and the FPT/SVD-based adaptive controllers. *WSEAS Transactions on Systems and Control*, Issue 9, Volume 3, September 2008, pp. 686–700, 2008.

Explaining Recommendations of Factorization-Based Collaborative Filtering Algorithms

István Pilászy[1], Domonkos Tikk[1]

Budapest University of Technology and Economics
Magyar Tudósok krt. 2.
Budapest, Hungary

Abstract: Recommender systems try to predict users' preferences on items, given their feedback. Netflix, a DVD rental company in US, announced the Netflix Prize competition in 2006. In that competition two powerful matrix factorization (MF) algorithms were proposed, one using alternating least squares (ALS), the other using gradient descent. Both approaches aim to give a low-rank representation of the matrix of ratings provided by users on movies.
Recently, an algorithm was proposed to explain predictions of ALS-based recommendations. This algorithm can explain why ALS "thinks" that a particular movie will suit the user's needs, where the explanation consists of those movies previously rated by the user which are most relevant to the given recommendation.
We propose a method that can explain predictions of gradient descent-based MF in the analog way. We validate the proposed method by experimentation.

1. Introduction

The goal of recommender systems is to give personalized recommendation on items to users. Typically the recommendation is based on the former and current activity of the users, and metadata about users and items, if available. Collaborative filtering (CF) methods are based only on the activity of users, while content-based filtering (CBF) methods use only metadata. Hybrid methods try to benefit from both information sources.

Users can express their opinion on items in different ways. They give explicit or implicit feedback on their preferences. The former mainly includes opinion expression via *ratings* of items on a predefined scale, while the latter consists of other user activities, such as purchasing, viewing, renting or searching of items.

[1]Both authors are also affiliated with Gravity Research & Development Ltd., H-1092 Budapest, Kinizsi u. 11., Hungary, info@gravitrd.com

A good recommender system recommends such items that meets the users' need. Measuring the effectiveness of real a recommender system is difficult, since the goal may be for example to maximize the profit of an online DVD-rental company, but in scientific context performance measures evaluate the recommendation systems offline, e.g. by measuring the error of prediction with RMSE (root mean square error) or MAE (mean absolute error).

The importance of a good recommender system was recognized by Netflix, an online DVD-rental company, who announced the Netflix Prize (NP) competition in October 2006. Netflix generously published a part of their rating dataset, which consists of 10^8 ratings of 480 189 customers on 17 770 movies. Ratings are integer numbers on a 1-to-5 scale. In addition to that, the date of the ratings and the title and release year of the movies are also provided. This is currently the largest available CF dataset and it fits onto a compact disc.

The NP competition motivated the development of new CF methods that are able to handle datasets of such a size. Typical running times of algorithms varies between few minutes and a couple of days. The NP competition focuses only on the good prediction performance: the competitors have to minimize the error of predicted ratings measured on a hold-out set. However, real recommender systems also have to address some other problems beside the good predictive performance. In this paper, we will focus on how to explain predictions when the performance measure is RMSE, and the model is found by an alternating least squares (ALS) approach or a gradient descent.

This paper is organized as follows. Section 2 introduces notations. Section 3 surveys related works and introduces the ALS-based and gradient descent based MF algorithms. We investigate explanations for ALS-based factorizations in Section 4, and for gradient descent based factorizations in Section 5. Section 6 describes the experiments.

2. Notation

In this paper we use the following notations:

N: number of users
M: number of movies or items.
$u \in \{1,\ldots,N\}$: index for users
$i,j \in \{1,\ldots,M\}$: indices for movies or items.
r_{ui}: the rating of user u on item i.
$\hat{r}_{ui}$: the prediction of r_{ui}. In general, superscript "hat" denotes the prediction of the given quantity, that is, $\hat{x}$ is the prediction of x.
$\mathbf{R}$: the matrix of r_{ui} values (for both explicit and implicit ratings).
$\mathcal{R}$: for explicit feedback, the set of (u,i) indices of $\mathbf{R}$ where a rating is provided; for implicit feedback: the set of all indices of $\mathbf{R}$.

n_u: number of ratings of user u, i.e. $n_u = |\{i : (u,i) \in \mathcal{R}\}|$
$\mathbf{A}_u$: used for ridge regression, denoting the covariance matrix of input (considering user u in context).
$\mathbf{d}_u$: used for ridge regression, denoting the input-output covariance vector.
$\mathbf{I}$: denotes the identity matrix of the appropriate size.
λ: regularization parameter.
η: learning rate for gradient methods.

3. Related Works

We assume that factorization methods strive to minimize the prediction error in terms of RMSE $= \sqrt{\sum_{(u,i)\in\mathcal{V}}(r_{ui} - \hat{r}_{ui})^2/|\mathcal{V}|}$, where $\mathcal{V}$ is a validation set.

Matrix factorization approaches have been applied successfully for both rating-based and implicit feedback-based CF problems [1, 4, 6, 10, 2, 8, 5]. The goal of MF methods is to approximate the matrix $\mathbf{R}$ as a product of two lower rank matrices: $\mathbf{R} \approx \mathbf{P}\mathbf{Q}^{\mathrm{T}}$, where $\mathbf{P} \in \mathbb{R}^{N\times K}$ is the user feature matrix, $\mathbf{Q} \in \mathbb{R}^{M\times K}$ is the item (or movie) feature matrix, K is the number of features that is a predefined constant, and the approximation is only performed at $(u,i) \in \mathcal{R}$ positions. The r_{ui} element of $\mathbf{R}$ is approximated by

$$\hat{r}_{ui} = \mathbf{p}_u^{\mathrm{T}}\mathbf{q}_i.$$

Here $\mathbf{p}_u \in \mathbb{R}^{K\times 1}$ is the user feature vector, the u-th row of $\mathbf{P}$, and $\mathbf{q}_i \in \mathbb{R}^{K\times 1}$ is the movie feature vector, the i-th row of $\mathbf{Q}$. A good approximation aims to minimize the error of prediction, $e_{ui} = r_{ui} - \hat{r}_{ui}$ while keeping the Euclidean norm of the user and movie feature vectors small. In the Netflix Prize competition, the most commonly used target function focusing on these goals is the following:

$$(\mathbf{P}^*,\mathbf{Q}^*) = \arg\min_{\mathbf{P},\mathbf{Q}} \sum_{(u,i)\in\mathcal{R}} \left(e_{ui}^2 + \lambda\mathbf{p}_u^{\mathrm{T}}\mathbf{p}_u + \lambda\mathbf{q}_i^{\mathrm{T}}\mathbf{q}_i\right).$$

The predefined regularization parameter λ trades off between small training error and small model weights. The optimal value of λ can be determined by trial-and-error using cross-validation.

In the NP competition two different algorithms were found to be very effective to approximately solve the above optimization problem: one based on alternating least squares (ALS), proposed by team BellKor [1] and the other based on incremental gradient descent (also know as stochastic gradient descent) method with biases (BRISMF) proposed by team Gravity [8].

BellKor's alternating least squares approach alternates between two steps: step 1 fixes $\mathbf{P}$ and recomputes $\mathbf{Q}$, step 2 fixes $\mathbf{Q}$ and recomputes $\mathbf{P}$. The recomputation of $\mathbf{P}$ is performed by solving a separate least squares problem for each user: for the u-th user it takes the feature vector ($\mathbf{q}_i$) of movies rated by the user as input variables, and the value of the ratings (r_{ui}) as output variables, and finds the optimal $\mathbf{p}_u$ by ridge regression (RR). BellKor proposed non-negative RR, however, in this paper, when not otherwise stated, ALS and RR refer to the general variants, where negative values are allowed.

Borrowing the notations from [1], let the matrix $\mathbf{Q}[u] \in \mathbb{R}^{n_u \times K}$ denote the restriction of $\mathbf{Q}$ to the movies rated by user u, the vector $\mathbf{r}_u \in \mathbb{R}^{n_u \times 1}$ denote the ratings given by the u-th user to the corresponding movies, and let

$$\mathbf{A}_u = \mathbf{Q}[u]^{\mathrm{T}}\mathbf{Q}[u] = \sum_{i:(u,i)\in\mathcal{R}} \mathbf{q}_i\mathbf{q}_i^{\mathrm{T}}, \quad \mathbf{d}_u = \mathbf{Q}[u]^{\mathrm{T}}\mathbf{r}_u = \sum_{i:(u,i)\in\mathcal{R}} r_{ui}\cdot\mathbf{q}_i. \tag{1}$$

Then RR recomputes $\mathbf{p}_u$ as

$$\mathbf{p}_u = (\lambda n_u \mathbf{I} + \mathbf{A}_u)^{-1}\,\mathbf{d}_u. \tag{2}$$

According to [1, 4], the number of recomputations needed ranges between 10 and a "few tens".

Gravity's BRISMF approach works as follows [8]: the dataset is first ordered by user id, and then by rating date. The update of the model is performed per-rating, not per-user or per-movie. Suppose that we are at the (u,i)-th rating of $\mathbf{R}$. Then compute e_{ui}, the error of the prediction. The update formulae for $\mathbf{p}_u$ and $\mathbf{q}_i$ are the followings:

$$\mathbf{p}'_u = \mathbf{p}_u + \eta\cdot(e_{ui}\cdot\mathbf{q}_i - \lambda\cdot\mathbf{p}_u), \tag{3}$$

$$\mathbf{q}'_i = \mathbf{q}_i + \eta\cdot(e_{ui}\cdot\mathbf{p}_u - \lambda\cdot\mathbf{q}_i) \tag{4}$$

In this way, the weights of the model ($\mathbf{P}$,$\mathbf{Q}$) are modified such that it better predicts the (u,i)-th rating. One iteration over the database requires $O(|\mathcal{R}|\cdot K)$ steps, and the required number of iterations ranges between 1 and 14 [10]. It has been pointed out that larger K yields more accurate predictions [4, 6, 10].

4. Explaining Recommendations Using ALS

Users may wonder why they are recommended with certain items, as recommended items may seem unpreferred at the first sight. Therefore, a good recommender system is able to explain its recommendations and gives hints on why a given product is likely to match the user's taste.

Hu et al. proposed a method to provide explanation for factorization based methods in [4] by explaining predictions. Although the method is for the implicit feedback case, it can be carried over to the explicit feedback as well:

Recall that in eqs. (1)–(2), ALS recalculates $\mathbf{p}_u$ in the following way:

$$\mathbf{p}'_u = (\lambda n_u \mathbf{I} + \mathbf{A}_u)^{-1}\mathbf{d}_u = \left(\lambda n_u \mathbf{I} + \mathbf{Q}[u]^{\mathrm{T}}\mathbf{Q}[u]\right)^{-1}\left(\mathbf{Q}[u]^{\mathrm{T}}\mathbf{r}_u\right)$$

Let us introduce $\mathbf{W}_u = (\lambda n_u \mathbf{I} + \mathbf{A}_u)^{-1}$; here it is rewritten for explicit feedback. Then we have

$$\hat{r}_{ui} = \mathbf{q}_i^{\mathrm{T}}\mathbf{W}_u\mathbf{d}_u = \sum_{j:\ (u,j)\in\mathcal{R}} \mathbf{q}_i^{\mathrm{T}}\mathbf{W}_u\mathbf{q}_j r_{uj}, \tag{5}$$

that is, for each movie that is rated (or watched) by the user, it is known how much is its contribution to the sum. Authors explain $\hat{r}_{ui}$ by those movies that have the highest contribution to the sum. Note the $\mathbf{W}_u\mathbf{q}_j$ vectors can be precomputed in $O(K^2 \cdot n_u)$ (assuming $\mathbf{W}_u$ is available) to speed up the explanation of an arbitrary movie. Also note that when a user changes one of its ratings (i.e. some r_{uj}), neither $\mathbf{W}_u$ nor $\mathbf{W}_u\mathbf{q}_j$ changes, which makes it efficient for such situations.

If $\lambda > 0$, then $\mathbf{W}_u$ is a symmetric positive definite matrix, thus it can be decomposed into $\mathbf{W}_u = \mathbf{V}_u^T\mathbf{V}_u$ where $\mathbf{V}_u \in \mathbb{R}^{K\times K}$. With this notation $\hat{r}_{ui}$ is rewritten as

$$\hat{r}_{ui} = \sum_{j:\ (u,j)\in\mathcal{R}} (\mathbf{V}_u\mathbf{q}_i)^{\mathrm{T}}(\mathbf{V}_u\mathbf{q}_j)r_{uj}.$$

With the above notation, $\mathbf{V}_u$ describes how user u "thinks" about the items, and the user-independent similarity $\mathbf{q}_i^{\mathrm{T}}\mathbf{q}_j$ between two items is replaced with the user-dependent similarity $(\mathbf{V}_u\mathbf{q}_i)^{\mathrm{T}}(\mathbf{V}_u\mathbf{q}_j)$.

4.1. Explaining Recommendations Using Dual Formulation of RR

To calculate $\mathbf{p}_u$, we can use the equivalent dual formulation (Kernel Ridge Regression with linear kernel) involving the Gram matrix $\mathbf{Q}[u]\mathbf{Q}[u]^{\mathrm{T}}$ [5]:

$$\mathbf{p}'_u = \mathbf{Q}[u]^{\mathrm{T}}\left(\lambda n_u \mathbf{I} + \mathbf{Q}[u]\mathbf{Q}[u]^{\mathrm{T}}\right)^{-1}\mathbf{r}_u. \tag{6}$$

Let $\mathbf{A}_u^{\text{dual}} = \mathbf{Q}[u]\mathbf{Q}[u]^{\mathrm{T}} \in \mathbb{R}^{n_u\times n_u}$. Let $\boldsymbol{\alpha}_u = (\lambda n_u \mathbf{I} + \mathbf{A}_u^{\text{dual}})^{-1}\mathbf{r}_u \in \mathbb{R}^{n_u\times 1}$; then the calculation of $\mathbf{p}_u$ is rewritten as: $\mathbf{p}_u = \mathbf{Q}[u]^{\mathrm{T}}\boldsymbol{\alpha}_u$. Let α_{ui} denote that element of $\boldsymbol{\alpha}_u$

which refers to the i-th movie rated by u. With this notation: $\mathbf{p}_u = \sum_{i:(u,i)\in\mathcal{R}} \alpha_{ui} \cdot \mathbf{q}_i$, and

$$\hat{r}_{ui} = \sum_{j:(u,j)\in\mathcal{R}} \alpha_{uj} \cdot \mathbf{q}_i^{\mathrm{T}} \mathbf{q}_j \tag{7}$$

Again, we know the contribution of each movie watched by the user in the sum.

5. Explaining Recommendations Using Incremental Gradient Descent

5.1. Dual formulation

Gravity's BRISMF approach was described in Section 3. In [10] the authors introduce a modification in the learning algorithm termed as "retraining user features". The modified learning scheme does not change the prediction performance significantly.

After obtaining $\mathbf{P}$ and $\mathbf{Q}$ with BRISMF, the algorithm resets $\mathbf{P}$, and reruns the training algorithm, but $\mathbf{Q}$ is now fixed. Let n denote the number of epochs in the second run. Now we show that this modification also helps explaining recommendations: Since $\mathbf{Q}$ does not change in second run, we can recompute $\mathbf{p}_u$ by iterating over the ratings of u for n times. We denote this variant as BRISMF-U. We can rewrite equation (3) as $\mathbf{p}'_u = \mathbf{p}_u \cdot (1 - \eta \cdot \lambda) + \eta \cdot e_{ui} \cdot \mathbf{q}_i$. Thus $\mathbf{p}_u$ is the linear combination of its initial value $\mathbf{p}_u^0$ and the $\mathbf{q}_i$ vectors rated by u:

$$\mathbf{p}_u = \alpha_{u0}\mathbf{p}_u^0 + \sum_{i:\ (u,i)\in\mathcal{R}} \alpha_{ui}\mathbf{q}_i \tag{8}$$

In this way, we can explain recommendations in a similar way as proposed in Section 4.1.

However, after each gradient step all of the α_{ui} values has to be maintained, which means that each needs to be multiplied with $(1 - \eta \cdot \lambda)$, which is slow. To overcome this, we can decompose α_{ui} into $\alpha_{ui} = c_u \cdot \gamma_{ui}$. We initialize with $\gamma_{ui} = 0$ and $c_u = 1$. Then for the gradient step of movie i the efficient version of maintaining α_{ui}-s is the following:

1. compute e_{ui}
2. gradient step: update $\mathbf{p}_u$ according to (3)
3. perform regularization: $c_u := c_u \cdot (1 - \eta \cdot \lambda)$
4. and then with the new c_u value: $\gamma_{ui} := \frac{c_u \cdot \gamma_{ui} + \eta \cdot e_{ui}}{c_u}$

At the end of the n-th epoch, we can compute the $\alpha_{ui} = c_u \cdot \gamma_{ui}$ values. This algorithm does not increase the time complexity of BRISMF-U algorithm.

5.2. Deriving Primal Formulation

ALS is not sensitive to the order of examples, as opposed to gradient methods, namely BRISMF-U. In [8] authors proposed to order examples by u and then by the date of the ratings, to get better RMSE scores. In [9] it has been pointed out that a BRISMF with simple manually set parameters can achieve test RMSE =0.9104 when examples are ordered only by u, and 0.9056 when they are ordered by u and then by the date of the ratings. In the NP competition, for each user, the examples in the test set are newer than in the training set, which is practical, since in a real recommender system the goal is to predict the present preference based on past ratings. Now we propose a method that can explain predictions of BRISMF-U in a way very similar to the explanation method for ALS with primal formulation (Section 4).

One can think at the ordering of the examples by date as giving higher weights (confidence) to newer examples. This leads to the idea that BRISMF-U can be related with weighted ridge regression (WRR), by finding the appropriate weight for each rating. In the followings, we assume user u in context, and further assume that $\alpha_{u0}\mathbf{p}_u^0 = \mathbf{0}$. We will show how to find weights for movies rated by u, such that the result of weighted ridge regression will be (almost) equal to the result of BRISMF-U.

Let c_{ui} denote the weight (confidence) of u on item i. Let $\mathbf{C}_u \in \mathbb{R}^{n_u \times n_u}$ be the diagonal matrix of the c_{ui} values. Let $\mathbf{A}_u = \sum_{i:\,(u,i)\in\mathcal{R}} c_{ui}\mathbf{q}_i\mathbf{q}_i^{\mathrm{T}} = \mathbf{Q}[u]^{\mathrm{T}}\mathbf{C}_u\mathbf{Q}[u]$ and $\mathbf{d}_u = \sum_{i:\,(u,i)\in\mathcal{R}} c_{ui}\mathbf{q}_i r_{ui} = \mathbf{Q}[u]^{\mathrm{T}}\mathbf{C}_u\mathbf{r}_u$ be the weighted covariance matrix and weighted covariance vector resp.

The cost function of weighted ridge regression is the following:

$$\lambda n_u \mathbf{p}_u^{\mathrm{T}}\mathbf{p}_u + \sum_{i:\,(u,i)\in\mathcal{R}} c_{ui}(\mathbf{p}_u^{\mathrm{T}}\mathbf{q}_i - r_{ui})^2 \tag{9}$$

The $\mathbf{p}_u$ minimizing this cost function can be calculated by letting its derivative be equal to zero:

$$2(\lambda n_u\mathbf{I} + \mathbf{A}_u) - 2\mathbf{d}_u = 0 \tag{10}$$

If $\lambda n_u\mathbf{I} + \mathbf{A}_u$ is positive definite, then the unique solution of this equation is the global minimum: $\mathbf{p}_u = (\lambda n_u\mathbf{I} + \mathbf{A}_u)^{-1}\mathbf{d}_u$. This is the case when all $c_{ui} \geq 0$ and $\lambda > 0$. When $\lambda n_u\mathbf{I} + \mathbf{A}_u$ is not positive definite, the solution is not unique, or no solutions exist at all (the function is not bounded from below). In the followings, we assume $\lambda > 0$, but c_{ui} may be arbitrary.

We can compute $\mathbf{p}_u$ using the equivalent dual formulation as well:

$$\mathbf{p}_u = \mathbf{Q}[u]^{\mathrm{T}}\boldsymbol{\alpha}_u, \text{ where } \boldsymbol{\alpha}_u = \mathbf{C}_u(\lambda n_u\mathbf{I} + \mathbf{Q}[u]\mathbf{Q}[u]^{\mathrm{T}}\mathbf{C}_u)^{-1}\mathbf{r}_u. \tag{11}$$

Note that when $\lambda \neq 0$, then:

$$\lambda n_u \mathbf{I} + \mathbf{Q}[u]^{\mathrm{T}} \mathbf{C}_u \mathbf{Q}[u] \text{ is invertible } \iff \lambda n_u \mathbf{I} + \mathbf{Q}[u] \mathbf{Q}[u]^{\mathrm{T}} \mathbf{C}_u \text{ is invertible,}$$

since the eigenvalues of $(\mathbf{Q}[u]^{\mathrm{T}} \mathbf{C}_u)\mathbf{Q}[u]$ and $\mathbf{Q}[u](\mathbf{Q}[u]^{\mathrm{T}} \mathbf{C}_u)$ are the same, apart from the eigenvalue 0. The addition of the term $\lambda n_u \mathbf{I}$ shifts these eigenvalues by λn_u. Thus, after the addition either both the new matrices have the eigenvalue 0, or none, which means that either both are invertible, or none. Furthermore, if $\lambda > 0$, then either both are positive definite, or none.

Note that after running BRISMF-U, we are given with the $\boldsymbol{\alpha}_u$ vector. A possible way to relate it with WRR is to find appropriate c_{ui} values, such that $\boldsymbol{\alpha}_u$ values computed from (11) equals to $\boldsymbol{\alpha}_u$ from BRISMF-U. Consider the above equation for $\boldsymbol{\alpha}_u$. Now we solve it for $\mathbf{C}_u$. First, multiplying it by $\mathbf{C}_u^{-1}$ from left (we will discuss later if $c_{ui} = 0$ for some i), it yields:

$$\mathbf{C}_u^{-1} \boldsymbol{\alpha}_u = (\lambda n_u \mathbf{I} + \mathbf{Q}[u] \mathbf{Q}[u]^{\mathrm{T}} \mathbf{C}_u)^{-1} \mathbf{r}_u$$

Now multiply with $(\cdot)^{-1}$ from left:

$$(\lambda n_u \mathbf{I} + \mathbf{Q}[u] \mathbf{Q}[u]^{\mathrm{T}} \mathbf{C}_u) \mathbf{C}_u^{-1} \boldsymbol{\alpha}_u = \mathbf{r}_u$$

Then it reduces to:

$$\lambda n_u \mathbf{C}_u^{-1} \boldsymbol{\alpha}_u + \mathbf{Q}[u] \mathbf{Q}[u]^{\mathrm{T}} \boldsymbol{\alpha}_u = \mathbf{r}_u$$

Let $\hat{\mathbf{r}}_u = \mathbf{Q}[u] \mathbf{Q}[u]^{\mathrm{T}} \boldsymbol{\alpha}_u$ denote the predictions of BRISMF-U. Then it is rewritten as:

$$\lambda n_u \mathbf{C}_u^{-1} \boldsymbol{\alpha}_u + \hat{\mathbf{r}}_u = \mathbf{r}_u$$

After reordering and multiplying with $\mathbf{C}_u$ it yields:

$$\lambda n_u \boldsymbol{\alpha}_u = \mathbf{C}_u (\mathbf{r}_u - \hat{\mathbf{r}}_u)$$

thus:

$$c_{ui}(r_{ui} - \hat{r}_{ui}) = \lambda n_u \alpha_{ui}. \tag{12}$$

Now let us discuss some special cases of this solution:

- if for some i, $\alpha_{ui} = 0$, then we can set c_{ui} to 0 (substituting it directly into (11) we can see that it is a good solution), and ignore in the above equations those rows and columns of $\mathbf{C}_u$, $\mathbf{Q}[u]$ and $\mathbf{r}_u$ that corresponds to these c_{ui}-s. Note that $c_{ui} = 0$ can occur only if $\alpha_{ui} = 0$ (recall that we assume $\lambda > 0$).

- Since c_{ui} values computed by (12) can be negative, $\lambda n_u \mathbf{I} + \mathbf{A}_u$ might not be positive definite. When $\lambda n_u \mathbf{I} + \mathbf{A}_u$ is not positive definite, then we cannot relate WRR and BRISMF-U in this way: for any c_{ui} values, which makes $\mathbf{A}_u$ positive definite, the $\boldsymbol{\alpha}_u$ computed from (11) will be different from $\boldsymbol{\alpha}_u$ of BRISMF-U.
- If for some i, $r_{ui} - \hat{r}_{ui} = 0$, i.e. that examples are learnt perfectly by BRISMF-U, but $\alpha_{ui} \neq 0$, then no c_{ui} values can satisfy the equation.

If $\lambda n_u \mathbf{I} + \mathbf{A}_u$ is positive definite and $\forall_i : r_{ui} - \hat{r}_{ui} = 0 \Longrightarrow \alpha_{ui} = 0$, then WRR and BRISMF-U can be related, and then we can explain predictions as in (5) by computing $\mathbf{W}_u = (\lambda n_u \mathbf{I} + \mathbf{Q}[u]^{\mathrm{T}} \mathbf{C}_u \mathbf{Q}[u])^{-1}$.

Even if $\lambda n_u \mathbf{I} + \mathbf{A}_u$ is not positive definite, but it is invertible, and all c_{ui}-s are defined by (12), we can still compute $\mathbf{W}_u$ by neglecting the fact that there does not exist an optimal solution for (9) with these c_{ui}-s. When not all c_{ui}-s are defined by (12), we may set $c_{ui} = 0$ for the undefined ones, however, in this case the corresponding α_{ui}-s will not be equal to those of BRISMF-U. We found by experimentation that setting negative and undefined c_{ui}-s to 0 does not affect prediction performance significantly: the largest difference of test RMSE of BRISMF-U and this modified WRR was 0.0002 on the Netflix Prize dataset.

Note that when we are given with $\hat{\mathbf{r}}_u$ and c_{ui}, then α_{ui} can be easily computed from (12), thus, there is no need to invert the regularized Gram matrix to get α_{ui}-s. In case of the original formulation of ALS, all $c_{ui} = 1$, and $\hat{\mathbf{r}}_u$ is also known.

5.3. Comparing Primal and Dual Formulation

Note that eqs. (5) and (7) explains predictions in two different ways. The primal formulation based explanation computes $\mathbf{W}_u$ that can compute a user-dependent similarity of any two items (as it has been noted by Hu et al.). The dual formulation computes α_{uj} values for each item j rated by u, which tells how the user-independent similarity $\mathbf{q}_i^{\mathrm{T}} \mathbf{q}_j$ should be replaced with $\mathbf{q}_i^{\mathrm{T}} \cdot (\alpha_{uj} \mathbf{q}_j)$.

One can think at the primal formulation based explanation that a prediction is a weighed sum of the ratings made by the user, while the dual formulation explains predictions as a weighed sum of training errors. Let us introduce the similarities s^p_{uij} and s^d_{uij}, that tells how similar are i and j, according to u, based on the primal and the dual formulation. Let $s^p_{uij} = \mathbf{q}_i^{\mathrm{T}} \mathbf{W}_u \mathbf{q}_j$ [4]. Then the primal formulation based explanation is rewritten as [4]:

$$\hat{r}_{ui} = \sum_{j:\ (u,j) \in \mathcal{R}} s^p_{uij} r_{uj}. \tag{13}$$

Let $s^d_{uij} = \mathbf{q}_i^{\mathrm{T}} \mathbf{q}_j c_{ui} / (\lambda n_u)$. Then, based on (12), the dual formulation based explanation

is rewritten as:

$$\hat{r}_{ui} = \sum_{j:\ (u,j)\in\mathcal{R}} s^d_{uij}(r_{uj} - \hat{r}_{uj}), \tag{14}$$

which is very unintuitive, since the prediction $\hat{r}_{ui}$ is a linear combination of prediction errors.

6. Experimental results

We perform experiments on the dataset released by Netflix for Netflix Prize, which is almost 2 orders of magnitude larger than previously used benchmarks (EachMovie, GroupLens). Netflix provides two validation sets: the Probe set and the Quiz set. The ratings of Quiz set is only known by Netflix, while the Probe set is released with ratings. For more information on the dataset see e.g. [3].

We evaluate the presented methods on a randomly selected 10% subset of the Probe set, which we refer to as Probe10, or as test set. [2]. The Probe10 set contains 140 840 ratings. We add the rest of the Probe set to the training set. We remark that in former experiments we measured only a slight difference between the RMSE values on the Probe10 and Quiz sets [8, 7, 10].

For the experiments, we use BRISMF#250U mentioned in [10], but with $K = 20$, as a matrix factorization model, which we refer to as BRISMF#20U. Probe10 RMSE of this model is 0.9051. By clipping the predicted value in the range of 1 to 5, it goes down to 0.9045.

6.1. Examining primal formulation for gradient methods

The c_{ui}-s computed from eq. (12) can be negative, or can be positive but too large, or can be even undefined (when $r_{ui} - \hat{r}_{ui} = 0$ and $\alpha_{ui} \neq 0$). We proposed a simple heuristic: set c_{ui} to 0 when $c_{ui} < 0$ or undefined. Intuitively a good matrix factorization algorithm with date-based ordering of examples should not differ too much from ridge regression, which treats all examples equally ($c_{ui} = 1$ for all i). Thus, we also propose to let $c_{ui} := 100$ when $c_{ui} > 100$. These heuristics increased Probe10 RMSE only to 0.9052. When we decreased the upper limit from 100 to 10, the Probe10 RMSE was again 0.9052. Thus, we are able to explain a 0.9051 prediction without involving too large numbers or indefinite optimization.

[2] A Perl script is available at our homepage, `gravityrd.com`, which selects the Probe10 from the original Netflix Probe set to ensure repeatability.

In practice, users are recommended with items having the highest predicted ratings. Matrix factorization is able to predict ratings for all of the unseen movies of u fast: the computational cost is only $O((M - n_u) \cdot K)$. Since the above heuristics does not change prediction performance significantly (but allows to explain predictions of BRISMF-U), we can first compute the top ranked movies, and then deal with the explanation.

We did not observed $r_{ui} - \hat{r}_{ui} = 0$ during testing. However, 2.3% of the c_{ui} values were below 0. The c_{ui} values ranged from $-6 \cdot 10^6$ to $5 \cdot 10^6$. The optimization problem in eq. (9) was not positive definite in 36% of the cases. Note that c_{ui} is computed for each training example, but eq. (9) is solved for each test example.

When all c_{ui}-s are nonegative, then the eigenvalues of $\lambda n_u \mathbf{I}\mathbf{A}_u$ are at least λn_u. We observed, that without the $c_{ui} \geq 0$ restriction that matrix had at least one eigenvalue being less than λn_u in the 76.4% of the cases.

6.2. Comparing different explanation algorithms

We evaluate different explanation algorithms by measuring how many times the most explanatory movies are the same. All methods are based on the result of BRISMF#20U, however, ALS-based methods use only the movie feature matrix $\mathbf{Q}$ of that model. We use the following abbreviations to refer to the methods:

- AP: primal formulation for ALS, according to eq. (5).
- AD: dual formulation for ALS, according to eq. (7).
- GD: dual formulation for gradient methods, like AD, but the α_{ui}-s are provided by the gradient method.
- GP: primal formulation for gradient methods, like AP, but confidence-weighted. Confidences are computed according to eq. (12).

We distinguish between two variants:

- R: using the $s^p_{uij} \cdot r_{uj}$ and $s^d_{uij} \cdot (r_{uj} - \hat{r}_{uj})$ with the highest absolute values, see eqs. (13)–(14).
- N: using only the s^p_{uij} and s^d_{uij} with the highest absolute values,

Results are summarized on Table. 1. The intent of this table is to give information about how the different methods are related to each other. We will examine their ability to provide reasonable explanations later.

Interestingly, GPR and APR are the most similar methods, however, Probe10 RMSE of ALS was only 0.9320 Although ALS could perform much better, we did not fine-tune the

	APR	GPR	APN	GPN	GDR	GDN
APR	100					
GPR	71	100				
APN	56	46	100			
GPN	37	42	48	100		
GDR	14	17	16	20	100	
GDN	9	8	10	14	6	100

Table 1: Comparing different explanation methods for the same model. Numbers indicate the percentage of how often two methods ranked the same movie as the most explanatory for a prediction, on the test set.

learning parameters, we used the same as BRISMF#20U. Both APR and APN were able to explain similarly to GPR, despite of the huge difference in prediction performance.

GDN is an outlier, as it has the least common most-explanatory movies with other methods. This may be due to the large numbers involved in the computation of s^d_{uij} caused by small training errors.

From the table, we can conclude that primal-based methods are similar to each other (similarity is always $> 40\%$), while the dual based methods are quite different: similarity between primal and dual based methods are always $< 20\%$.

We also examined manually the explanations of the proposed methods. First, we looked how the movies of the *Matrix* trilogy (*The Matrix*, *The Matrix: Reloaded* and *The Matrix: Revolutions*) are explained. We examined also a variant of BRISMF#20U, where K is 63. In this case, Probe10 RMSE is 0.8988 for the gradient method, and 0.9253 for the ALS method. We refer to the $K = 63$ variant by appending "63" to the method name. There were 254 cases out of the 140840, when one of the above 3 *Matrix* movies was to be predicted. Results are summarized on Table. 2.

Both GP and AP methods had similar success in explaining a *Matrix* movie by a *Matrix* movie. The numbers for the dual methods are poor, and reflect that they are not good in explaining predictions.

Second, we examined a popular TV-series, the *Friends*, which aired for 10 seasons. We examined how often a *Friends* movie is explained by a *Friends* movie. In the Netflix Prize database there are 9 DVDs for the 9 out of 10 seasons, and there are also "best of" DVDs. Together, it is 17 DVDs with 443 ratings in the Probe10 set. Again, we found that the primal formulation based methods list these movies amongst the top explanatory movies much frequently, than the dual based methods. Third, we examined the *Die Hard* trilogy,

Movie collection	n	APR	GPR	APN	GPN	GDR	GDN	APR63	GPR63	GDR63
The Matrix	254	137	131	136	135	38	28	154	154	79
Friends	443	370	359	376	376	218	243	372	367	250
Die Hard	131	46	44	42	17	13	12	55	48	25

Table 2: This table indicates how many times a prediction for a movie from a set of similar movies is mostly explained by a rating on a movie from the same set, e.g. a prediction of a *Matrix* movie is top-explained by a rating on a *Matrix* movie. n is the total number of such movies in the test set (Probe10).

with similar conclusions.

A main drawback of the above comparison is that a simple neighbor method will reach the highest numbers, explaining *Matrix* movies always by *Matrix* movies in the first place. However, such an explanation method is unpersonalized, i.e. the s_{uij} similarities do not depend on u.

It may occur that a user rates *The Matrix: Revolutions* as a 3, and *The Matrix: Reloaded* as a 5. When the recommendation system is recommending her *The Matrix* as a 5, GPN and APN does not take into account that one is higher-rated than the other. It may seem strange, that the system is recommending *The Matrix*, because the user rated *The Matrix: Revolutions* as a 3. GPR and APR take this information into account. However, when we change the rating system from "5 is the best" to "1 is the best", this advantage turns into disadvantage, since the system is going to recommend low-predicted (thus good) movies by high-rated (thus bad) movies.

7. Conclusion

Hu et al. [4] showed how the predictions of weighted ridge regression can be explained, where the explanation lists the most influental training examples regarding the predicted example.

In this work we introduced how to explain predictions of gradient descent based learning methods. Although the presented methods are applicable to any kind of gradient descent based learning (e.g. text document categorization), we put the work in the context of the Netflix Prize dataset and explanation of movie recommendations.

We showed how to relate gradient descent methods to weighted ridge regression (WRR), by assigning appropriate importance weights to the training examples. Given a model generated by gradient descent, we can almost always find importance weights such that

a weighed ridge regression with these weights will yield exactly the same linar model. However, in some situations it is impossible to relate them, or the weights are too large. For these cases, we proposed a simple heuristics: set negative weights to zero, and clamp the too large values. We showed by experimentation that in this case the prediction of the weighted ridge regression only slightly differs from the prediction of the gradient method.

We also proposed two other explanation methods based on the dual formulation of gradient methods and WRR, but these methods did not proved to be efficient in providing reasonable explanations.

8. Future work

In the Netflix Prize competition, blending multiple predictions is very common to get better RMSE [2]. When the blending is linear, and the prediction of the algorithms can be decomposed as a summation over the movies rated by the user, then the blended prediction can be decomposed as well, enabling the explanation of the prediction. In the Netflix Prize competition many approach were proposed that can inherently explain predictions (neighbor methods, NSVD1 methods). ALS and gradient gescent methods can also be explained. The blending of these approaches may result in a much better prediction. An important question to investigate: is the correct explanation of a better prediction a better explanation?

The explanation method of Hu et al. decomposes a prediction into the following form:

$$\hat{r}_{ui} = \sum_{j:\,(u,j)\in\mathcal{R}} s_{uij} r_{uj}. \tag{15}$$

Note that this equation is a weighted nearest neighbour approach, where the s_{uij} values are to be defined. They can be computed via:

$$s_{uij} = (\mathbf{V}_u \mathbf{q}_i)^{\mathrm{T}} (\mathbf{V}_u \mathbf{q}_j) \tag{16}$$

Where $\mathbf{q}_i$ is the feature vector of the active example (active item, for which the rating is to be predicted), and $\mathbf{q}_j$ is the feature vector of the j-th training example. The calculation of $\mathbf{V}_u$ is defined in Section 4.

Our proposed method of explaining predictions of gradient descent method relies only on the dual formulation of the method, i.e. the weight vector of the model is a linear combination of the training examples. Many linear model has a dual representation like Adaline (gradient descent in the Netflix Prize competition), Perceptron, linear Support Vector Machines (SVM), ridge regression, centroid classifier, etc.

This implies, that, for example, in a text categorization task where the classifier is a linear SVM, we are able to explain the category prediction of an unseen document. A traiditional approach to explain is to query documents that are similar according to Euclidean or cosine distance. By computing $\mathbf{V}_u$ from the dual representation of the underlying linear model, we can transform the documents from the original vector space into a vector space, where the above weighted nearest neighbour method gives the same predictions, as the underlying linear model. An important question: are these explanations better than the traditional cosine distance based ones?

References

[1] R. M. Bell and Y. Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *Proc of. ICDM-07, 7th IEEE Int. Conf. on Data Mining*, pages 43–52, Omaha, Nebraska, USA, 2007.

[2] R. M. Bell, Y. Koren, and Ch. Volinsky. The BellKor solution to the Netflix Prize. Technical Report, AT&T Labs Research, 2007. `http://www.netflixprize.com/assets/ProgressPrize2007_KorBell.pdf`.

[3] J. Bennett, Ch. Eklan, B. Liu, P. Smyth, and D. Tikk. KDD Cup and Workshop 2007. *ACM SIGKDD Explorations Newsletter*, 9(2):51–52, 2007.

[4] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Proc. of ICDM-08, 8th IEEE Int. Conf. on Data Mining*, pages 263–272, Pisa, Italy, 2008.

[5] A. Paterek. Improving regularized singular value decomposition for collaborative filtering. In *Proc. of KDD Cup Workshop at SIGKDD-07, 13th ACM Int. Conf. on Knowledge Discovery and Data Mining*, pages 39–42, San Jose, California, USA, 2007.

[6] G. Takács, I. Pilászy, B. Németh, and D. Tikk. On the Gravity recommendation system. In *Proc. of KDD Cup Workshop at SIGKDD-07, 13th ACM Int. Conf. on Knowledge Discovery and Data Mining*, pages 22–30, San Jose, California, USA, 2007.

[7] G. Takács, I. Pilászy, B. Németh, and D. Tikk. Matrix factorization and neighbor based algoritms for the Netflix Prize problem. In *Proc. of RecSys-08, ACM Conf. on Recomender Systems*, pages 267–274, Lausanne, Switzerland, 2008.

[8] G. Takács, I. Pilászy, B. Németh, and D. Tikk. A unified approach of factor models and neighbor based methods for large recommender systems. In *Proc. of ICADIWT-*

08, 1st IEEE Workshop on Recommender Systems and Personalized Retrieval, pages 186–191, August 2008.

[9] Gábor Takács, István Pilászy, Bottyán Németh, and Domonkos Tikk. Scalable collaborative filtering approaches for large recommender systems. *Journal of Machine Learning Research*, 10:623–656, 2009.

[10] Gábor Takács, István Pilászy, Bottyán Németh, and Domonkos Tikk. Investigation of various matrix factorization methods for large recommender systems. In *2nd Netflix-KDD Workshop*, Las Vegas, NV, USA, August 24, 2008.

Fuzzy Linear Systems Applied to Leontief Input-Output Model

Pasi Luukka[a] and Jorma K. Mattila[a]

[a] *Laboratory of Applied Mathematics,*
Lappeenranta University of Technology
P.O. Box 20, FIN-53851 Lappeenranta, Finland,

Abstract

A general fuzzy linear system is investigated using fuzzy numbers and Gauss-Seidel iteration formula. We have used our fuzzy linear system to solve Leontief input-output model with fuzzy entries. When solving Leontief input-output model one is usually making the assumption that we know entirely the consumption matrix from industrial entries and we are certain about the final demand. These assumptions however depend heavily on estimates and information received from the industry and hence in these estimates, uncertainty plays a crucial role. To address this type of uncertainty fuzzy methods are needed to model this and in this article we are giving a procedure to solve this problem. Numerical examples are given to illustrate the procedure. Among them also the famous example from Leontief himself where he solved the production levels for U.S. economy in 1958.

Key words: Fuzzy Linear System, Leontief input-output model, fuzzy numbers, Gauss-Seidel, SOR

1 Introduction

Linear systems played an essential role in the Nobel prize-winning work of Wassily Leontief. The economic model described by him is the basis for more elaborate models now in many parts of the world.

We introduce the topic by basing it on David C. Lay's text book [1]. Suppose the nation's economy is divided into n sectors that produce goods or services, and let $\mathbf{x}$ be a *production vector* in $\mathbb{R}^n$ that lists the output of each sector for one year. Also, suppose another part of the economy (called the *open sector*) does not produce goods or services but only consumes them, and let $\mathbf{d}$ be a *final demand vector* (or *bill of final demands*) that lists the value of the goods and services demanded

from the various sectors by the nonproductive part of the economy. The vector $\mathbf{d}$ can represent consumer demand, government consumption, surplus production, exports, or other external demand.

As the various sectors produce goods to meet consumer demand, the produces themselves create additional *intermediate demand* $\mathbf{i}$ for goods they need as inputs for their own production. The interrelations between the sectors is very complex, and the connection between the final demand and the production is unclear. Leontief asked if there is a production level $\mathbf{x}$ such that the amounts produced (or "supplied") will exactly balance the total demand for that production, so that

$$\mathbf{x} = \mathbf{i} + \mathbf{d} \tag{1}$$

The basic assumption of Leontief's input-output model is that for each sector, there is a *unit consumption vector* in $\mathbb{R}^n$ that lists the inputs needed *per unit of output* of the sector. All input and output units are measured in millions of dollars, rather that in quantities such as tons or bushels. Prices of goods and services are held constant.

Example 1.1 Suppose the economy consists of three sectors – manufacturing, agriculture, and services – with unit consumption vectors $\mathbf{c}_1$, $\mathbf{c}_2$, $\mathbf{c}_3$ shown in the table below:

Purchased from	Manufacturing	Agriculture	Services
Manufacturing	0.50	0.40	0.20
Agriculture	0.20	0.30	0.10
Services	0.10	0.10	0.30

The columns of manufacturing, agriculture, and services consist of inputs consumed per unit of output. The manufacturing column is $\mathbf{c}_1$, the agriculture column is $\mathbf{c}_2$, and the services column is $\mathbf{c}_3$. What amounts will be consumed by the manufacturing sector if it decides to produce 100 units?

For the solution, compute

$$100\mathbf{c}_1 = 100 \begin{pmatrix} 0.50 \\ 0.20 \\ 0.10 \end{pmatrix} = \begin{pmatrix} 50 \\ 20 \\ 10 \end{pmatrix}$$

To produce 100 units, manufacturing will order (i.e., "demand") and consume 50 units from other parts of the manufacturing sector, 20 units from agriculture, and 10 units from services.

2 Leontief input-output model

Input-output tables (I/O-table, for short) form a coherent frame for keeping of accounts for describing itinerant commodity floods in national economy. From the I/O-table, the *establishing balance equation*

$$x_i = x_{i1} + x_{i2} + \ldots + x_{in} + y_i \quad (i = 1, \ldots, n) \tag{2}$$

appears. This shows that the whole production x_i of the each sector is used as intermediate inputs $x_{i1}, x_{i2}, \ldots, x_{in}$ in the sectors $1, 2, \ldots, n$ and partly to the final output y_i. (Notice that in some sector i, x_{ii} may differ from zero, the all other x_{ij}'s may be equal to zero, or y_i may be equal to zero.)

The *input coefficients* are

$$a_{ij} = \frac{x_{ij}}{x_j} \tag{3}$$

where x_{ij} stands for the use of products from a sector i as input in a sector j and x_j stands for the total production in a sector j.

From the input coefficients a_{ij} we form the *consumption matrix*

$$A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{pmatrix} \tag{4}$$

The *basic assumption* of the input-output model is that the input coefficients are fixed, i.e., the "recipe" of the model is supposed to keep constant regardless of the production amount. Using input coefficients, the establishing balance equation (2) can be put into the form

$$x_i = a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n + y_i \quad (i = 1, \ldots, n). \tag{5}$$

These equations can be put into the matrix form

$$X = AX + Y \tag{6}$$

where

$$X = (x_1, \ldots, x_n)^{\mathrm{T}} \quad \text{and} \quad Y = (y_1, \ldots, y_n)^{\mathrm{T}}.$$

Input-output model can be used in several tasks. Some of them are

(a) to determine total production of each sectors and transactions between them if the final demand is known,

(b) to determine the need of basic inputs if the final demand is known,
(c) to analyze the relational change of prices and to find out the expense construction.

In the case (a), we want to find out all accumulating influences, if a certain amount of products is produced for final demand (producing bread needs corn, producing corn needs tractors, producing tractors needs metal, workers of metal industry need bread etc.). Mathematically, this means that Y is known in the equation (6) and we want to calculate X. The solution can be found as follows. First we put (6) into the form

$$Y = X - AX = (I - A)X, \tag{7}$$

from which we have

$$X = (I - A)^{-1}Y. \tag{8}$$

Substituting the vector $Y = (0, 0, \ldots, 0, 1, 0, \ldots, 0)$ to (7) where the jth component equals to 1 and other components are equal to zero, we see that the element $\widehat{a}_{kj}$ of the matrix $(I-A)^{-1}$ has a relevant interpretation. The element $\widehat{a}_{kj}$ expresses how much together we need to produce, taking into consideration all the accumulating influences, in order to produce one unit for final demand in the sector j.

Example 2.1 *Determine the total demand X for industry sectors* 1, 2, *and* 3 *if*

$$A = \begin{pmatrix} 0,3 & 0,4 & 0,1 \\ 0,5 & 0,2 & 0,6 \\ 0,1 & 0,3 & 0,1 \end{pmatrix} \quad \textit{and} \quad Y = \begin{pmatrix} 20 \\ 10 \\ 30 \end{pmatrix}.$$

Hence, by (6) we have

$$\begin{gathered} X - AX = Y \iff (I - A)X = Y \iff \\ X = (I - A)^{-1}Y \implies \end{gathered} \tag{9}$$

$$\begin{aligned} (I - A) &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0,3 & 0,4 & 0,1 \\ 0,5 & 0,2 & 0,6 \\ 0,1 & 0,3 & 0,1 \end{pmatrix} \\ &= \begin{pmatrix} 0,7 & -0,4 & -0,1 \\ -0,5 & 0,8 & -0,6 \\ -0,1 & -0,3 & 0,9 \end{pmatrix} \implies \end{aligned} \tag{10}$$

$$(I-A)^{-1} = \frac{1}{0,151}\begin{pmatrix} 0,54 & 0,39 & 0,32 \\ 0,51 & 0,62 & 0,47 \\ 0,23 & 0,25 & 0,36 \end{pmatrix} \Longrightarrow$$

$$X = \frac{1}{0,151}\begin{pmatrix} 0,54 & 0,39 & 0,32 \\ 0,51 & 0,62 & 0,47 \\ 0,23 & 0,25 & 0,36 \end{pmatrix}\begin{pmatrix} 20 \\ 10 \\ 30 \end{pmatrix}$$

$$= \begin{pmatrix} 160,93 \\ 201,99 \\ 118,54 \end{pmatrix}.$$

The result is $x_1 = 160,93$, $x_2 = 201,99$ *and* $x_3 = 118,54$.

In the case (b), the need of basic inputs, when the final demand is known, is considered as follows. First, to find out the total production in different sectors, we proceed in the similar way as is done in the case (a). After this, the *coefficient matrix of basic inputs* created in the same way as the matrix A is used. A *basic input coefficient* is defined as

$$d_{ij} = \frac{z_{ij}}{x_j} \tag{11}$$

where z_{ij} stands for the use of the basic input i in the sector j and x_j stands for the total demand of the sector j. Hence, the coefficient matrix of basic inputs is

$$D = \begin{pmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2n} \\ \vdots & \vdots & & \vdots \\ d_{n1} & d_{n2} & \dots & d_{nn} \end{pmatrix}. \tag{12}$$

If Z is a basic input vector then we have

$$Z = D(I-A)^{-1}Y. \tag{13}$$

Also here, an element $\widehat{d}_{kj}$ of the matrix $D(I-A)^{-1}$ has a relevant interpretation. The element $\widehat{d}_{kj}$ indicates the total amount of the basic input k in order to produce one unit for the final demand j.

In the case (c), it is thought that the price of each product is composed of the costs of used intermediate inputs and basic inputs. The price equation now corresponding

to establishing balance equation (2) is

$$p_j = a_{1j}p_1 + a_{2j}p_2 + \ldots + a_{nj}p_n + w_j \qquad (j = 1, \ldots, n), \tag{14}$$

where p_j is the unit price of production of the sector j and w_j is the unit costs of basic inputs of the sector j. The equation (14) in matrix form is

$$P = A^{\mathrm{T}}P + W. \tag{15}$$

If we suppose that W is known, P then has the form

$$P = (I - A^{\mathrm{T}})^{-1}W. \tag{16}$$

The model we just introduced is crisp. As we noticed above, the basic assumption of the model is that the input coefficients are fixed and hence, the core of model is constant. One reason for this is, that the matrix A, is in practice very big. But this assumption causes sometimes some troubles. To avoid them, one way may be to fuzzify A. So, the rest of the paper considers possible fuzzy tools for solving the problem by constructing a fuzzy model where the matrix A consists of fuzzy numbers.

3 Fuzzy Linear Systems

Fuzzy linear systems can occur in several research fields, e.g. in control problems, statistics, physics, engineering, information, economics, and finance science. In [2] following fuzzy linear system (FLS) was considered,

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = y_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = y_2 \\ \qquad\qquad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = y_n \end{cases} \tag{17}$$

where their coefficient matrix $A = (a_{ij})$ was considered to be crisp matrix and $\mathbf{y} = (\mathrm{y_i})$ was a fuzzy vector, $1 \leq i, j \leq n$. They solved this type of fuzzy system using embedding method. In this article our fuzzy linear system differs from [2] so that we consider the coefficient matrix $A = (a_{ij})$ to be fuzzy coefficients also instead of just $\mathbf{y}$ vector to be fuzzy. We solve this problem by using arithmetic operations for left-right(LR)-type fuzzy numbers and apply them to Gauss-Seidel algorithm. Based on Friedman et al [2] work many numerical methods [3–13] have been presented for this type of FLS. In this article we present a way to solve FLS

for the case where also coefficient matrix $A = (a_{ij})$ is considered to be fuzzy. The concept of fuzzy numbers and arithmetic operations with these numbers were first introduced and investigated by Zadeh [14,15]. The Gauss-Seidel algorithm was first applied to FLS with crips coefficient matrix A in [5]. This method was modified to cover Successive Over Relaxation (SOR) method [6] later. In this article also coefficient matrix A is fuzzy and we apply Gauss-Seidel algorithm to solve our FLS and use LR-type fuzzy numbers and arithmetic operations to them introduce by Dubois & Prade [16]. We also present how this can be modified to Successive Over Relaxation (SOR) method. We apply our method to solve Leontief input-output model.

We are solving

$$AX = Y$$

where A is matrix and X and Y are vectors. We assume that instead of A and Y being crips they are considered to consist of fuzzy numbers. We use left-right-type fuzzy numbers and Gauss-Seidel method to solve X for fuzzy numbers. This way we also receive fuzzy support area for X.

Gauss-seidel iteration formula is

$$x_i^{k+1} = \frac{1}{a_{ii}} \left(y_i - \sum_{j=1}^{i-1} a_{ij} x_j^{k+1} - \sum_{j=i+1}^{N} a_{ij} x_j^k \right) \tag{18}$$

$i = 1, ..., N, a_{ii} \neq 0$. We are using Gauss-seidel iteration because with this we can simply use extented addition, difference, extented product and extented quotient introduced in [16] which are relatively straightforward to implement. We are using left-right type fuzzy numbers, which can be written as

$$\mu_A(x) = \begin{cases} L((M-x)/l) \text{ if } x \leq M \\ R((x-M)/r) \text{ if } x \geq M \end{cases} \tag{19}$$

for $l > 0, r > 0$. For L-R-type function we used $L(x) = R(x) = max\{1 - x, 0\}$.

We used following extensions: For fuzzy number $A_1 = (M_1, l_1, r_1)_{LR}$, M_1 is the modal value, l_1 is left support length and r_1 is right support length. and same for $A_2 = (M_2, l_2, r_2)_{LR}$.

Extented sum:

$$A_1 \oplus A_2 = (M_1 + M_2, l_1 + l_2, r_1 + r_2)_{LR} \tag{20}$$

Extented difference:

$$A_1 \ominus A_2 = (M_1 - M_2, l_1 + r_2, r_1 + l_2)_{LR} \tag{21}$$

The extented product:

When $M_1 > 0$ and $M_2 > 0$

$$A_1 \odot A_2 \approx (M_1 M_2, M_1 l_2 + M_2 l_1, M_1 r_2 + M_2 r_1)_{LR} \tag{22}$$

When $M_1 < 0$ and $M_2 > 0$

$$A_1 \odot A_2 \approx (M_1 M_2, M_2 l_1 - M_1 r_2, M_2 r_1 - M_1 l_2)_{RL} \tag{23}$$

if spreads are small w.r.t. M_1 and M_2, and instead of equation (22)

$$A_1 \odot A_2 \approx (M_1 M_2, M_1 l_2 + M_2 l_1 - l_1 l_2, M_1 r_2 + M_2 r_1 - r_1 r_2)_{LR} \tag{24}$$

should be used if spreads are not small w.r.t. M_1 and M_2. Similarly with equation (23), see more about these product rules e.g. in Dubois and Prades book [16]. Notice that we do not need to deal with cases when $M_2 < 0$ when solving production levels using Leontief's I/O model with fuzzy entries since production levels can not be negative. This basically means that when solving $AX = Y$, X which is production level vector we can not have negative production level values.

The extented quotient can be expressed as

$$A_1 \oslash A_2 \approx (\frac{M_1}{M_2}, \frac{r_2 M_1 + l_1 M_2}{M_2^2}, \frac{l_2 M_1 + r_1 M_2}{M_2^2})_{LR} \tag{25}$$

Using these LR-type fuzzy operations our fuzzy Gauss-Seidel iteration formula looks like

$$x_i^{k+1} = \left(y_i \ominus \left(\oplus_{j=1}^{i-1} a_{ij} \odot x_j^{k+1}\right) \ominus \left(\oplus_{j=i+1}^{N} a_{ij} \odot x_j^{k}\right)\right) \oslash a_{ii} \tag{26}$$

Usually $||\cdot||_\infty$ is used for the ending criterion to iterations. Here we also apply $\infty-$norm but now we extended it so that it also takes into account the support areas. So $||X^{k+1} - X^k||_\infty < \epsilon$ is now extented to $max\{||M^{k+1} - M^k||_\infty, \quad ||l^{k+1} - l^k||_\infty, ||r^{k+1} - r^k||_\infty\} < \epsilon$. This change guarantees that also support areas are converged. This Gauss-Seidel iteration formula was implemented using Matlab^{TM}

Modification to Successive Over Relaxation (SOR) method

Gauss-Seidel method can be modified to relaxation parameter. Modifying Gauss-Seidel method to include this relaxation parameter is often also called as Successive Over Relaxation (SOR) method [17]. When applying Gauss-Seidel method the components for the change $\epsilon = x^{k+1} - x^k$ are usually with same sign without depending on iteration number k. In this case one can accelerate the iteration process so that instead of giving the result from Gauss-Seidel iteration

$$\tilde{x}_i^{k+1} = \frac{1}{a_{ii}} \left(y_i - \sum_{j=1}^{i-1} a_{ij} x_j^{k+1} - \sum_{j=i+1}^{N} a_{ij} x_j^k \right) \tag{27}$$

the iteration result is given in

$$x_i^{k+1} = x_i^k + \omega(\tilde{x}_i^{k+1} - x_i^k) \tag{28}$$

where the normal result $\epsilon_i = \tilde{x}_i^{k+1} - x_i^k$, which gives the result $x_i^{k+1} = x_i^k + \epsilon_i = \tilde{x}_i^{k+1}$, is enhanced by coefficient ω. Value $\omega = 1$ gives the usual Gauss-Seidel method. When $\omega < 1$, we have under relaxation and when $\omega > 1$ we have over relaxation. This relaxation method converges when $\omega \in (0, 2)$. Optimal ω is here found by trial. As a computational formula for this method we get

$$x_i^{k+1} = (1-\omega)x_i^k + \frac{\omega}{a_{ii}} \left(y_i - \sum_{j=1}^{i-1} a_{ij} x_j^{k+1} - \sum_{j=i+1}^{N} a_{ij} x_j^k \right) \tag{29}$$

Now when using above mentioned fuzzy arithmetics, besided previously introduced formulas we also need scalar multiplication which is defined as

$$\omega A_1 = (\omega M_1, |\omega| l_1, |\omega| r_1)_{LR} \tag{30}$$

and now fuzzy SOR iteration formula is

$$x_i^{k+1} = (1-\omega)x_i^k \oplus \omega \left(y_i \ominus \left(\oplus_{j=1}^{i-1} a_{ij} \odot x_j^{k+1} \right) \ominus \left(\oplus_{j=i+1}^{N} a_{ij} \odot x_j^k \right) \right) \oslash a_{ii} \tag{31}$$

Now notice that since scalar multiplication is defined as above we suggest that relaxation parameter is now chosen so that $\omega \in (0, 1]$.

4 Results

We illustrate the performance of our method using three different examples.

Example 1: Consider the economy whose consumption matrix is given as

$$C = \begin{bmatrix} 0.5 & 0.4 & 0.2 \\ 0.2 & 0.3 & 0.1 \\ 0.1 & 0.1 & 0.3 \end{bmatrix}$$

Suppose the final demand is 50 units for manufacturing, 30 units for agriculture, and 20 units for services. Assume that these entries are fuzzy numbers and left and right fuzzy support length equaling $l = r = 0.01$. Next we need to find the production level x that will satisfy this demand. In Table 1 there are results from this experiment with crisp case and also with fuzzy entries. In Figure 1 we have also plotted the membership values for our fuzzy production level.

Table 1
Production levels solve with crisp solution in left and solution with fuzzy entries on the right. Solution for fuzzy entries is given in form X=[M,l,r] where M is modal value and l and r are left and right support.

X **crisp**	X **fuzzy**
225.93	[225.93 25.08 25.08]
118.51	[118.52 14.89 14.89]
77.78	[77.78 11.76 11.76]

Example 2: Steel manufacturer, who controlled 30% of the markets, wants to investigate how the change in demand for car industry effects the demand in steel manufacturing industry. To simplify the example only car industry, steel industry, railroad industry and mining industry is considered. We assumme that entries (A) were calculated for this year and are given in Table 2. Demand for the end product was this year $Y_1 = (10, 100, 30, 10)$. For the next year it was predicted that demand for cars would grow by 20% and demand for others would stay the same, so $Y_2 = (10, 120, 30, 10)$. Needed production levels can then be solved by $X_i = (I - A)^{-1}Y_i, \quad i = 1, 2$. where I is identity matrix.

Solving this for crisp case for both years we received results reported in Table 3.

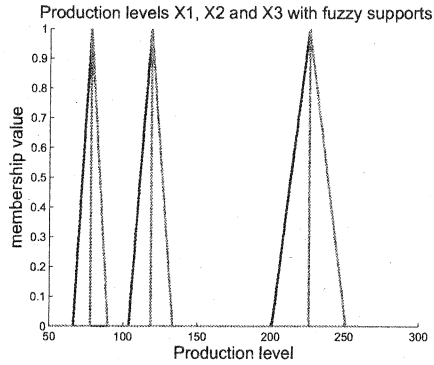


Fig. 1. Fuzzy production levels in example 1, when consumption matrix and final demand is considered to be fuzzy.

Table 2
Industries entries

	steel industry	car industry	railroad industry	mining industry
steel industry	0.1	0.4	0.1	0.1
car industry	0	0.1	0.4	0.2
railroad ind.	0.2	0.1	0	0.2
mining ind.	0.3	0	0	0

Table 3
Production levels for this year and predictions for next

	X_1	X_2
steel industry	90.0	102.3
car industry	150.7	176.0
railroad industry	70.4	76.0
mining industry	37.0	40.7

Next we solved this with fuzzy matrix A and vector Y with left and right fuzzy support length equaling $l = r = 0.05$ for all (a_{ij}) values in fuzzy matrix A. We received results reported in Table 4. Results are given so that first modal value is given then left support and then right support. Results are also plotted in Figure 2 a and 2 b. In Figure 2 c we calculated the differences in production levels for these two years. As can be seen from the figure uncertainties in the predictions can effect quite much the final production level growth.

As we can see the modal values are about the same as in crisp cases. What is notable is how much the support areas in matrix A are influencing the support areas of our solution X. This clearly shows that one needs to take into account the uncertainty that can exists in A and how they are influencing the solution. To study

Table 4
Production levels for this year and predictions for next year with fuzzy supports

	X_1	X_2
steel industry	[90.0 11.6 11.6]	[102.5 51.5 51.5]
car industry	[150.7 11.3 11.3]	[176.3 44.8 44.8]
railroad industry	[70.5 11.2 11.2]	[76.3 41.2 41.2]
mining industry	[37.0 14.0 14.0]	[40.7 33.2 33.2]

Table 5
Differences in spreads if spread is double in car industry sector

	$X2_{orig}$	$X2_{double}$
steel industry	[102.5 51.5 51.5]	[102.5 60.2 60.2]
car industry	[176.3 44.8 44.8]	[176.3 65.5 65.5]
railroad industry	[76.3 41.2 41.2]	[76.3 45.5 45.5]
mining industry	[40.7 33.2 33.2]	[40.7 35.6 35.6]

more about how different support areas are influencing the results we decided to double the support area to all a_{ij} values for one row and keep everything else as in previous experiment. In Table 5 one can see how doubling the support area in car industry and keeping the support areas the same for other industries influences the results. In second column is the original results and in third column the results from this experiment. These results are also plotted in Figure 2 d. As can be seen support areas are growing in all cases when fuzziness is increasing but much less for railroad industry and mining industry than car industry.

As our last example to demonstrate how our method can be applied in Leontief's input output model we consider the famous case, which Leontief solved for U.S. economy data in 1958. This time of course, we do it with fuzzy case.

Example 3: The consumption matrix C below is based on input-output data for the U.S. economy in 1958, with data for 81 sectors grouped into 7 larger sectors: (1) nonmetal household and personal products, (2) final metal products (such as motor vehicles), (3) basic metal products and mining, (4) basic nonmetal products and agriculture, (5) energy (6) services, and (7) entertainment and miscellaneous products [18]. We need to find the production levels to satisfy the final demand y_1. (Units are in millions of dollars.)

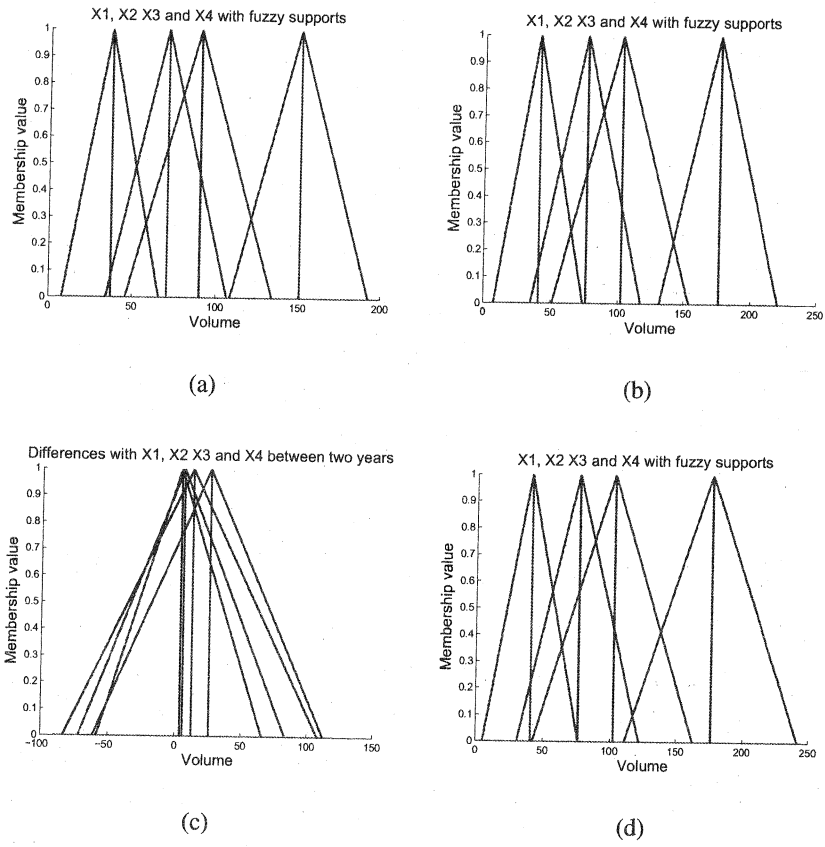


Fig. 2. Production levels with fuzzy spreads a) for year 1, b)predictions for the year 2, c) differences between predictions for next year and production levels for this year, and d) predictions for the year 2 when car industry is considered to have double amount the uncertainty than in others.

$$C = \begin{bmatrix} 0.1588 & 0.0064 & 0.0025 & 0.0304 & 0.0014 & 0.0083 & 0.1594 \\ 0.0057 & 0.2645 & 0.0436 & 0.0099 & 0.0083 & 0.0201 & 0.3413 \\ 0.0264 & 0.1506 & 0.3557 & 0.0139 & 0.0142 & 0.0070 & 0.0236 \\ 0.3299 & 0.0565 & 0.0495 & 0.3636 & 0.0204 & 0.0483 & 0.0649 \\ 0.0089 & 0.0081 & 0.0333 & 0.0295 & 0.3412 & 0.0237 & 0.0020 \\ 0.1190 & 0.0901 & 0.0996 & 0.1260 & 0.1722 & 0.2368 & 0.3369 \\ 0.0063 & 0.0126 & 0.0196 & 0.0098 & 0.0064 & 0.0132 & 0.0012 \end{bmatrix}$$

$y_1 = [74000 \quad 56000 \quad 10500 \quad 25000 \quad 17500 \quad 196000 \quad 5000].$

Table 6
Production levels for this year and predictions for next year with fuzzy supports

X_{1958} crisp	X_{1958} with fuzzy entries	X_{1964}	X_{1964} with fuzzy entries
99576	[99576 6151 6151]	134034	[134034 8283 8283]
97703	[97703 8584 8584]	131686	[131686 11560 11560]
51231	[51231 9012 9012]	69472	[69472 12136 12136]
131570	[131570 12494 12494]	176912	[176912 16825 16825]
49488	[49488 7587 7587]	66596	[66596 10217 10217]
329554	[329554 13995 13995]	443773	[443773 18847 18847]
13835	[13835 4550 4550]	18431	[18431 6127 6127]

We solve production levels for this problem using left and right support values as $l = r = 0.005$. In Table 6 are the production levels solved with fuzzy I/O-model. Leontief also solved production levels for year 1964 using the same consuption matrix C with demand vector y_2 now being

$$y_2 = [99640 \quad 75548 \quad 14444 \quad 33501 \quad 23527 \quad 263985 \quad 6526].$$

We repeated this experiment with out fuzzy entries and the solution with this demand vector is also given in Table 6. As can be seen we calculated the production levels and predicted production levels for U.S. economy and take the uncertainty involved, quite easily with our proposed method.

5 Discussion

In this article we applied Gauss-Seidel iterative method and SOR interative method to approximate of the unique solution for fuzzy linear system. We have applied this solution to solve Leontief input-output model. This seems practical and gives valuable information since in the cases presented here, the consumption matrix is inherently fuzzy. This is especially true in cases where it is used to approximated production levels for the next year as we have demonstrated in our examples. We can even calculate approximations going further than just next year as our last example showed. Here also the final demand is clearly fuzzy. Using fuzzy I/O-model we managed to approximate the production levels needed for production in the year in question and also estimate production levels for year to come and succesfully calculated the support area when the consumption matrix and final demand

was considered to be fuzzy numbers.

References

[1] Lay, D. C., *Linear Algebra and Its Applications*, 2nd ed., Addison Wesley Longman, Inc., 2000.

[2] Friedman, M., Ming, M. and Kandel, A., Fuzzy linear systems, Fuzzy Sets and Systems. 96 (1998), pp. 201-209.

[3] Abbasbandy, S. Ezzati, A. and Jafarian, A. LU decomposition method for solving fuzzy system of linear equations, Appl. Math. Comput. 172 (2006), pp. 633-643.

[4] Abbasbandy, S., Jafarian, A. and Ezzati, A. Conjugate gradien method for fuzzy symmetric positive definite system of linear equations, Appl. Math. Comput. 171 (2005), pp. 1184-1191.

[5] Allahviranloo, T., Numerical methods for fuzzy system of linear equations, Appl. Math. Comput. 155 (2004), pp. 493-502

[6] Allahviranloo, T., Successive over relaxation iterative method for fuzzy system of linear equations, Appl. Math. Comput. 162 (2005), pp. 189-196.

[7] Allahviranloo, T., The Adomian decomposition method for fuzzy system of linear equations. Appl. Math. Comput. 163 (2005), pp. 553-562.

[8] Allahviranloo, T. Ahmady, E., Ahmady, N. & Alketaby, K.S., Block Jacobi two-stage method with Gauss-Sidel inner iterations for fuzzy system of linear equations, Applied Mathematics and Computation, vol 175, 2, (2006), pp. 1217-1228.

[9] Yin, J.F. & Wang, K. Splitting iterative methods for fuzzy system of linear equations, Computational Mathematics and Modelling 20, 3, (2009), pp. 326-335.

[10] Wang, K. Chen, G. & Wei, Y., Perturation analysis for a class of fuzzy linear systems, Journal of Computational and Applied Mathematics, 224, 1, (2009), pp. 54-65.

[11] Yeh, C.T., Reduction of fuzzy linear systems of dual equations, International Journal of Fuzzy Systems 9, 3, (2007), pp. 173-178.

[12] Zheng, B. & Wang, K., General fuzzy linear systems, Applied Mathematics and Computation 181, 2, (2006), pp. 1276-1286.

[13] Wang, K. Zheng, B., Symmetric successive overrelaxation methods for fuzzy linear systems, Applied Mathematics and Computation 175, 2, (2006), pp. 891-901.

[14] Zadeh, L.A., The concept of a linguistic variable and its application to approximate reasoning, Inform. Sci. 8 (1975), pp. 199-249.

[15] Chang, S.L. and Zadeh, L.A., On fuzzy mapping and control, IEEE Trans. Syst. Man Cyb. 2 (1972), pp. 30-34.

[16] Dubois, D. and Prade, H. (1980). Fuzzy Sets and Systems. Theory and Applications. Academic Press, New York.

[17] Mäkelä, M. Nevalinna, O. & Virkkunen, J. (1982), Numeerinen matematiikka, Mäntän Kirjapaino Oy, Finland.

[18] W.W. Leontief, The Structure of the U.S. Economy, Scientific American, April 1965, pp. 30-32.

Studies on Hysteresis Characteristics of Fuzzy Muller-C Logic Models

Péter Keresztes

Department of Automation, Széchenyi István University,
H-9026 Győr, Egyetem tér 1, Hungary
e-mail: keresztp@sze.hu

Abstract: The crisp-logic implementation of Muller-C, *(CMC)* which plays a fundamental role in delay insensitive logic circuits, is a classical, Huffman-type asynchronous network, i.e. a single output combinational logic network with direct feedback. The most typical feature of *CMC* is the so called *logic-hysteresis*. The aim of the work presented in this paper was to extend the structure of *CMC* to a new architecture, which consists of various-norm fuzzy logic elements, and investigate the behaviour from the point of view of hysteresis-like operation. Based on the mathematical formules describing the next-state functions of various norms n-input fuzzy Muller-C *(FMC^n)* units were modeled by a simple concurrent signal assignment statement, and simulated. The different norm fuzzy-union and intersection functions were implemented in a VHDL package. The simulation results are presented and discussed in the paper. The *Zadeh*-norm Muller-C does not have hysteresis like behaviour, but it shows an other interesting feature, which is also presented. Last point of the paper contains important advices for implementation of fuzzy *Muller-C* models with crisp-logic components.

Keywords: fuzzy logic asynchronous networks, fuzzy Muller-C, crisp logic hysteresis, fuzzy logic hysteresis, dual rail logic, delay insensitive logic circuits.

6. Introduction: The concept of fuzzy-logic hysteresis based on extension of crisp-logic Muller-C asynchronous circuit

One of the basic principle of the implementation of delay insensitive logic is the dual rail representation of logical variables. The various input number Muller-C elements are the basic parts of the dual rail combinational logic and storage circuits [2], [4], [6]. Muller-C elements are single output combinational networks with feedback, and they can be described with the Huffmann model of the asynchronous crisp logic networks The n-input crisp Muller-C *(CMC^n)* can be defined with its NEXT-STATE function as follows:

$$y' = v(x_1, x_2, \ldots x_n, y) = x_1x_2 \ldots x_n + x_1y + x_2y + \ldots + x_ny$$

where $x_1, x_2, \ldots x_n$ are the input variables, y is the current y' is the next state.

Logical hysteresis is a common property of the CMC^n units. If all inputs are 0, this results the stable state $y' = 0$. The condition of setting the state $y = 1$ is, that all inputs have to rise to level 1. Returning to the state $y = 0$ is possible if and only if all inputs fall back to the level 0.

The aim of the work presented in this paper was the extension of the NEXT-STATE function of CMC^n to fuzzy systems with the well known fuzzy logic norms, and the investigation of these models focusing on the hysteresis-like behaviour. In order to detect a hysteresis-like operation, a definition for the hysteresis has to be introduced so that it must not be limited to the *n*-input fuzzy Muller-C models (FMC^n). Since the Muller-C structures consisiting of fuzzy logic elements can be considedered as a member of the family of fuzzy flip-flops, the work presented in this paper is fitting in the series of papers [1], [3], [5].

For the definition of the hysteresis a classification of state transitions of fuzzy asynchronous systems is needed.

7. Classification of states and state-transitions in fuzzy asynchronous models, and definition of the fuzzy hysteresis

An attemption to give a simulation oriented definition for hysteresis-like behaviour of a single output fuzzy logic asynchronous unit is detailed int he following points. Classification of the observable states of a fuzzy asynchronous logic system is the first task. There are two main groups of the observable states, transient and not-transient states. The non-transient states can be stable and quasistable. Corresponding to these type of states, themselves the state-transitions also can be classified.

7.1. Simulation model of a single state-output fuzzy asynchronous network

The FMC^n models can be considered single state-output asynchronous networks, the NEXT-STATE function $v(x_1, x_2, x_n, y)$ of which are expressed by fuzzy operations. If the tool of observation of the behaviour is a VHDL simulation, the NEXT-STATE function can easily be modeled with a simple inertia delay concurrent signal assignment, as follows:

$$y <= v(x_1, x_2, x_n, y) \text{ after } Td;$$

7.2. Non-transient states and their classification based on simulation experiments

If a new input vector is applied on a single state-output fuzzy asynchronous model, and the run time of simulation is long enough, a *tape of states* will be induced. There are three cases on the observed tape of states. If for any long running time the states of the tape following each other are different values, the tape consists of fully *transient states*. If there is an ending part of the tape, which contains equal value states, or a final state value appears before the running time is expired, and there are no more lines on the list, these states are called *non-transient*. In the first case the states of the ending part are considered a single state called *quasistable state*, and in the second case the final state of the tape is called *stable state.*

7.3. Observable state-transitions in single state-output asynchronous fuzzy models

In the following examples, let $y^{(0)}$ be a current state in time T_0.

- If the simulaton is run for more than T_d with unchanged input values $x_1, x_2, \ldots x_n$, and does not appear a new line on the list, $y^{(0)}$ is stable. In a stable state

$$v(x_1, x_2, \ldots y^{(0)}) = y^{(0)}.$$

- If the simulation is run for longer than $2T_d$ with a new vector of inputs $(x_1, x_2, \ldots x_n)$, state $y^{(0)}$ is a stable one, and the only one new line on the simulation list is addressed by time T_0+T_d, and the new state is y' in the new line, a *single step stable-stable transition* is observed. In this case

$$v(x_1, x_2, \ldots x_n\, y^{(0)}) = y^{(1)}) \text{ and } v(x_1, x_2, \ldots x_n\, y^{(1)}) = y^{(1)}$$

- Let k be a natural. If the simulation is run for longer than $(k+1)T_d$ with a new vector of inputs $x_1, x_2, \ldots x_n$ from a non-transient $y^{(0)}$, and the new line addressed by time $T_0 + kT_d$ is the last one on the simulation list, for the state of this line it is true that

$$v(x_1, x_2, \ldots x_n, y^{(k-1)}) = y^{(k)} \text{ and } v(x_1, x_2, \ldots x_n, y^{(k)}) = y^{(k)},$$

  a *multistep stable-stable or quasistatic-stable transition* is observed.

- Let $k_1 \leq k_2 \leq k_3$ are naturals. Run the simulation longer than k_2T_d with a new vector of inputs $x_1, x_2, \ldots x_n$ from the non-transient state $y^{(0)}$. If from the new line k_1 to new line k_2 the state does not change, $y^{(k2)}$ is a *quasistatic state*, and a *stable-quasistatic transition* is observed. If the running time is prolonged to k_3T_d, and $k_3 \rightarrow \infty$,

$$v(x_1, x_2, \ldots x_n\, y^{(k3+1)}) - v(x_1, x_2, \ldots x_n\, y^{(k3)}) \rightarrow 0,$$

  The reason of the particiular form of this list is that the simulation cannot display the difference between the successive states from the line k_1.

Let us introduce a new notation: If it exists, let $y^*\ (x_1, x_2 \ldots x_n, y)$ be the *final non-transient state* on the tape of states from y with $x_1, x_2, \ldots x_n$. It follows from the above that a final non-transient state can be stable or quasistable. Summarize the various state transitions. A single step stable-stable state transition is observed if there are not transient states between two stable states, a multistep stable-stable state-transition is observed, if there is at least one transient state between two stable states, and a stable-quasistatic state-transition can be observed, if a limit-value exists in the series of transient states.

7.4. Definition of hysteresis for n-input, single state output fuzzy asynchronous models

The following definition was found and applied:

An n-input, single-output fuzzy asynchronous model with a next-state function $v(x_1, x_2, \ldots x_n, y)$ is considered to have hysteresis-like operation if a sub-interval $[x_{min}, x_{max}]$ of [0,1] exists, for each x value of it is true that

- $y = 0$ and $y = 1$ states are stable,

- both $y^*(x, x, \ldots x, 1)$ and $y^*(x, x, \ldots x, 0)$ exist, and
- $y^*(x, x, \ldots x, 1) > y^*(x, x, \ldots x, 0)$.

There is an advantage of this approach for the definition of fuzzy hysteresis: Regardless of the number of inputs the hysteresis can easily be drawn up on the two dimension plane $x_1=x_2 = \ldots = x_n = x$ of the $n+1$ dimension space. Another advantage of the approach is that the measure of the hysteresis can easily be determined, so the various norm solutions can easily be compared considering the measure of the hysteresis.

8. The NEXT-STATE function and simulation model of FMCns

Denote the various norms of fuzzy intersection and union with symbols $FAND_{\square}$ and $FOR_{\square}$ respectively, the NEXT-STATE function CMC^n can be given with the following expression:

$$y' = FOR\square(FAND\square(x, x, \ldots x), FAND\square(x, y), FAND\square(x, y), \ldots FAND\square(x, y))$$

The lower case square symbol has to be substituted for by characters specifying the given norms. A concurrent signal assignment statement with inertia type delay T_d models the behaviour the FMC^n in the VHDL simulation:

y <= FOR□(FAND□ (x, x, . . . x),FAND□(x, y),FAND□(x, y), . .FAND□(x, y)) after Td;

It is true for the models CMC^n that $n \geq 2$. The number of inputs can be extended to $n=1$ for *FMC* models. The NEXT-STATE function and the concurrent signal assignment of FMC^1 are given as follows:

$$y' = FOR\square(x, FAND\square(x, y)), \quad y <= FOR\square(x, FAND\square(x, y)) \text{ after } Td;$$

9. Classification of various norm FMC^n models from the point of view of hysteresis

The FMC^n models having hysteresis which corresponds to the definition detailed above are easily selected out from the tables presented below. The tables 1-4 shows the values of state-output, which were the results of $x = 0$, of $x = 0.5$ during the up-magnetisation process, of $x = 1$ in the end of up-magnetisation, and then of $x = 0.5$ again during the down-magnetisation, and of the final $x = 0$. In the cases of norms with parameters, the values of parameters are given in the headings of the tables.

Table 1. Behaviour of FMC^1 models with only one intermediate value (0.5) in the ascending and descending section of "magnetization"

	x	y_Z	y_L	y_Y $w = 2$	y_D $\alpha = 2$	y_A	y_H $v = 2$	y_{DP} $d = 0.5$	y_{SS} $p = 2$
↑	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	0.5	**0.5**	**0.5**	*0.625*	*0.5599*	*0.6667*	*0.6972*	**0.5**	**0.5**
	1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
↓	0.5	**0.5**	**1**	*0.625*	*0.5599*	*0.6667*	*0.6972*	**0.5**	**1**
	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Table 2. Behaviour of FMC[2] models with only one intermediate value (0.5) in the ascending- and descending section of "magnetization"

	$x_1 = x_2 = x$	y_Z	y_L	y_Y $w = 2$	y_D $\alpha = 2$	y_A	y_H $v = 2$	y_{DP} $\alpha = 2$	y_{SS} $p = 2$
↑	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	0.5	**0.5**	**0**	*0.5388*	*0.5716*	*0.6667*	*0.6416*	**0.5**	**0**
	1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
↓	0.5	**0.5**	**1**	*0.5388*	*0.5716*	*0.6667*	*0.6416*	**0.5**	**1**
	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Table 3. Behaviour of FMC[3] models with only one intermediate value (0.5) in the ascending- and descending section of "magnetization"

	$x_1 = x_2 = x_3 = x$	y_Z	y_L	y_Y $w = 2$	y_D $\alpha = 2$	y_A	y_H $v = 2$	y_{DP} $\alpha = 2$	y_{SS} $p = 2$
↑	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	0.5	**0.5**	**0**	**0.134**	*0.6111*	*0.8201*	*0.8897*	**0.5**	**0**
	1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
↓	0.5	**0.5**	**1**	*0.8260*	*0.6111*	*0.8201*	*0.8897*	**0.5**	**1**
	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Table 4. Behaviour of FMC[4] models with only one intermediate value (0.5) in the ascending- and descending section of "magnetization"

	$x_1 = x_2 = x_3 = x_4 = x$	y_Z	y_L	y_Y $w = 2$	y_D $\alpha = 2$	y_A	y_H $v = 2$	y_{DP} $\alpha = 2$	y_{SS} $p = 2$
↑	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	0.5	**0.5**	**0**	**0**	*0.6458*	*0.9204*	*0.9710*	**0.5**	**0**
	1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
↓	0.5	**0.5**	**1**	**1**	*0.6458*	*0.9204*	*0.9710*	**0.5**	**1**
	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Notations used in the tables:

1. The values of stable states are written with bold, the values of quasistable states are written in italics.
2. The lines marked with symbol "↑" contain the data of ascending, the lines marked with symbol "↓" contain the data of descending part

The following consequences and additional tasks can be derived from the tables:

1. Hysteresis-like beaviour corresponding the applied definition is found in all *Lukasiewicz* and *Schweizer-Sklar*, and the 3- and 4-input *Yager* models.

2. Only the *Lukasiewicz Schweizer-Sklar* models show stable-states at $x = 0.5$. The goal of refined simulation processes is to investigate, wether these models show quasistable states at different x values
3. Also with refined simulations to be investigated the dependence of hysteresis on the parameters of the norms.
4. Additional simulations are needed to investigate how the measure of hysteresis depends on the number of inputs.
5. From the point of view of application the *Lukasiewicz* and *Schweizer-Sklar* models seem to be the most important. An additional task is to determine the breaking points of these hysteresis curves with mathematical analysis.

In the following chapters and points of the paper these problems and questions will be solved and answered respectively.

10. The Lukasiewicz and Schwezer-Sklar norm *FMC*^{n}s.

In the following figures (*Figures 1-5*) the ascending- and descending curves of the *FMC*^{n} models are shown, which have hyteresis-like behaviour.

It can be seen that the measure of the *Lukasiewicz* norm is effectively zero, but the rest of the models verify expectations. It is true for them, that the greater the value of the measure of hysteresis the higher that of the number of inputs goes. Each curve can be considered to consist of seven sections, limited by the pairs of points 0-1, 1-2, 2-3, 3-4, 4-5, 5-6 and 6-0. There are several common properties of these sections as follows:

- The state represented by the section 0-1 is the stable state $y = 0$.
- The transitions between the states of section 1-2 are single step stable-stable transitions.
- The transition from the stable state of point 2 to the state of point 3 is a multistep stable-stable transition.
- The state represented by the sections 3-4 and 4-5 is the stable state $y = 1$.
- The transition from the stable state of point 5 to the state of point 6 is a multistep stable-stable transition.
- The state represented by the section 6-0 is the stable state $y = 0$.

In the characteristics of the models given in *Figure 1*.a and *3.a*, the points 0 and 1 are located to $x = 0$. (0-1 shrinks into one point at $x = 0$), and in the characteristic of the model given in *Figure 5* the points 1 and 2 are located to the same value of x.

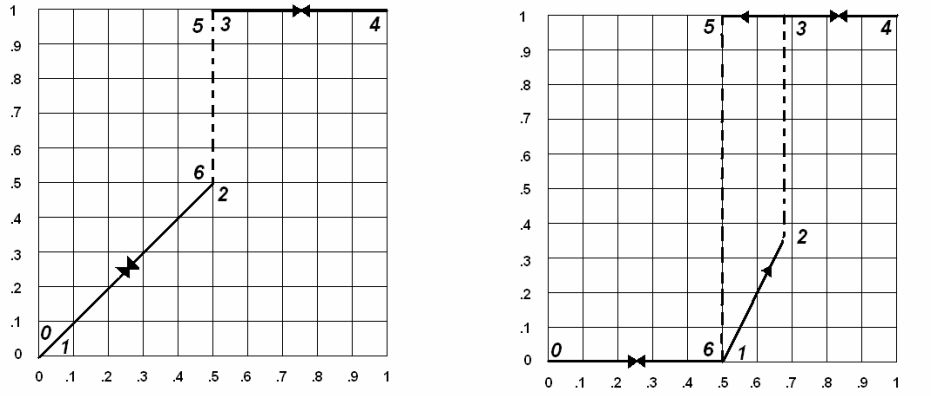


Figure 1. Characteristics of single input (a) and two-input (b) Lukasiewicz-norm FMCs

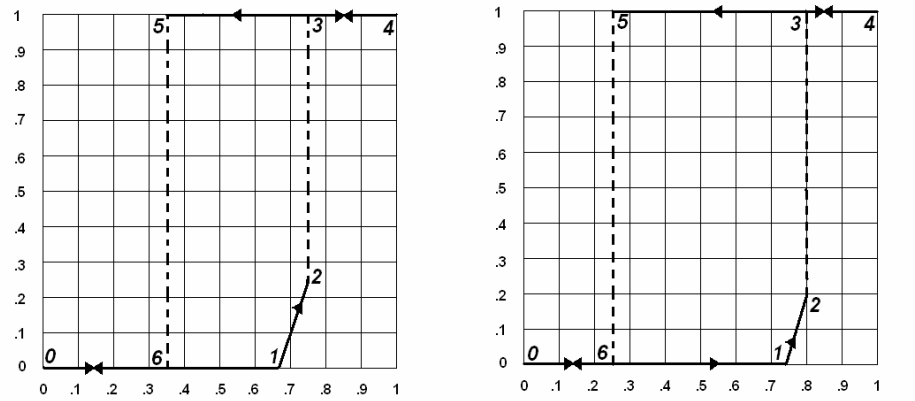


Figure 2. Characteristics of 3 input (a) and 4 input (b) Lukasiewic- norm FMCs

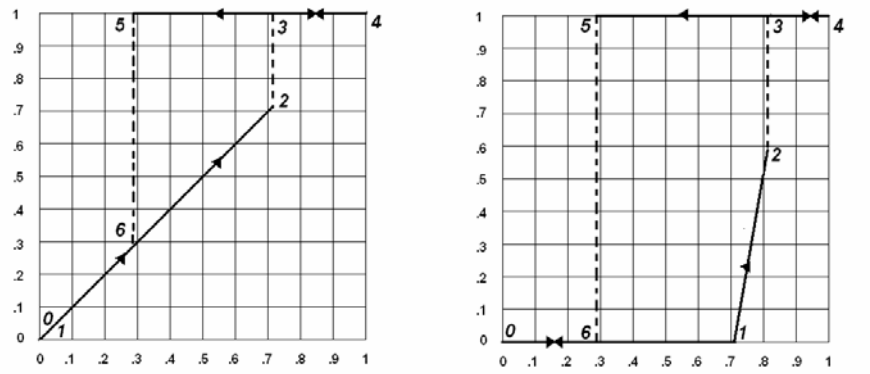


Figure 3. Characteristics of single input (a) and two-input (b) Shcweizer-Sklar-norm FMCs

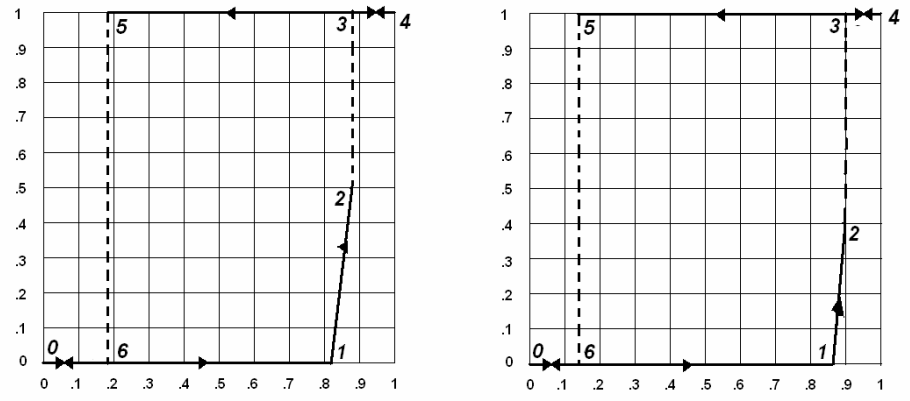


Figure 4. Characteristics of 3 input (a) and 4 input (b) Shweizer-Skla- norm FMCs

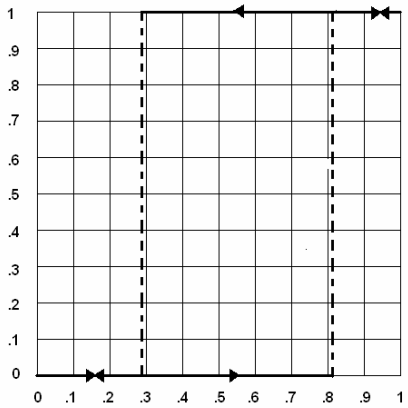


Figure 5. Characteristic of two-input Schweizer-Sklar-norm FMC with enhaced parameters p = 4 and p = 8).

11. Determination of the breaking points of the hysteresis curves of Lukasiewicz and Schweizer-Sklar models

The breaking points of the hysteresis curves of the *Lukasiewicz* and *Schweizer-Sklar* FMC^n models were determined by analysis and/or simulation. The results are shown in the following tables. (*Tables 5-7.*) Both the analysis and the simulation has shown that only the members of these two groups are free of the quasistatic states, i.e. exclusively single- and multi-step stable state transitions can be observed in their operation.

Table 5. Breaking points of the Lukasiewicz models

	FMC_L^1		FMC_L^2		FMC_L^3		FMC_L^4	
	x1 x2	***y***	***x1 x2***	***y***	***x1 x2***	***y***	***x1 x2***	***y***
0-1	0 1/2	x	0 1/2	0	0 2/3	0	0 3/4	0
1-2	1/2	-	1/2 2/3	2x–1	2/3 3/4	3x–2	3/4 4/5	4x–3
2-3	1/2	-	2/3 2/3	-	3/4 3/4	-	4/5 4/5	-
3-4	1/2 1	1	2/3 1	1	3/4 1	1	4/5 1	1
4-5	1 1/2	1	1 1/2	1	3/4 1/3	1	1 1/4	1
5-6	1/2 1/2	-	1/2 1/2	-	1/3 1/3	-	1/4 1/4	-
6-0	1/2 0	x	1/2 0	0	1/3 0	0	1/4 0	0

Table 6. Breaking points of the 1 and 2 input Schweizer-Sklar models

	FMC_{SS}^1		FMC_{SS}^2	
	x1 x2	***y***	***x1 x2***	***y***
0-1	0 $\sqrt{1/2}$	x	0 $1/\sqrt{2}$	0
1-2	$\sqrt{1/2}$	-	$1/\sqrt{2}$ $\sqrt{(2/3)}$	$\sqrt{(2x^2-1)}$
2-3	$\sqrt{1/2}$	-	$\sqrt{(2/3)}$ $\sqrt{(2/3)}$	-
3-4	$\sqrt{1/2}$ 1	1	$\sqrt{(2/3)}$ 1	1
4-5	1 $1-\sqrt{1/2}$	1	1 $1-1/\sqrt{2}$	1
5-6	$1-\sqrt{1/2}$ $1-\sqrt{1/2}$	-	$1-1/\sqrt{2}$ $1-1/\sqrt{2}$	-
6-0	$1-\sqrt{1/2}$ 0	0	$1-1/\sqrt{2}$ 0	0

Table 7. Breaking points of the 3 and 4 input Schweizer-Sklar models

	FMC_{SS}^3		FMC_{SS}^4	
	x1 x2	***y***	***x1 x2***	***y***
0-1	0 $\sqrt{(2/3)}$	0	0 $\sqrt{(3/4)}$	0
1-2	$\sqrt{(2/3)}$ $\sqrt{(3/4)}$	$\sqrt{(3x^2-2)}$	$\sqrt{(3/4)}$ $\sqrt{(4/5)}$	$\sqrt{(4x^2-3)}$
2-3	$\sqrt{(3/4)}$ $\sqrt{(3/4)}$	-	$\sqrt{(4/5)}$ $\sqrt{(4/5)}$	-
3-4	$\sqrt{(3/4)}$ 1	1	$\sqrt{(4/5)}$ 1	1
4-5	1 $1-\sqrt{(2/3)}$	1	1 $1-\sqrt{(3/4)}$	1
5-6	$1-\sqrt{(2/3)}$ $1-\sqrt{(2/3)}$	-	$1-\sqrt{(3/4)}$ $1-\sqrt{(3/4)}$	-
6-0	$1-\sqrt{(2/3)}$ 0	0	$1-\sqrt{(3/4)}$ 0	0

12. About the implementations of FMC^n models

The multi-step stable-stable state transitions render both synchronous and asynchronous implementations of FMC^n models more difficult. The multi-step transitions demand more than one clock cycles in the synchronous, and multiple-delay time transitions in the asynchronous implementations. Since the appearing of a stable state after a multi-step transition has to be detected, both solutions must contain a module, which detects the appearing of a stable states, i.e. it generates a signal, if two successive same-value states have appeared.

If the hardware environment of the synchronous implementation is ordered under a global clock, The FMC^n unit can communicate with its environment under the control of a global timing and control unit (an FSM, with control signals)

If the FMC^n implementation is ordeder under a local-clock, the communication can be solved with the application of the classical four-phase return-to-zero hand-shaking, with with *request* and *acknowledge* input and output signals.

An additional problem has to be solved in the case of the asynchronous implementation. Since the bit-vector representation of the fuzzy logic variables involves different delay-time feed-back loops, *critical race-situations* can lead to a wrong operation. It means, that delay insensitive logic has to be applied, for example *dual-rail combinational* and *register* elements.

The used formules for the various fuzzy intersection and union norms with the used parameter values are shown in the Appendix.

13. Peculiar behaviour of the FMC^n model with Zadeh-norms.

The Zadeh-norm FMC^n models have not hysteresis, but they show a very interesting property in their operation. If we analyze or simulate the model described by the expression

$$y' = FOR_Z(FAND_Z(x_1. . . x_n), FAND_Z(x_1, y). . .FAND_Z(x_n, y)),$$

the following state-transitions will be derived:

- If it is true for the new input vector (x1,. . . xn) that min(x1,. . .xn) ≤ y ≤ max(x1, . . xn), then y' = y,
- If it is true for the new input vector (x1,. . . xn) that y ≤ min (x1, . . xn), then y' = min (x1, . . xn),
- If it is true for the new input vector (x1,. . . xn) that y ≥ max (x1, . . xn), then y' = max (x1, . . xn).

All state transitions are single input stable-stable type.

The Zadeh-norm FMC_Z^n can be described as a paragon of conservatism. If its present state is inside of the interval [$min(x_1,. . . x_n)$, $max(x_1, . . . x_n)$] the state will not change. Otherwise, if the value of its present state is less than the minimum input value, the state will stick to the minimum-value input, but if the value of its present state is more then the maximum input value, the state will stick to the maximum-value input. Thus the FMC_Z^n follows the changes on its input in a very conservative way.

Conclusions

Running the VHDL simulation for the *FMC* models, the classification of the observable states and the state transitions was successful. For the current states $0 < y < 1$ the next states can be stable, quasistable and transient. The observable state transitions from a stable state *y* which is more than 0 and less than 1 can be classified as single step stable-stable, multi step stable-stable and multi step stable-quasistable. Among the investigated fuzzy logic Muller-C models only the *Lukasiewicz*- and the *Schweizer-Sklar*-norm models have exclusively stable-stable state transiton hysteresis-like behaviour. The others do not show hysteresis, or many observed state transitions lead to quasistable states. The *Zadeh*-norm model does not have a hysteresis-like behaviour, but has different sorts of interesting properties. Through the multistep state-transitions, the

hardware-implementation of the *Lukasiewicz* and *Schweizer-Sklar* models, consisting of classical crisp logic elements, has to contain a special unit to ensure the detection of the stable states.

References

[1] Choi, B. Tipnis, K.: *New Components for Building Fuzzy Logic Circuits,* Proc. of the 4th Int Conf. on Fuzzy Systems and Knowledge Discovery, Vol. 2., (2007), pp. 586-590.

[2] Lavagno, L., Sangiovanni-Vincentelli, A.: *Algorithms for synthesis and testing of asynchronous circuits*, Kluwer Academic Publishers, (1993), pp. 18-25.

[3] Lovassy, R, Kóczy, L. T., Gál, L.: *Analyzing Fuzzy Flip-Flops Based on Various Fuzzy Operations*, Acta Technica Jaurinensis, Vol. 1. No. 3. (2008), pp. 447-465.

[4] Muller, D. E., Bartky, W. C.: *A theory of asynchronous circuits,* Annals of Computing Laboratory of Harward University, (1959), pp. 204-243.

[5] Ozawa, K., Hirota, K., Kóczy, L. T.: *Fuzzy flip-flop*, In : Patyra, M. J., Mlynek, D. M. (eds.), Fuzzy Logic. Implementation and Applications, Wiley, Chichester (1996), pp. 197-336.

[6] Sparso, J., Furber, S.: *Principles of asynchronous circuit design – A system perspective*, Kluwer Academic Publisher, (2001), pp. 11-13.

Appendix

Author	Intersection of a and b (FAND)	Union of a, and b (FOR)
Zadeh	$min(a, b)$	$max(a, b)$
Lukasiewicz	$max(a+b-1, 0)$	$min(a+b, 1)$
Yager	$1–min[1, ((1–a)^w + (1–b)^w)^{1/w}]$ $0 < w <\infty$ $w = 2^*$	$min(1, (a^w + b^w)^{1/w})$ $0 < w <\infty$ $w = 2^*$
Hamacher	$ab/(r + (1–r)(a+b – ab))$ $0 < r < \infty$ $r = 2^*$	$(a+b– (2–r)ab)/(1– (1–r)ab)$ $0 < r < \infty$ $r = 2^*$
Dombi	$1/[1+((1/a –1)^\alpha +(1/a–1)^\alpha)^{1/\alpha}]$ $0 < \alpha < \infty$ $\alpha = 2^*$	$1/[1+((1/a –1)^{-\alpha} +(1/a –1)^{-\alpha})^{-1/\alpha}]$ $0 < \alpha < \infty$ $\alpha = 2^*$
Dubois-Prade	$ab/max(a, b, \alpha)$ $0 \le \alpha \le 1$ $\alpha = 0.5^*$	$(a+b–ab–min(a,b,1–\alpha))/max(1–a,1–b,\alpha)$ $0 \le \alpha \le 1$ $\alpha = 0.5^*$
Algebraic	ab	$a+b – ab$
Schweizer-Sklar	$\{ max(0, a^p + b^p –1) \}^{1/p}$ $p\neq0$ $p = 2, 4, 8^*$	$1– \{ max(0, (1–a)^p +(1– b)^p–1 \}^{1/p}$ $p\neq0$ $p = 2, 4, 8^*$

* *Value of the parameter which were applied*

Reconfigurable Mixed-Signal Neural Network with Embedded Visual Sensing

Tamás Zeffer, Timót Hidvégi

The Faculty of Information Technology
Pázmány Péter Catholic University
H-1083, Budapest, Práter utca 50/a,
e-mail: tamas_zeffer@yahoo.com
Department of Automation,
Széchenyi István University,
H-9026, Győr, Egyetem tér 1.,
e-mail: hidvegi@sze.hu

Abstract In this paper, the architecture of a reconfigurable neural network implemented with mixed-signal VLSI hardware is presented. The proposed architecture provides a test substrate for mixed-signal hardware neuro-computing in the area of visual processing. This design consists of a reconfigurable Artificial Neural Network (ANN) utilized as a more general-purpose visual processor with a 352 x 288 photodetecting sensor array, analog RAM, and off chip sensory signal processing capability. The CMOS sensor array possesses high performance readout circuitry and embedded data storage capability. For further compactness, a new template memory structure is built that requires about quarter of silicon size compared to previous designs. The VLSI chip comprises most of the modules of a sensory system and keeps tight relationship between the sensors and the processing neural network.

1. Introduction

In case of neural networks, quality-processing possesses reconfigurability. Reconfigurability of a neural network describes the ability to transform the network topology from a used one to a better one. This feature in mind can provide several advantages compared to fixed topologies where the interconnections of neurons are commonly hardwired in a VLSI chips. Firstly, the network can turn to a more general problem solving network since new problems often call for different topologies. Secondly, some problems require decision regions to be set properly by the rearrangement of neurons in additional layers in order to converge to the desired solution. Further importance of reconfigurability is network resolution that depends on the number of synapses connected to the summation node in order to alter computation accuracy.

Considering all of the mentioned reasons of reconfigurability, we designed a mixed-signal VLSI architecture for neural networks with programmable topology. With our

work, we aim to provide a test substrate for mixed-signal hardware neuro-computing in the area of visual processing.

Besides the architectural ability of exploiting different network topologies in visual processing, the sensor of this chip needs to provide quality image acquisition. Quality-imaging demands high performance pixel and readout circuits for technology scaling down below 0.35 micrometer. For 0.25 and 0.18 micrometer technologies, leakage current and lower quantum efficiency are dominant factors and can massively degrade the photosignal [1]. For quality-imaging, we utilize the shared zero-bias buffer that has proven ability keeping dark current of the photodiodes and the leakage current of the switching transistors at low values [2].

The proposed chip is effective in system integration where the chip contains most of the modules, such as image acquisition, analog/digital processing, and memory. This quality is obtained by the implementation of a more compact template/weight memory structure. For further compactness and integration, we also introduce a new sensor circuit where analog RAM is embedded in the sensor array.

In the second section, the overall architecture of the vision chip is presented. The third section details the basic module of the reconfigurable network: the three weighted input neuron. The fourth section describes the sensor array with the embedded analog RAM. The fifth section is focusing on the new template memory, its structure and operation. We conclude our design with simulated chip performance regarding to speed and the sizes of the sensor array, the arithmetic, and the template memory.

2. The Overall Architecture of Vision Chip

The overall architecture of the proposed vision chip is shown in Figure 1. The arithmetic unit of the chip can receive data from two different sources, the on chip optical sensor and an external sensor (also can be other then optical). Analog to digital converter (A/D) is placed between the sensed analog signal and the input of the Programable Logic Unit (PLU). The PLU receives a programmable interconnect pattern from the Control RAM module in order to ensure rearrangement of data between some already stored input values and the Configurable ANN. The stored input values are in registers that are part of the PLU. This unit is also able to reconfigure neural connections depending on the network topology chosen. The Reconfigurable ANN that performs the arithmetic of the network receives both the weights and the input, or the feedback signals. This unit consists of ‘three weighted input neuron’ modules, the basic modules of the configurable network, also described in the third section. With the aid of this module, the support of different topologies [3]-[8] are established The Reconfigurable ANN unit can include multiple hardware cores working in parallel [6] or other architecture based neural networks [7]-[8]. Feedback or multilayered topologies can be established via the Sensor/ARAM or directly from the Switch Matrix. The Optical Sensor Array is able to detect visual information via an on board 352 x 288 photodiode array or operate as an analog random access memory (ARAM) storing data in a 176 x 144 array. The ARAM is also able to work as 176 x 144 digital bit RAM (equivalent to 3.2 Kbytes capacity).

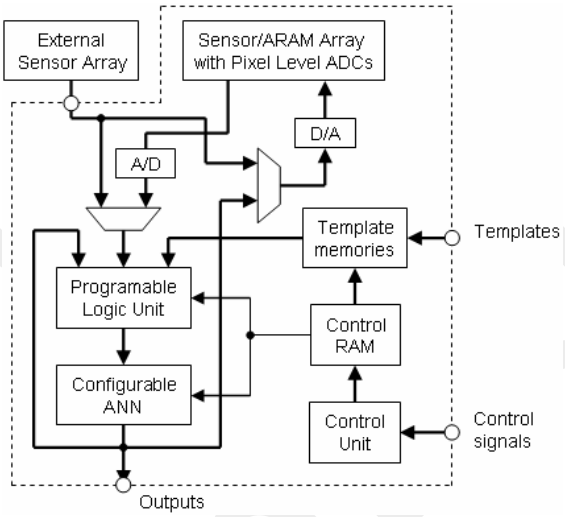


Figure 1. The overall architecture of the vision chip consists of the Configurable Artificial Neural Network, the Programmable Logic Unit, the redesigned template memories, their control circuitry, and the Optical Sensor/ARAM Array

3. The Basic Module of the Reconfigurable Neural Network

The reconfigurable neural network is built of multipliers, adders, and other simple digital circuits that are able to take two main configuration states. The first state forms m pieces of CNN (CASTLE) architecture working in parallel and the second state forms a Multilayered Feedforward Neural Network (MFNN) or a Hopfield network as we described in [7] and [8]. The neural network is built out of modules where one basic module is a three weighted input neuron shown in Figure 2.

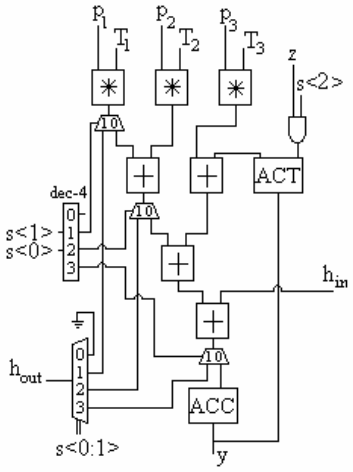


Figure 2. The three weighted input neuron, the basic module of the configurable neural network.

The module is similar to the neurons presented in previous publications, but it also includes an additional adder, a decoder, a multiplexer, and three demultiplexers. The modules can be cascaded through ports h_{in} and h_{out}. Depending on the control signal s<0:1>, the module can multiplex (p_1T_1) or (p_1T_1+p_2T_2) or (p_1T_1+ p_2T_2+ p_3T_3) to the left neighboring module or can receive the same values of the right neighboring module. Control signal s<2>=1 defines z_m in CNN or U_n in Hopfield, s<2>=0 disables this port by adding zero to the sum. The m letter in the subscript is the number of modules and n is the number of neurons in the MFNN or in the Hopfield network.

4. The Sensor/Aram Array

Because of scaling down of CMOS technology, photodiodes have much lower quantum efficiency (QE) and transistors possess dominant leakage currents [9] that overall, massively degenerate the photosignal in simpler but high resolution source-follower-per-detector (SFD) sensor arrays [10]. Quality image acquisition, however; requires high-performance circuitry that keeps the detectors' current stable and at low bias in order to minimize dark current and detector noise [11]-[12]. Our choice of high performance readout circuit is the shared zero-biased-buffer circuit with four n-diffusion/p-substrate photodiode as it is suggested [13].

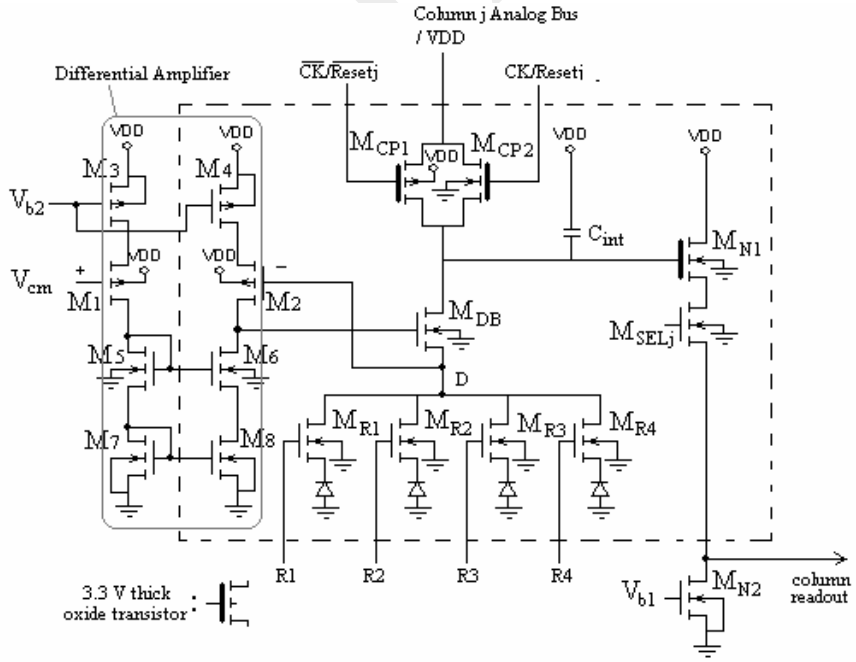


Figure 3. Unit Cell comprises differential amplifier, complementary switch (M_{CP1} and M_{CP2}), and four photodiodes with switching transistors (M_{R1}-M_{R4})

We applied two modifications to the circuit as shown in Figure 3. Firstly, we store analog signals on the integration capacitor C_{int} by applying analog signal at the input of the complementary switch consisting M_{CP1} and M_{CP2}. The complementary switch either charges the capacitor to the voltage of analog signal (the array is utilized as Analog

RAM) or resets the photodiode to voltage VDD. Secondly, we utilize only one transistor, M_{DB} between the switching transistors of the photodiodes and the integrating capacitor. This solution provides lower input impedance to the detector, $R_{in} = g_m /(1+A)$ where g_m is the transconductance of M_{DB} and A is the voltage gain of the differential amplifier. The gain of the operational amplifier is designed to be 80 V/V operated at common voltage of 150 mV (V_{cm}). Lower input resistance provides higher injection efficiency, increases bandwidth and decreases input referred noise [14]-[16]. The buffer with one transistor gives a more stable (still close to zero) detector bias voltage for a wider input range of photocurrent compared to the original circuitry, with the diode connected transistor. The reverse detector voltages are compared in Figure 4. Because of leakage currents the design includes 3.3 voltage thick oxide transistors for the complementary switch (M_{CP1} and M_{CP2}) and for the follower (M_{N1}).

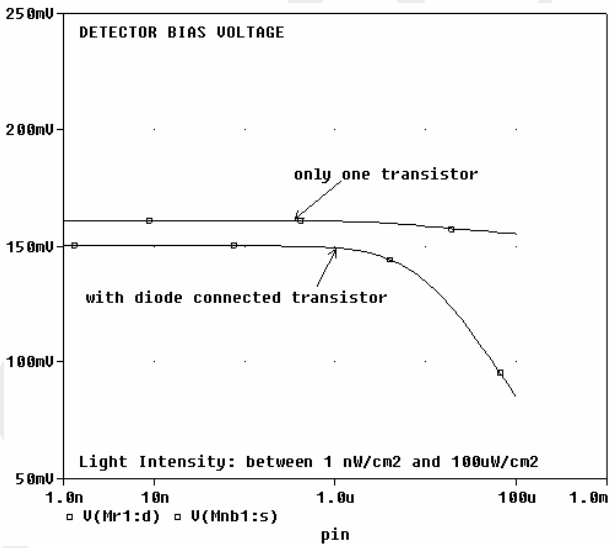


Figure 4. The Comparison of reverse detector voltages of the readout circuit with only one and with diode connected transistor

5. Template Memory

The Figure 5. shows the traditional template memory [4], [5]. The disadvantages of this solution are the big area on silicon (or on FPGA).

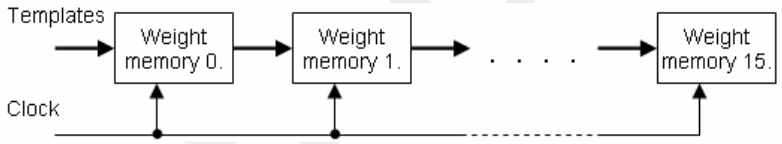


Figure 5. The traditional template memory

The reconfigurable template memory is shown in Figure 6. The template memory consists of thirty registers and multiplexers. The accuracy of the registers is twenty-four

(3 x 8) bits. The templates' bus accuracy is eight bits. The registers are loaded via this bus.

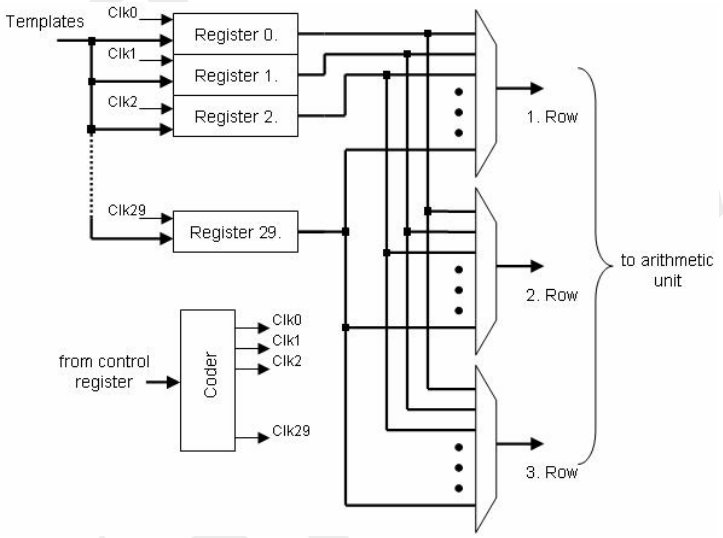


Figure 6. The reconfigurable template memory

The templates are treated as three by eight registers. The registers are used for three by three templates. The "Coder" creates thirty different clock signals to control the thirty template registers as shown in Figure 7.

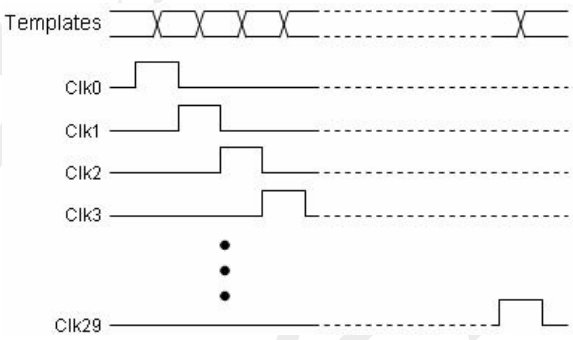


Figure 7. The different clock signals and the Templates bus

This CNN operates with 3 x 3 templates. One register contains one row of templates. So, the template memory is able to store thirty templates. From these rows the templates are assembled. The CASTLE architecture works with four registers to store a template. Our solution utilizes only one register. Figure 8 shows that the new template memory is more compact.

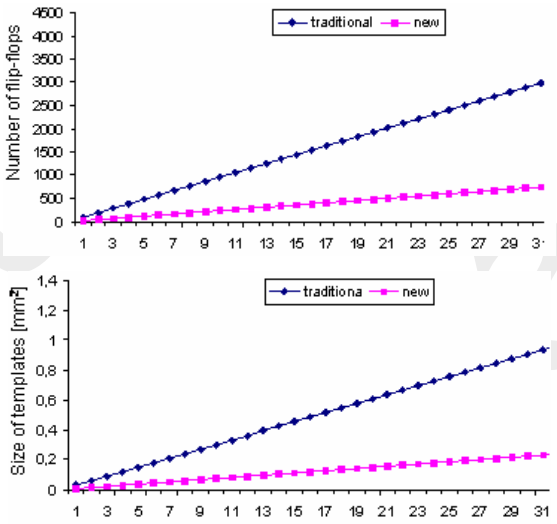


Figure 8. On the upper graph, the number of flip-flops, on the under graph, the realized silicon size of the traditional and the new solutions vs. the number of templates are depicted.

This circuit-solution realizes thirty memory-blocks but only thirty registers and multiplexers are used in the template memory.

6. Conclusion

The design of a reconfigurable neural network built, using CMOS VLSI has been presented. The utilization of new blocks, namely the reconfigurable Artificial Neural Network and the reconfigurable template memory, has made the neural network in terms of computation more general and efficient. In this paper, we did not argue the kind and the structure of the network topology for different visual tasks but rather provided an architectural solution with which a test bed for mixed-signal hardware neuro-computing in the area of visual processing can be established. The system is more compact compared previous solution [5] because the sensor array consists of analogue RAM and the template memory includes quarter amount of registers in comparison with the original CASTLE architecture [4], [5]. The new structure of template memory is able to store up to thirty templates. The area and speed performance of a pixel and some of the important digital modules are summed in Table I.

Table 1. The simulated results

	Pixel cell	*Register (8 bit)*	*Multiplexer/neuron*	**One neuron arithmetic**
Size on silicon [μm^2]	13	2520	705	340428
Delay [ns]	330	0.5	0.5	0.5

References

[1] Buchanan, D., Lo., S. H.: *Growth, characterization and the limits of ultrathin SiO_2-based dielectrics for future CMOS applications,* The physics and chemistry of SiO_2 and the Si-SiO_2 interface e-3, Electrochemical Society Meeting Proceeding 90(1), (1996), pp. 3-14. [9]

[2] Hackbarth, E., Tang, D. D.: *Inherent and Stress-Induced Leakage in Heavily Doped Silicon Junctions,* IEEE Transactions on Electronic Devices 35(12), (1988), pp. 2018-2118. [12]

[3] Hewitt, M. J., Vampola, J. K., Black, S. H., Nielsen, C. J.: *Infrared readout electronics: A historical perspective,* in Proc. SPIE Infrared Readout Electronics II, vol. 2226, (1994), pp. 108-120. [16]

[4] Hidvégi, T., Keresztes, P., Szolgay, P.: *Enhanced Modified Analized Emulated Digital CNN-UM (CASTLE) Arithmetic Cores,* Journal of Circuits, Systems, and Computers, Special Issue on CNN Technology and Visual Microprocessors [5]

[5] Hsish, C. C., Wu, C. Y., Jih, F. W., Sun, T. P.: *Focal-Plane arrays and CMOS readout techniques of infrared imaging systems,* IEEE Trans. Circuits Syst. Video Technol. Vol. 7., (1997), pp. 594-605. [15]

[6] Paasio, A., Kananen, A., Porra, V.: *A 176*144 processor binary I/O CNN-UM chip design* Proc. of ECCTD '99, Stresa, (1999), pp. 82-86.[6]

[7] Roska, T., Chua, L O.: *The CNN Universal Machine: an analogic array computer,* IEEE Transactions on Circuits and Systems-II Vol. 40., (1993), pp. 163-173.

[8] Tian, H., Liu, X., Lim, S., Kleinfelder, S., El Gamal, A.: *Active pixel sensors fabricated in a standard 0.18 um CMOS technology,* technical report, Information System Laboratory, Stanford University. [10]

[9] Vampola, J. L.: *Readout electronics for infrared sensors,* in The infrared and Electro-optical Systems Handbook. Ballingham, VA: SPIE, Vol. 3., Ch. 5, (1993), pp. 286-324. [14]

[10] Vincent, G., Chantre, A., Bios, D.: *Electric Field Effect on the Thermal Emission of Traps in Semiconductor Junctions,* Journal of Applied Physics 50, (1979), pp. 5484-5487. [11]

[11] Wong, H.: *Technology and device scaling considerations for CMOS imagers,* IEEE Transactions on Electron Devices 43 (12), (1996), pp. 2131-2141. [1]

[12] Wu. C. Y., Hsieh, C. C.: *New Design Techniques for complementary metal-oxide semiconductor current readout integration circuit for infrared detector arrays,* Opt. Eng., vol. 34., no. 1., (1995), pp. 160-168.[2]

[13] Yu-Chuan Shih, Chung-Yu Wu: *Optimal Design of CMOS pseudoactive pixel sensor (PAPS) structure for low-dark-current and large-array-size imager applications,* IEEE Sensors Journal, Vol. 5., No 5., (2005). [13]

[14] Zarándy, A., Keresztes, P., Roska, T., Szolgay, P.: *An emulated digital architecture implementing the CNN Universal Machine,* Proc. of the fifth IEEE Int. Workshop on Cellural Neural Networks and their Applications, (1998), pp. 249-252. [4]

[15] Zeffer, T., Hidvégi, T.: *The Configurable Digital Cellular Neural – Hopfield Network,* 10th IEEE International Conference on Intelligent Engineering Systems, (2006). [8]

[16] Zeffer, T., Hidvégi, T.: *The Configurable Digital Neural Network with Emulated Digital Cellular Neural Network Cores,* IEEE International Conference on Mechatronics, (2006) [7]

Efficient Algorithms for Determining the Linear and Convex Separability of Point Sets

Gábor Takács

Széchenyi István University, Győr, Hungary

Abstract: Determining the linear and the convex separability of the classes are interesting questions in the data exploration phase of building intelligent classifier systems. In this paper I propose novel algorithms for finding the answer to these questions efficiently. I demonstrate by experiments on real-world datasets that the proposed algorithms compare favorably in running time with other known methods.

Keywords: machine learning, linear separability, convex separability, linear programming

1. Introduction

One of the most important goals of *machine learning* is to provide tools for building intelligent systems that extract knowledge from an available set of data. The heart of such a system is often a binary classifier (e.g. in spam filtering or computer-aided breast cancer recognition). The task of the classifier is to predict the value of a high-level binary attribute (e.g. spam / non-spam, sick / healty) from the values of some observable, low-level features (e.g. the list of words appearing in an e-mail, pixel intenities of an X-ray image).

If we have d low-level features, and all of them are encoded as real numbers, then each observation can be represented as a point in $\mathbb{R}^d$. Observations belonging to the same class form a set of points.

An important initial step of building a classifier system is exploring the data. This can help in choosing the appropriate classification algorithm for the problem. One typical question in this phase is whether the classes are *linearly separable* from each other in the training set. Another interesting question is whether the classes are *convexly separable*.

In this paper I will propose novel algorithms for answering these questions efficiently. It will be demonstrated by experiments on real-world datasets that the proposed algorithms are better than other known methods in terms of running time.

2. Problem definition

Assume that $\mathcal{P}$ and $\mathcal{Q}$ are two finite point sets in $\mathbb{R}^d$. One may ask different questions about separating $\mathcal{P}$ and $\mathcal{Q}$:

A) Is there a half-space $\mathcal{S} \subset \mathbb{R}^d$ such that $\mathcal{P} \subset \mathcal{S}$ and $\mathcal{Q} \subset \bar{\mathcal{S}}$?[1]

B) Is there a convex polytope (intersection of half-spaces) $\mathcal{S} \subset \mathbb{R}^d$ such that $\mathcal{P} \subset \mathcal{S}$ and $\mathcal{Q} \subset \bar{\mathcal{S}}$?

C) For fixed K, is there a convex K-polytope (intersection of K half-spaces) $\mathcal{S} \subset \mathbb{R}^d$ such that $\mathcal{P} \subset \mathcal{S}$ and $\mathcal{Q} \subset \bar{\mathcal{S}}$?

This paper will deal with questions A and B. They are known as the problem of *linear separability* (A) and the problem of *convex separability* (B). We will see that both can be decided in polynomial time. In contrast, answering question C is NP hard[2]. The only easy case is $K = 1$, where the task is to determine linear separability. It is interesting that even the case $K = 2$ called *wedge separation* is NP hard [1, 12].

Now let us proceed by introducing some notation and definitions, in order to investigate the problem in more detail. Let $\mathcal{T}$ be a subset of $\mathbb{R}^d$. The *convex hull* of $\mathcal{T}$ denoted by $\text{conv}(\mathcal{T})$ is the minimal convex subset of $\mathbb{R}^d$ that contains $\mathcal{T}$. If $\mathcal{T}$ is finite, then $\text{conv}(\mathcal{T})$ is a convex polytope. A convex polytope in $\mathbb{R}^d$ can be given either by its vertices or its $(d-1)$-dimensional facets. The former is called *vertex representation*, the latter is called *half-space representation*. Typically, the half-space representation cannot be computed in high dimensions, because the number of $(d-1)$-dimensional facets is intractably large.

Let $\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_m\} \subset \mathbb{R}^d$ and $\mathcal{Q} = \{\mathbf{q}_1, \ldots, \mathbf{q}_n\} \subset \mathbb{R}^d$ be two finite point sets.

Definition 1. $\mathcal{P}$ and $\mathcal{Q}$ are *linearly separable* if and only if $\text{conv}(\mathcal{P}) \cap \text{conv}(\mathcal{Q}) = \emptyset$.

Definition 2. $\mathcal{P}$ and $\mathcal{Q}$ are *convexly separable* if and only if $\mathcal{P} \cap \text{conv}(\mathcal{Q}) = \emptyset$ or $\mathcal{Q} \cap \text{conv}(\mathcal{P}) = \emptyset$. If $\mathcal{P} \cap \text{conv}(\mathcal{Q})$ and $\mathcal{Q} \cap \text{conv}(\mathcal{P})$ are both empty, then $\mathcal{P}$ and $\mathcal{Q}$ are called *mutually convexly separable*. If $\mathcal{Q} \cap \text{conv}(\mathcal{P})$ is empty, but $\mathcal{P} \cap \text{conv}(\mathcal{Q})$ is not, then $\mathcal{P}$ is called the *inner set* and $\mathcal{Q}$ the *outer set*.

Figure 1 shows examples for different types of separability. Note that linear separability implies mutual convex separability, but the reverse is not true.

[1] $\bar{\mathcal{S}}$ means $\mathbb{R}^d \setminus \mathcal{S}$, the complement of $\mathcal{S}$.

[2] It is important to note that the dimension d is not fixed.

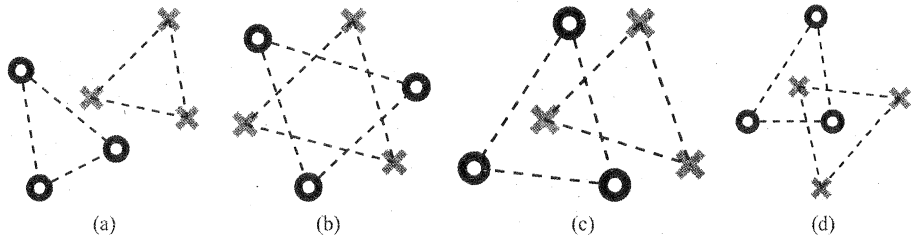


Figure 1: Examples for linearly separable (a), mutually convexly separable (b), convexly separable (c), and convexly nonseparable (d) point sets.

3. Algorithms for linear separability

The question of linear separability can be formulated as a linear programming problem:

$$\begin{aligned} &\text{variables: } \mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R} \\ &\text{minimize: } 1 \\ &\text{subject to: } \mathbf{w}^T\mathbf{p}_i + b \geq +\varepsilon, \quad i = 1,\ldots,m \\ &\qquad\qquad \mathbf{w}^T\mathbf{q}_j + b \leq -\varepsilon, \quad j = 1,\ldots,n \end{aligned} \tag{1}$$

where ε is an arbitrary positive constant. The constraints express that the elements of $\mathcal{P}$ and $\mathcal{Q}$ have to be on the opposite sides of the hyperplane $\mathbf{w}^T\mathbf{x} + b = 0$. $\mathcal{P}$ and $\mathcal{Q}$ are linearly separable if and only if the problem is feasible. This basic and straightforward method will be referred as LSEP_1.

Maybe it is a bit unusual in LSEP_1 that the function to minimize is constant. By introducing slack variables we can obtain a formulation that has a non-constant objective function, and that always has a feasible and bounded solution:

$$\begin{aligned} &\text{variables: } \mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R}, \mathbf{s} \in \mathbb{R}^m, \mathbf{t} \in \mathbb{R}^n \\ &\text{minimize: } \sum_{i=1}^{m} s_i + \sum_{j=1}^{n} t_j \\ &\text{subject to: } \mathbf{w}^T\mathbf{p}_i + b \geq +\varepsilon - s_i, \quad s_i \geq 0, \quad i = 1,\ldots,m \\ &\qquad\qquad \mathbf{w}^T\mathbf{q}_j + b \leq -\varepsilon + t_j, \quad t_j \geq 0, \quad j = 1,\ldots,n \end{aligned} \tag{2}$$

$\mathcal{P}$ and $\mathcal{Q}$ are linearly separable if and only if the solution is $\mathbf{s} = \mathbf{0}$,$\mathbf{t} = \mathbf{0}$. Linear programming can be solved in polynomial time e.g. by using Karmarkar's algorithm [9], therefore linear separability can be decided in polynomial time.

The dual of the LSEP_1 formulation (referred as LSEP_1^*) is the following:

$$\begin{aligned} &\text{variables: } \boldsymbol{\alpha} \in \mathbb{R}^m, \boldsymbol{\beta} \in \mathbb{R}^n \\ &\text{maximize: } \varepsilon \left(\sum_{i=1}^{m} \alpha_i + \sum_{j=1}^{n} \beta_j \right) \\ &\text{subject to: } \sum_{i=1}^{m} \alpha_i \mathbf{p}_i = \sum_{j=1}^{n} \beta_j \mathbf{q}_j \\ &\qquad \sum_{i=1}^{m} \alpha_i = \sum_{j=1}^{n} \beta_j, \quad \boldsymbol{\alpha}, \geq \mathbf{0}, \quad \boldsymbol{\beta} \geq \mathbf{0} \end{aligned} \tag{3}$$

Note that the problem is always feasible. If $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are not zero vectors, then the constraints are expressing that the $\mathrm{conv}(\mathcal{P})$ and $\mathrm{conv}(\mathcal{Q})$ have a common point. $\mathcal{P}$ and $\mathcal{Q}$ are linearly separable if and only if the solution is $\boldsymbol{\alpha} = \mathbf{0}, \boldsymbol{\beta} = \mathbf{0}$. If $\mathcal{P}$ and $\mathcal{Q}$ are not linearly separable, then the solution is unbounded.

It is natural to introduce a slightly modified version of LSEP_1^* (referred as LSEP_1^+):

$$\begin{aligned} &\text{variables: } \boldsymbol{\alpha} \in \mathbb{R}^m, \boldsymbol{\beta} \in \mathbb{R}^n \\ &\text{maximize: } \varepsilon \left(\sum_{i=1}^{m} \alpha_i + \sum_{j=1}^{n} \beta_j \right) \\ &\text{subject to: } \sum_{i=1}^{m} \alpha_i \mathbf{p}_i = \sum_{j=1}^{n} \beta_j \mathbf{q}_j \\ &\qquad \sum_{i=1}^{m} \alpha_i = \sum_{j=1}^{n} \beta_j = 1, \quad \boldsymbol{\alpha}, \geq \mathbf{0}, \quad \boldsymbol{\beta} \geq \mathbf{0} \end{aligned} \tag{4}$$

The difference from LSEP_1^* is that now the components must sum to 1 in $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. $\mathcal{P}$ and $\mathcal{Q}$ are linearly separable if and only if the problem is infeasible. If $\mathcal{P}$ and $\mathcal{Q}$ are not linearly separable, then the problem has a feasible solution.

An interesting modification of LSEP_1 (referred as LSEP_2) tries to find a separating hyper-

plane with small norm:

$$\begin{aligned} &\text{variables: } \mathbf{w},\mathbf{v} \in \mathbb{R}^d,\ b \in \mathbb{R} \\ &\text{minimize: } \sum_{k=1}^{d}(w_k + v_k) \\ &\text{subject to: } (\mathbf{w}-\mathbf{v})^T\mathbf{p}_i + b \geq +\varepsilon, \quad i = 1,\ldots,m \\ &\qquad\qquad (\mathbf{w}-\mathbf{v})^T\mathbf{q}_j + b \leq -\varepsilon, \quad j = 1,\ldots,n \\ &\qquad\qquad \mathbf{w} \geq \mathbf{0}, \quad \mathbf{v} \geq \mathbf{0} \end{aligned} \tag{5}$$

$\mathcal{P}$ and $\mathcal{Q}$ are linearly separable if and only if the problem is feasible. The price of penalizing the L1 norm of $\mathbf{w}$ and $\mathbf{v}$ is that LSEP_2 has (nearly) twice as many variables as LSEP_1. We will see that this extra computational cost can pay off in certain cases.

The dual of LSEP_2 (referred as LSEP_2^*) is the following:

$$\begin{aligned} &\text{variables: } \boldsymbol{\alpha} \in \mathbb{R}^m,\ \boldsymbol{\beta} \in \mathbb{R}^n \\ &\text{maximize: } \varepsilon\left(\sum_{i=1}^{m}\alpha_i + \sum_{j=1}^{n}\beta_j\right) \\ &\text{subject to: } -\mathbf{1} \leq \sum_{i=1}^{m}\alpha_i\mathbf{p}_i - \sum_{j=1}^{n}\beta_j\mathbf{q}_j \leq \mathbf{1} \\ &\qquad\qquad \sum_{i=1}^{m}\alpha_i = \sum_{j=1}^{n}\beta_j, \quad \boldsymbol{\alpha}, \geq \mathbf{0}, \quad \boldsymbol{\beta} \geq \mathbf{0} \end{aligned} \tag{6}$$

where $\mathbf{1}$ denotes the all-one vector. $\mathcal{P}$ and $\mathcal{Q}$ are linearly separable if and only if the solution is $\boldsymbol{\alpha} = \mathbf{0}, \boldsymbol{\beta} = \mathbf{0}$.

Finally, a quadratic programming based formulation (referred as LSEP_S) is the following:

$$\begin{aligned} &\text{variables: } \mathbf{w} \in \mathbb{R}^d,\ b \in \mathbb{R},\ \mathbf{s} \in \mathbf{R}^m,\ \mathbf{t} \in \mathbb{R}^n \\ &\text{minimize: } \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\left(\sum_{i=1}^{m}s_i + \sum_{j=1}^{n}t_j\right) \\ &\text{subject to: } \mathbf{w}^T\mathbf{p}_i + b \geq +1 - s_i, \quad s_i \geq 0, \quad i = 1,\ldots,m \\ &\qquad\qquad \mathbf{w}^T\mathbf{q}_j + b \leq -1 + t_j, \quad t_j \geq 0, \quad j = 1,\ldots,n \end{aligned} \tag{7}$$

Note that this is equivalent with linear support vector machine (SVM) [4] training. $\mathcal{P}$ and $\mathcal{Q}$ are linearly separable if and only if there exist a $C > 0$ for which the solution

has the following property: $s_1,\ldots,s_m,t_1,\ldots,t_n < 1$. In practice it is not possible to check this property for all C values, just for a reasonably large one. Therefore, we cannot completely rely on the answer, if LSEP_S says no. Obviously, introducing the term $\frac{1}{2}\mathbf{w}^T\mathbf{w}$ into the objective function makes the optimization problem harder. The rationale behind this formulation is that for linear SVM training there exist efficient specialized algorithms (e.g. sequential minimal optimization [13]), and fine-tuned software containing them (e.g. svm-light [7], libsvm [5]).

3.1. Proposed new methods

The algorithms presented so far try to solve the full problem in one step. Here I propose a novel approach (referred as LSEPX) that is incremental:

1. Let $\mathcal{P}_1,\ldots,\mathcal{P}_L$ and $\mathcal{Q}_1,\ldots,\mathcal{Q}_L$ be two a systems of sets such that $\mathcal{P}_1 \subset \cdots \subset \mathcal{P}_L = \mathcal{P}$ and $\mathcal{Q}_1 \subset \cdots \subset \mathcal{Q}_L = \mathcal{Q}$.
2. For $k = 1,\ldots,L$:
   - Check whether $\mathcal{P}_k$ and $\mathcal{Q}_k$ are separable using LSEP_1, or LSEP_2. The result of this step is a yes or no answer and a separating hyperplane $\mathbf{w}_k^T\mathbf{x} + b_k = 0$ if the answer is yes.
   - If the answer is no, then $\mathcal{P}$ and $\mathcal{Q}$ are not linearly separable.
   - If the hyperplane $\mathbf{w}_k^T\mathbf{x} + b_k = 0$ separates $\mathcal{P}$ and $\mathcal{Q}$, then $\mathcal{P}$ and $\mathcal{Q}$ are linearly separable.

$\mathcal{P}_k$ and $\mathcal{Q}_k$ can be called the active sets in the k-th iteration. The last iteration is equivalent with solving the full problem. The advantage of the approach is that there is a chance of getting the answer before the last iteration. However, there is no guarantee for that. A reasonable heuristic for defining the active sets is the following:

1. $\mathcal{P}_1$, $\mathcal{Q}_1 \leftarrow$ random $\min\{d,|\mathcal{P}|\}$ and $\min\{d,|\mathcal{Q}|\}$ element subsets of $\mathcal{P}$ and $\mathcal{Q}$.
2. At the k-th iteration:
   - For each $\mathbf{x} \in \mathcal{P} \cup \mathcal{Q}$ calculate $\delta_k(\mathbf{x}) = (\mathbf{w}_k^T\mathbf{x} + b)(-1)^{I\{\mathbf{x}\in\mathcal{Q}\}}$.
   - Denote the set of γ_k points with smallest δ_k values by $\mathcal{U}_k$.
   - $\mathcal{P}_{k+1} \leftarrow \mathcal{P}_k \cup (\mathcal{U}_k \cap \mathcal{P})$, $\quad \mathcal{Q}_{k+1} \leftarrow \mathcal{Q}_k \cup (\mathcal{U}_k \cap \mathcal{Q})$.

Thus, in each iteration the points with largest "errors" are added to the active sets. Some possible choices for γ_k are $\gamma_k \equiv 1$, $\gamma_k \equiv d$, or $\gamma_k = 2^k d$.

A possible disadvantage of LSEPX is that points are never removed from the active sets. As a consequence, the active sets may contain redundant elements, which can increase running time. On the other hand, if we allow removals from the active sets without restrictions, then there is no guarantee for stopping. A simple solution to the dilemma is to allow removing points only once.

The modified version of LSEPX (referred as LSEPY) defines the active sets as follows:

1. $\mathcal{P}_1, \mathcal{Q}_1 \leftarrow$ random $\min\{d, |\mathcal{P}|\}$ and $\min\{d, |\mathcal{Q}|\}$ element subsets of $\mathcal{P}$ and $\mathcal{Q}$.
2. At the k-th iteration:
   - For each $\mathbf{x} \in \mathcal{P} \cup \mathcal{Q}$ calculate $\delta_k(\mathbf{x}) = (\mathbf{w}_k^T \mathbf{x} + b)(-1)^{I\{\mathbf{x} \in \mathcal{Q}\}}$.
   - Denote the set of γ_k points with smallest δ_k values by $\mathcal{U}_k$.
   - Denote the set of points with δ_k value greater than $\varepsilon_2 > 0$ by $\mathcal{V}_k$.
   - $\mathcal{V}_k \leftarrow \mathcal{V}_k \setminus (\cup_{l=1}^{k-1} \mathcal{V}_l)$, in order to avoid multiple removals.
   - Denote the element of $\mathcal{P}$ and $\mathcal{Q}$ with minimal δ_k value by $\mathbf{p}'$ and $\mathbf{q}'$. Remove $\mathbf{p}'$ and $\mathbf{q}'$ from $\mathcal{V}_k$, in order to keep at least 1 point from $\mathcal{P}$ and $\mathcal{Q}$.
   - $\mathcal{P}_{k+1} \leftarrow \mathcal{P}_k \cup (\mathcal{U}_k \cap \mathcal{P}) \setminus (\mathcal{V}_k \cap \mathcal{P})$, $\quad \mathcal{Q}_{k+1} \leftarrow \mathcal{Q}_k \cup (\mathcal{U}_k \cap \mathcal{Q}) \setminus (\mathcal{V}_k \cap \mathcal{Q})$.

It is also possible to introduce an incremental method based on the dual based formulation LSEP_1^+. The outline of the algorithm (referred as LSEPZ) is the following:

1. Create reduced versions of $\mathcal{P}$ and $\mathcal{Q}$ by keeping only γ randomly selected coordinates (features). Denote the result by $\mathcal{P}^1$ and $\mathcal{Q}^1$.
2. For $k = 1, \ldots n$:
   - Check whether $\mathcal{P}^k$ and $\mathcal{Q}^k$ are separable using LSEP_1^+. The result of this step is a yes or no answer and an α and a β vector, if the answer is no.
   - If the answer is yes, then $\mathcal{P}$ and $\mathcal{Q}$ are linearly separable.
   - If the answer is no, then:
     * If $\mathcal{P}_k = \mathcal{P}$ and $\mathcal{Q}_k = \mathcal{Q}$, then $\mathcal{P}$ and $\mathcal{Q}$ are not linearly separable.
     * Calculate $\mathbf{s} = \sum_{i=1}^{m} \alpha_i \mathbf{p}_i - \sum_{j=1}^{n} \beta_j \mathbf{q}_j$, where $\mathbf{p}_i$-s and $\mathbf{q}_j$-s are from the original $\mathcal{P}$ and $\mathcal{Q}$ sets.
     * Denote the coordinates with largest $|s_k|$ values by $\mathcal{U}^k$.
     * Define $\mathcal{P}^{k+1}$ and $\mathcal{Q}^{k+1}$ as the extension of $\mathcal{P}^k$ and $\mathcal{Q}^k$ with the coordinates in $\mathcal{U}^k$.

It is interesting to observe that the dual based LSEPZ is not perfectly "symmetric" to the previous two primal based approaches. While LSEPX and LSEPY are able to achieve a speedup both in the separable and the nonseparable case, LSEPZ is capable of that only in the nonseparable case. Note that only LSEP_1^+ can be used among the dual based basic methods in LSEPZ, since LSEP_1^* and LSEP_2^* can have unbounded solution in the nonseparable case.

Finally, I would like to mention that it is possible to define hybrid methods (referred as LSEPZX and LSEPZY) based on the previous algorithms:

1. Run LSEPZ and try to reduce the number of coordinates (features).
2. If the answer of LSEPZ is yes, then run LSEPX or LSEPY on the reduced dataset.

4. Algorithms for convex separability

Recall that $\mathcal{P}$ and $\mathcal{Q}$ are called convexly separable, if and only if $\mathcal{P} \cap \text{conv}(\mathcal{Q}) = \emptyset$ or $\mathcal{Q} \cap \text{conv}(\mathcal{P}) = \emptyset$. Without loss of generality assume that we want to decide whether $\mathcal{Q} \cap \text{conv}(\mathcal{P})$ is empty. The other property can be checked exactly the same way.

At first we overview a naive approach with exponential time complexity:

1. Compute a half-space representation of $\text{conv}(\mathcal{P})$. This yields $\mathbf{w}_k$-s and b_k-s ($k = 1,\ldots,r$) such that $\text{conv}(\mathcal{P}) = \{\mathbf{x} \in \mathbb{R}^d : \mathbf{w}_1^T\mathbf{x} \geq b_1, \ldots, \mathbf{w}_r^T\mathbf{x} \geq b_r\}$.
2. $\mathcal{Q} \cap \text{conv}(\mathcal{P}) = \emptyset$ if and only if $\max_{j=1,\ldots,n}\{\min_{k=1,\ldots,r}\{\mathbf{w}_k^T\mathbf{q}_j - b_k\}\} < 0$.

The problem with this algorithm is that the size of half-space representation r typically grows exponentially with d.

The methods presented from now will all have polynomial time complexity. A basic and straightforward approach (referred as CSEP) is to separate each element of $\mathcal{Q}$ from $\mathcal{P}$ individually. The primal based version of the algorithm is the following:

1. $\mathcal{U} \leftarrow \emptyset$, $\mathbf{q} \leftarrow$ a random element of $\mathcal{Q}$.
2. For $k = 1,\ldots,n$:
   - Check whether $\mathcal{P}$ and $\{\mathbf{q}\}$ are linearly separable using LSEP_1, or LSEP_2, LSEPX, LSEPY, LSEPZX or LSEPZY. The result of this step is a yes or no answer and a separating hyperplane $\mathbf{w}_k^T\mathbf{x} + b_k = 0$ if the answer is yes.

- If the answer is no, then $\mathcal{P}$ and $\mathcal{Q}$ are not convexly separable.
- If the answer is yes, then:
  * If $\mathcal{U} = \mathcal{Q}$, then $\mathcal{P}$ and $\mathcal{Q}$ are convexly separable
  * For each $\mathbf{x} \in \mathcal{Q} \setminus \mathcal{U}$ calculate $\delta_k(\mathbf{x}) = -(\mathbf{w}_k^T\mathbf{x} + b_k)$.
  * If the smallest δ_k value is greater than 0, then $\mathcal{P}$ and $\mathcal{Q}$ are convexly separable.
  * Add the point with smallest δ_k value to $\mathcal{U}$.

The dual based version of CSEP is the following:

1. For $k = 1,\ldots,n$:
   - Check whether $\mathcal{P}$ and $\{\mathbf{q}_k\}$ are linearly separable using LSEP_1^*, LSEP_1^+, LSEP_2^*, or LSEPZ. The result of this step is a yes or no answer.
   - If the answer is no, then $\mathcal{P}$ and $\mathcal{Q}$ are not convexly separable.
2. $\mathcal{P}$ and $\mathcal{Q}$ are convexly separable.

Note that the primal based version is able to finish in less than n iterations both in the separable and the nonseparable case. In contrast, the dual based version always runs $|\mathcal{Q}|$ iterations, if $\mathcal{P}$ and $\mathcal{Q}$ are convexly separable.

4.1. Proposed new methods

At first I introduce a fast algorithm (referred as CSEPC) that performs approximate convex separation:

1. $\mathcal{V} \leftarrow \emptyset$. $s_1,\ldots,s_n \leftarrow \infty$.
2. Compute the centroid of $\mathcal{P}$ as $\bar{\mathbf{p}} \leftarrow \frac{1}{m}(\mathbf{p}_1 + \cdots + \mathbf{p}_m)$.
3. Choose $\mathbf{q}_k$ from $\mathcal{Q}$ such that $k = \arg\max_{j=1,\ldots,n}\{s_j\}$.
4. If $s_k \leq 0$, then return $\mathcal{V}$.
5. Compute $\mathbf{w} \leftarrow (\bar{\mathbf{p}} - \mathbf{q}_k)/\|\bar{\mathbf{p}} - \mathbf{q}_k\|$ and $b \leftarrow \min_{i=1,\ldots,m}\{\mathbf{w}^T\mathbf{p}_i\}$.
6. $\mathcal{V} \leftarrow \mathcal{V} \cup \{(\mathbf{w},b)\}$.
7. For all $s_j > 0$: $s_j \leftarrow \min\{s_j, \mathbf{w}^T\mathbf{q}_j - b\}$. $s_k \leftarrow 0$.
8. Go to step 4.

The idea of the algorithm is to define hyperplanes by connecting the elements of $\mathcal{Q}$ with the centroid of $\mathcal{P}$, and translate the hyperplanes to the boundary of $\mathrm{conv}(\mathcal{P})$. Therefore the algorithm can be called the *centroid method*. The centroid method is often able to separate most of the elements of $\mathcal{Q}$ from $\mathrm{conv}(\mathcal{P})$. However, it does not guarantee to find a convex separation even for for convexly separable point sets.

Now I propose an exact algorithm (referred as CSEPX) that uses the centroid method as a preprocessor:

1. Run CSEPC on $\mathcal{P}$,$\mathcal{Q}$. The result of this step is a set of weight–bias pairs $\mathcal{V} = \{(\mathbf{w}_1,b_1),\ldots,$
$(\mathbf{w}_r,b_r)\}$ and a set of not separated points $\mathcal{Q}' = \{\mathbf{q} \in \mathcal{Q} : \min_{k=1,\ldots,r}\{\mathbf{w}_k^T\mathbf{q}+b_k\} \geq 0\}$.
2. Run CSEP on $\mathcal{P}$,$\mathcal{Q}'$. The result of this step is a yes or no answer A indicating whether $\mathcal{Q}' \cap \mathrm{conv}(\mathcal{P})$ is empty, and a set of weight–bias pairs $\mathcal{W}$, if the answer is yes.
3. If A is no, then answer no. If A is yes, then answer yes and return $\mathcal{V} \cup \mathcal{W}$.

In many practical cases CSEPX can achieve a large speedup over the other presented methods. The efficiency of the algorithm will demonstrated by experiments in the Experiments section. I remark that an earlier version of the proposed CSEPC and CSEPX approaches appeared in [14].

5. Experiments

This section describes experiments with the previous algorithms on real-world datasets. The first part will be about determining and analyzing the type of separability in the given datasets. Then the second part will compare the running times of the algorithms.

5.1. Datasets

The datasets involved in the experiments were the following:

- **VOTES**: This dataset is part of the UCI (University of California, Irvine) machine learning repository [3], which is a well known collection of benchmark problems for testing machine learning algorithms. The dataset includes votes for each of the U.S. House of Representatives Congressmen on the $d = 16$ key votes identified by the Congressional Quarterly Almanac (98th Congress, 2nd session, 1984). The

Congressional Quarterly Almanac lists nine different types of votes: voted for, paired for, and announced for (these three are encoded as 1), voted against, paired against, and announced against (these three are encoded as -1), voted present, voted present to avoid conflict of interest, and did not vote or otherwise make a position known (these three are encoded as 0). The dataset contains $M = 2$ classes (democrat or republican) and $n = 435$ examples (267 democrat, 168 republican).

- **WISCONSIN**: This dataset is also part of the UCI machine learning repository. It was originally obtained from the University of Wisconsin Hospitals, Madison, Wisconsin. The task to solve is to determine the benignness or malignancy ($M = 2$) of tumors based on $d = 9$ features extracted from mammograms. After removing examples with missing feature values the number of examples is $n = 683$ (444 benign, 239 malignant).
- **MNIST28**: This dataset is the training set part of the MNIST handwritten digit recognition database [10]. The $d = 784$ features represent pixel intensities of 28×28 sized images. The classes are associated with the digits, therefore the number of classes is $M = 10$. The number of examples is $n = 60000$ (5923 zeros, 6742 ones, 5958 twos, 6131 threes, 5842 fours, 5421 fives, 5918 sixes, 6265 sevens, 5851 eights, 5949 nines). Figure 2 shows some example images from the database. The dataset is sparse, the average frequency of 0 as a feature value is 80.9 %.
- **MNIST14**: This dataset was extracted from the MNIST28 by reducing image size to 14×14 pixels. Intensity values of the 14×14 images were obtained by averaging intensities in the corresponding 2×2 part of the original image. The average frequency 0 as a feature value is 74.4 %
- **MNIST7**: This dataset was extracted from the MNIST28 by reducing image size to 7×7 pixels. Intensity values of the 7×7 images were obtained by averaging intensities in the corresponding 4×4 part of the original image. The average frequency of 0 as a feature value is 62.3 %
- **MNIST4**: This dataset was extracted from the MNIST28 by reducing image size to 4×4 pixels. Intensity values of the 4×4 images were obtained by averaging intensities in the corresponding 7×7 part of the original image. The average frequency of 0 as a feature value is 41.3 %

If we want to refer to the subset of an MNIST dataset containing only classes A and B, then we will use postfix $/AB$ (e.g. MNIST04/49).

5.2. Algorithms

For determining linear separability the following basic algorithms were tested:

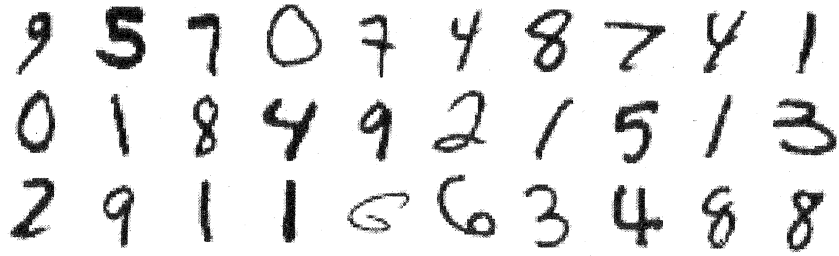


Figure 2: Examples from the MNIST28 database.

- **LSEP$_1$**: The most straightforward, linear programming based method. It tries to find an arbitrary separating hyperplane (see page 3 for the details).
- **LSEP$_1^*$**: The dual formulation of LSEP$_1$ (see page 4).
- **LSEP$_1^+$**: A modified version of LSEP$_1^*$ (see page 4).
- **LSEP$_2$**: An alternative linear programming based method that tries to find a solution with small norm (see page 5).
- **LSEP$_2^*$**: The dual formulation of LSEP$_2$ (see page 5).
- **LSEP$_S$**: The support vector machine based method (see page 5). In the experiments it was used with setting $C = 10^6$.

All of these algorithms try to answer the question whether two point sets are linearly separable or not. The primal based methods (LSEP$_1$, LSEP$_2$) are constructive in the sense that they also provide a separating hyperplane, if the answer is yes. The dual based methods (LSEP$_1^*$, LSEP$_1^+$ and LSEP$_2^*$) cannot do this, but they are able to output an easily verifiable proof, if the answer is no. The parameter ε was always set to 0.001 in the experiments.

The support vector based approach (LSEP$_S$) can only be considered as an approximate method, because C cannot be set to ∞, therefore the answer is not perfectly reliable in the nonseparable case.

For determining linear separability the following enhanced algorithms were tested:

- **LSEPX$_1$**, **LSEPX$_2$**: The first proposed method (see page 6). It tries to solve the problem incrementally. The index indicates which basic method (LSEP$_1$, LSEP$_2$) is used for solving subproblems.

- **LSEPY**$_1$, **LSEPY**$_2$: The modified version of the previous method (see page 7). It tries to further reduce running time by removing points from the active sets.
- **LSEPZ**: A dual based alternative of the previous two approaches (see page 7). It uses LSEP$_1^+$ for solving subproblems. It can be able to reduce the number of features in the linearly separable case.
- **LSEPZX**$_1$, **LSEPZX**$_2$, **LSEPZY**$_1$, **LSEPZY**$_2$: A hybrid method that tries to reduce the number of features with LSEPZ first, and then runs LSEPX$_1$, LSEPX$_2$, LSEPY$_1$, or LSEPY$_2$ on the reduced dataset (see page 8).

The variants of LSEPX, LSEPY, LSEPZX and LSEPZY are constructive methods, while LSEPZ is not. LSEPX and LSEPY was always run with setting $\gamma_k \equiv d$. The parameter ε was always set to 0.001 and ε_2 to 0.002.

For determining convex separability the following algorithms were tested:

- **CSEP**$_{X2}$, **CSEP**$_{Y2}$, **CSEP**$_{ZX2}$, **CSEP**$_{ZY2}$: Straightforward, linear programming based approach that tries to separate each outer point from the inner set individually (see page 8). The indices indicate which primal based method is used for the individual linear separations.
- **CSEP**$_2^*$, **CSEP**$_Z$: The dual based version of the the previous approach (see page 9). The indices indicate which dual based method is used for the individual linear separations.
- **CSEPC**: Proposed method for approximate convex separation (see page 9). It considers the lines connecting the centroid of the inner set with the outer points, and places hyperplanes perpendicular to these lines.
- **CSEPX**$_{X2}$, **CSEPX**$_{Y2}$, **CSEPX**$_{ZX2}$, **CSEPX**$_{ZY2}$, **CSEPX**$_2^*$, **CSEPX**$_Z$: Proposed method for fast and exact convex separation (see page 10). At first it tries to reduce the size of the outer set by applying CSEPC, and then it runs a variant of CSEP on the reduced dataset. The indices indicate which version of CSEP is used in the second phase.

All algorithms were implemented in Python [15], using the numerical (NumPy) [2] and the scientific (SciPy) [8] modules. The internal linear programming solver was the primal simplex method of the GNU Linear Programming Kit (GLPK) [11]. GLPK was accessed from Python via the PyGLPK interface [6]. The internal support vector machine implementation was libsvm [5]. The hardware environment was a notebook PC with Intel Pentium M 2 GHz CPU and 1 Gb memory.

5.3. Types of separability

As we have seen previously, if we have point sets $\mathcal{P}$ and $\mathcal{Q}$, then it is possible to define different types of separability between them:

- S_0: $\mathcal{P}$ and $\mathcal{Q}$ are (convexly) inseparable, if $\mathcal{P} \cap \text{conv}(\mathcal{Q}) \neq \emptyset$ and $\mathcal{Q} \cap \text{conv}(\mathcal{P}) \neq \emptyset$.
- S_1: $\mathcal{P}$ and $\mathcal{Q}$ are convexly separable, if $\mathcal{P} \cap \text{conv}(\mathcal{Q}) = \emptyset$ or $\mathcal{Q} \cap \text{conv}(\mathcal{P}) = \emptyset$.
- S_2: $\mathcal{P}$ and $\mathcal{Q}$ are mutually convexly separable, if $\mathcal{P} \cap \text{conv}(\mathcal{Q}) = \emptyset$ and $\mathcal{Q} \cap \text{conv}(\mathcal{P}) = \emptyset$.
- S_3: $\mathcal{P}$ and $\mathcal{Q}$ are linearly separable, if $\text{conv}(\mathcal{P}) \cap \text{conv}(\mathcal{P}) = \emptyset$.

Note that S_3 implies S_2, and S_2 implies S_1, therefore S_3 can be considered as the strongest and S_1 as the weakest type of separability. Also observe that S_1 is the opposite of S_0.

In the following experiments the type of separability is determined for each pair of classes in the given datasets. At first, linear separability was checked with the LSEPZ method. If the point sets were not linearly separable, then also convex separability was checked with the CSEPX_{X2} method.

The type of separability in the VOTES and the WISCONSIN dataset turned out to be S_1. The results for the MNIST28 dataset can be seen in Table 1, for the MNIST 14 dataset in Table 2, for the MNIST7 dataset in Table 3, and for the MNIST4 dataset in Table 4.

In the case of the MNIST28 dataset 38 of the subproblems are S_3- and 7 are S_2-type. This means that the majority of the 2-class subproblems are linearly separable, therefore it would be reasonable to use simple models, if we wanted to build classifiers.

If we reduce the size of the images, then it is natural to expect that the subproblems will become less separable. In the case of the MNIST14 dataset 18 of the subproblems are S_3- and 27 are S_2-type. In the case of the MNIST7 dataset 2 of the subproblems are S_3- and 43 are S_2-type. In the case of the MNIST4 dataset 2 of the subproblems are S_2-, 3 are S_1- and 40 are S_0-type. An interesting observation is that S_2 (mutual convex separability) occurs more frequently than S_1 (non-mutual convex separability).

If two point set are not convexly separable (or not mutually convexly separable), then it is natural to ask that how far they are from convex separability (mutual convex separability). A possible measure of inseparability is $\mathcal{I}_{\mathcal{PQ}} = |\mathcal{P} \cap \text{conv}(\mathcal{Q})|$, the number of points from $\mathcal{P}$ that are contained in the convex hull $\mathcal{Q}$.

Class 1	**Class 2**								
	1	2	3	4	5	6	7	8	9
0	S_3	S_3	S_3	S_3	S_3	S_3	S_3	S_3	S_3
1		S_3	S_3	S_3	S_3	S_3	S_3	S_3	S_3
2			S_2	S_3	S_3	S_3	S_3	S_2	S_3
3				S_3	S_2	S_3	S_3	S_2	S_3
4					S_3	S_3	S_3	S_3	S_2
5						S_3	S_3	S_2	S_3
6							S_3	S_3	S_3
7								S_3	S_2
8									S_3

Table 1: Types of separability in the MNIST28 dataset.

Class 1	**Class 2**								
	1	2	3	4	5	6	7	8	9
0	S_3	S_2	S_2	S_3	S_2	S_2	S_3	S_2	S_2
1		S_2	S_2	S_3	S_3	S_3	S_3	S_2	S_2
2			S_2	S_3	S_3	S_3	S_3	S_2	S_3
3				S_2	S_2	S_2	S_2	S_2	S_2
4					S_3	S_3	S_3	S_3	S_2
5						S_2	S_2	S_2	S_2
6							S_3	S_2	S_3
7								S_2	S_2
8									S_2

Table 2: Types of separability in the MNIST14 dataset.

Class 1	**Class 2**								
	1	2	3	4	5	6	7	8	9
0	S_3	S_2	S_2	S_2	S_2	S_2	S_2	S_2	S_2
1		S_2	S_2	S_2	S_2	S_2	S_2	S_2	S_2
2			S_2	S_2	S_2	S_2	S_2	S_2	S_2
3				S_2	S_2	S_2	S_2	S_2	S_2
4					S_2	S_2	S_2	S_2	S_2
5						S_2	S_2	S_2	S_2
6							S_3	S_2	S_2
7								S_2	S_2
8									S_2

Table 3: Types of separability in the MNIST7 dataset.

Class 1	**Class 2**								
	1	2	3	4	5	6	7	8	9
0	S_0	S_0	S_0	S_1	S_0	S_0	S_1	S_0	S_0
1		S_0	S_0	S_0	S_0	S_0	S_0	S_0	S_0
2			S_0	S_1	S_0	S_0	S_0	S_0	S_0
3				S_0	S_0	S_2	S_0	S_0	S_0
4					S_0	S_0	S_0	S_0	S_0
5						S_0	S_0	S_0	S_0
6							S_2	S_0	S_0
7								S_0	S_0
8									S_0

Table 4: Types of separability in the MNIST4 dataset.

Class 1	Class 2									
	0	1	2	3	4	5	6	7	8	9
0	5923	57	36	67	3	110	22	0	389	21
1	36	6742	43	228	14	21	14	126	260	109
2	1	6	5958	34	0	3	3	4	16	2
3	1	74	151	6131	7	65	0	119	41	58
4	0	13	15	4	5842	27	17	102	9	490
5	1	8	1	34	4	5421	5	11	27	13
6	5	7	22	0	7	8	5918	0	13	2
7	1	17	7	44	50	1	0	6265	4	372
8	273	310	162	498	2	458	9	69	5851	160
9	2	66	5	129	933	42	3	945	65	5949

Table 5: Number of examples from Class 1 contained in the convex hull of Class 2 in the MNIST4 dataset.

The original versions of the presented convex separability algorithms are not able to determine the value of $\mathcal{I}_{\mathcal{PQ}}$, because they exit when the first inseparable outer point is found. However, it is trivial to modify the algorithms so that they calculate $\mathcal{I}_{\mathcal{PQ}}$. We just have to introduce a counter for the inseparable outer points and not exit when we find one.

The $\mathcal{I}_{\mathcal{PQ}}$ values for the MNIST4 dataset calculated by modified CSEPX$_{X2}$ are shown in Table 5. It is true to a certain degree that digit pairs considered to be more similar by humans have higher values. For example, there are 490 fours in the convex hull of nines and 933 nines in the convex hull of fours, but only 1 seven in the convex hull of zeros and 0 zeros in the convex hull of sevens.

5.4. Running times

5.4.1 Linear separability

The running times of basic methods for determining linear separability can be seen in Table 6. We get a measure of relative efficiency, if we rank the algorithms by running time for each problem, and then calculate the average rank of each algorithm [3]. According to

[3] If there is a tie from position A to position B, then the rank is considered as $(A + B)/2$.

Problem		Running time (seconds)					
		$LSEP_1$	$LSEP_1^*$	$LSEP_1^+$	$LSEP_2$	$LSEP_2^*$	$LSEP_S$
VOTES	S_1	0.089	0.035	0.039	0.134	0.034	>1800
WISCONSIN	S_1	0.177	0.033	0.034	0.228	0.032	>1800
MNIST04/24	S_1	122.0	1.01	0.89	135.9	1.03	>1800
MNIST04/49	S_0	87.3	0.94	0.91	97.0	0.87	>1800
MNIST07/01	S_3	259.7	2.10	2.06	299.6	2.03	15.1
MNIST07/02	S_2	310.1	2.54	2.59	333.1	2.59	>1800
MNIST14/01	S_3	744.0	9.56	9.60	1010.3	9.91	19.7
MNIST14/02	S_2	872.0	13.0	14.8	1243.5	14.9	>1800
MNIST28/01	S_3	>1800	97.9	107.7	>1800	56.3	36.3
MNIST28/02	S_3	>1800	232.7	232.2	>1800	179.9	52.7

Table 6: Running times of basic algorithms for determining linear separability.

this measure the order of basic methods on the given problems is the following: 1. $LSEP_2^*$ (1.95), 2. $LSEP_1^*$ (2.2), 3. $LSEP_1^+$ (2.45), 4. $LSEP_1$ (4.5), 5. $LSEP_S$ (4.6), 6. $LSEP_2$ (5.3). The first 3 methods are the dual based ones. The fastest primal based method was $LSEP_1$.

It is interesting that sometimes $LSEP_S$ was the fastest method, however its performance was not very stable. The running time of $LSEP_S$ was more than 1800 seconds in every linearly nonseparable case (and even in some linearly separable ones).

The running times of the proposed incremental algorithms (LSEPX, LSEPY, and LSEPZ) are shown in Table 7. The order of the methods according to the average rank measure is: 1. $LSEPX_2$ (2.3), 2. $LSEPY_2$ (2.5), 3. $LSEPX_1$ (2.6), 4. $LSEPY_1$ (3.6), 5. LSEPZ (4.2).

If we consider the average running time instead of the average rank, then the superiority of $LSEPX_2$ and $LSEPY_2$ is even stronger. For example, on the MNIST14/01 dataset $LSEPX_2$ was 12 times faster than $LSEPX_1$, and $LSEPY_2$ was 17 times faster than $LSEPY_1$. Recall that among the basic methods $LSEP_2$ was always slower than $LSEP_1$ (since $LSEP_2$ tries to find a small norm solution, and uses nearly twice as many variables to achieve this). It is interesting to see that it can pay off to apply the slower basic method in the incremen-

Problem		Running time (seconds)				
		$LSEPX_1$	$LSEPX_2$	$LSEPY_1$	$LSEPY_2$	LSEPZ
VOTES	S_1	0.029	0.041	0.029	0.038	0.11
WISCONSIN	S_1	0.049	0.031	0.048	0.055	0.081
MNIST04/24	S_1	0.27	0.28	0.47	0.37	2.63
MNIST04/49	S_0	0.26	0.36	0.36	0.36	2.37
MNIST07/01	S_3	1.08	0.68	1.55	0.69	4.52
MNIST07/02	S_2	0.39	0.49	0.67	1.57	12.7
MNIST14/01	S_3	31.2	2.44	36.2	2.10	7.45
MNIST14/02	S_2	5.38	3.59	5.67	4.78	133.5
MNIST28/01	S_3	>1800	388.0	>1800	88.1	14.8
MNIST28/02	S_3	>1800	640.6	>1800	284.9	143.2

Table 7: Running times of LSEPX, LSEPY, and LSEPZ.

tal algorithm. The reason for that is that the hyperplanes found by $LSEP_2$ have larger margin, and therefore less of them are needed.

On the largest datasets (MNIST28/01 and MNIST28/02) $LSEPY_2$ was significantly faster than $LSEPX_2$. This demonstrates that removing points from the active sets can convey large speedups in certain cases. It can also be observed that on the largest datasets the dual based LSEPZ was the fastest method, while on the other datasets it was the slowest one.

The running times of the proposed hybrid algorithms (LSEPZX and LSEPZY) are shown in Table 8. The order of the methods according to the average rank is: 1. $LSEPZY_2$ (1.9), 2. $LSEPZX_1$ (2.25), 3. $LSEPZX_2$ (2.6), 4. $LSEPZY_2$ (3.25). The differences between the running times are not as drastic as in the previous experiments. This is because the first step of the algorithm is the same in each case: running LSEPZ and possibly reducing the number of features. The unique second step is run only on the reduced dataset.

If we compare primal based basic methods with the proposed ones, then it turns out that the proposed algorithms are better. Typically, the speedup factor between a basic method

Problem		Running time (seconds)			
		LSEPZX$_1$	LSEPZX$_2$	LSEPZY$_1$	LSEPZY$_2$
VOTES	S_1	0.11	0.11	0.11	0.11
WISCONSIN	S_1	0.088	0.083	0.093	0.083
MNIST04/24	S_1	2.61	2.96	3.01	2.70
MNIST04/49	S_0	2.39	2.49	2.37	2.74
MNIST07/01	S_3	5.70	5.40	5.54	5.17
MNIST07/02	S_2	12.7	12.9	12.6	13.6
MNIST14/01	S_3	10.2	16.3	15.9	8.24
MNIST14/02	S_2	136.1	151.2	160.5	161.0
MNIST28/01	S_3	26.1	27.0	31.1	15.5
MNIST28/02	S_3	338.1	228.8	378.0	215.0

Table 8: Running times of LSEPZX and LSEPZY.

and its corresponding LSEPX/LSEPY/LSEPZX/LSEPZY variant is greater than 100.

The dual based basic methods (LSEP$_1^*$, LSEP$_1^+$ and LSEP$_2^*$) are sometimes slightly faster than the proposed LSEPX/LSEPY/LSEPZX/LSEPZY algorithms, but in contrast with the proposed methods they do not construct a separating hyperplane in the separable case.

5.4.2 Convex separability

The running times of basic methods for determining convex separability can be seen in Table 9. Each experiment consisted of two runs: at first the first class was the inner set, and then the second. The numbers shown in the table are the summed running times of the two runs (given in seconds).

The order of the basic methods according to the average rank measure is: 1. CSEP$_{X2}$ (2.15), 2. CSEP$_{Y2}$ (2.45), 3. CSEP$_{ZX2}$ (2.75), 4. CSEP$_{ZY2}$ (3.45), 5–6. CSEP$_2^*$, CSEP$_Z$ (5.1). The dual based methods (CSEP$_2^*$ and CSEP$_Z$) performed much worse than the other ones. This is because the dual based methods are not able to cut more than 1 outer point in 1 iteration. None of the basic methods was able to finish on MNIST28/02 in less than

Problem		Running time (seconds)					
		CSEP_{X2}	CSEP_{Y2}	CSEP_{ZX2}	CSEP_{ZY2}	CSEP_2^*	CSEP_Z
VOTES	S_1	0.78	0.80	1.47	1.48	4.38	5.36
WISCONSIN	S_1	0.62	0.53	0.89	0.92	5.42	4.13
MNIST04/24	S_1	44.0	49.4	136.8	143.4	>1800	>1800
MNIST04/49	S_0	4.63	5.65	11.8	24.7	>1800	>1800
MNIST07/01	S_3	10.3	10.8	29.2	29.4	>1800	>1800
MNIST07/02	S_2	58.1	58.8	152.6	170.3	>1800	>1800
MNIST14/01	S_3	54.6	51.8	68.7	67.8	>1800	>1800
MNIST14/02	S_2	>1800	>1800	310.5	375.1	>1800	>1800
MNIST28/01	S_3	>1800	>1800	221.3	222.4	>1800	>1800
MNIST28/02	S_3	>1800	>1800	>1800	>1800	>1800	>1800

Table 9: Running times of basic algorithms for determining convex separability.

1800 seconds.

The proposed CSEPX approach consist of two steps: at first the outer set is pruned by running the centroid (CSEPC) method, and then a basic method (CSEP) is run on the pruned dataset. The more outer points CSEPC can cut, the smaller problem the CSEP step has to solve, therefore the "pruning efficiency" of the CSEPC step can greatly influence the running time of CSEPX.

Table 10 shows the percentage of outer points cut by CSEPC for each dataset. (In each cell the values of the corresponding 2 runs are averaged.) We can see that the pruning efficiency was typically high on the given problems: in 8 cases it was greater than 90 %, and in 3 cases it was 100 %.

The running times of the proposed CSEPC and CSEPX methods can be seen in Table 11 and Table 12. The order of CSEPX variants according to the average measure rank is: 1–2. CSEPX_{X2}, CSEPX_{Y2} (2.4), 3. CSEPX_{ZX2} (3.75), 4. CSEPX_Z (4.05), 5. CSEPX_2^* (4.15), 6. CSEPX_{ZY2} (4.25). The difference between the running times is smaller than in the case of basic methods, since each CSEPX variant runs CSEPC as the first step.

If we compare the best basic method (CSEP_{X2}) with the best proposed one (CSEPX_{X2}),

Problem		Pruning efficiency	Problem		Pruning efficiency
VOTES	S_1	98.92 %	MNIST07/02	S_2	98.91 %
WISCONSIN	S_1	93.24 %	MNIST14/01	S_3	99.99 %
MNIST04/24	S_1	47.62 %	MNIST14/02	S_2	100 %
MNIST04/49	S_0	12.69 %	MNIST28/01	S_3	100 %
MNIST07/01	S_3	99.73 %	MNIST28/02	S_3	100 %

Table 10: Percentage of outer points cut by the centroid method (CSEPC).

Problem		Running time	Problem		Running time
VOTES	S_1	0.043	MNIST07/02	S_2	1.69
WISCONSIN	S_1	0.041	MNIST14/01	S_3	0.40
MNIST04/24	S_1	11.3	MNIST14/02	S_2	2.47
MNIST04/49	S_0	15.2	MNIST28/01	S_3	0.85
MNIST07/01	S_3	0.26	MNIST28/02	S_3	6.26

Table 11: Running times of the centroid method (CSEPC).

then we can observe the following: In the case of the S_0-typed MNIST04/49 problem the basic method was ~5 times faster, and in the case of the S_1-typed MNIST04/24 it was slightly faster. This situation can happen, because CSEP_{X2} exits immediately after finding an inseparable outer point, while the CSEPC step of CSEPX_{X2} cannot do this. In every other cases, CSEPX_{X2} was faster than CSEP_{X2}, often with orders of magnitude. For example, in the case of S_3-typed MNIST14/01 the speedup factor was ~60, and in the case of MNIST28/01 it was more than 2000. In terms of average and maximal running time the proposed CSEPX_{X2} method is far better than CSEP_{X2}, indicating that the performance of the proposed method is more stable.

6. Conclusion

In this paper, I proposed novel algorithms for determining the linear and the convex separability of point sets. First, I investigated the possibility of deciding linear separability with an incremental method. I suggested a linear programming based direct method (LSEP_2) that has favorable properties for being applied as a component of an incremental solution,

Problem	Running time (seconds)					
	CSEPX_{X2}	CSEPX_{Y2}	CSEPX_{ZX2}	CSEPX_{ZY2}	CSEPX_{2}^{*}	CSEPX_{Z}
VOTES	0.14	0.17	0.24	0.26	0.080	0.13
WISCONSIN	0.60	0.53	0.80	0.84	0.51	0.61
MNIST04/24	53.9	51.4	111.6	116.8	>1800	>1800
MNIST04/49	22.2	23.8	50.4	44.2	>1800	>1800
MNIST07/01	3.21	3.48	11.6	12.0	20.3	13.1
MNIST07/02	21.0	20.1	58.4	58.5	125.5	62.2
MNIST14/01	0.86	0.86	1.18	1.25	1.98	1.04
MNIST14/02	2.47	2.47	2.47	2.47	2.47	2.47
MNIST28/01	0.85	0.85	0.85	0.85	0.85	0.85
MNIST28/02	6.26	6.26	6.26	6.26	6.26	6.26

Table 12: Running times of enhanced algorithms for determining convex separability.

and I proposed heuristics (LSEPX, LSEPY, LSEPZX, LSEPZY) for choosing the active constraints and variables of the next iteration.

Second, I gave an approximate algorithm with low time requirement (CSEPC) and an exact algorithm with low expected time requirement (CSEPX) for the convex separation of two point sets. Third, I demonstrated by experiments on real-world datasets that the proposed algorithms compare favorably in running time with other known methods.

References

[1] E.M. Arkin, F. Hurtado, J.S.B. Mitchell, C. Seara, and S.S. Skiena. Some lower bounds on geometric separability problems. *International Journal of Computational Geometry and Applications*, 161:1–26, 2006.

[2] D. Ascher, P.F. Dubois, K. Hinsen, J. Hugunin, and T. Oliphant. Numerical Python, 2001. URL: http://www.numpy.org/.

[3] A. Asuncion and D.J. Newman. UCI Machine Learning Repository, 2007. URL: http://www.ics.uci.edu/~mlearn/MLRepository.html.

[4] C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[5] C. Chang and C. Lin. *LIBSVM: a library for support vector machines*. National Taiwan University, 2001. URL: http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[6] T. Finley. PyGLPK, version 0.3, 2008.
URL: http://www.cs.cornell.edu/ tomf/pyglpk/.

[7] T. Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in kernel methods - Support vector learning*. MIT Press, 1999.

[8] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for python, 2001. URL: http://www.scipy.org/.

[9] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[10] Y. LeCun and C. Cortes. The MNIST database of handwritten digits, 1999.
URL: http://yann.lecun.com/exdb/mnist/.

[11] A. Makhorin. GNU Linear Programming Kit, version 4.37, 2009.
URL: http://www.gnu.org/software/glpk/.

[12] N. Megiddo. On the complexity of polyhedral separability. *Discrete and Computational Geometry*, 3:325–337, 1988.

[13] John C. Platt. *Fast training of support vector machines using sequential minimal optimization*, pages 185–208. MIT Press, 1999.

[14] G. Takács and B. Pataki. Deciding the convex separability of pattern sets. In *Proc. of the 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS'2007)*, pages 278–80, 2007.

[15] Guido van Rossum. An introduction to Python, 2006.
URL: http://www.network-theory.co.uk/docs/pytut/.