AN INVESTOR RISK TOLERANCE ASSESSMENT USING INTERFACE AGENT IN MULTI-AGENTS DECISION SUPPORT SYSTEM

Andrius Jurgutis1,2, Rimvydas Simutis1

1Kaunas University of Technology, Department of Process Control, Studentu str. 50-163, Kaunas, Lithuania, andrius.jurgutis@stud.ktu.lt, rimvydas.simutis@ktu.lt
2Bank “Finasta”, Maironio str. 11, Vilnius, Lithuania

Abstract. Investment management is a complex and integrated task. This article introduces a multi-agent stock assessment system that helps to take investment decisions and researches one of the most important issues for the system - how to assess an investor risk tolerance, because system structure, system proposal structure and reliability depend on this assessment. To solve this problem we introduce our fuzzy logic expert system implemented in interface agent. The use of fuzzy logic in multi-agent system makes it more intellectual, i.e. the system is able to adapt automatically to the changeable investor risk tolerance. We also introduce a new questions and answers formulation principle used in the system that allows questioning an investor more precisely about his relation to risk.

Keywords: multi-agents decision support system, risk tolerance assessment, fuzzy logic.

1 Introduction

Nowadays more and more information technologies penetrate into different spheres of people’s life and activity. Modern information technologies let to replace a human being performing certain tasks, to automate various management processes and so on. The sphere of financial service provision, or to be precise finance and investment management (consulting) spheres in which information technologies influenced, influence and will influence this sphere using the elements of artificial intelligence such as fuzzy logic, genetic programming and neuron network and so on, is no exception.

The purpose of our works and researches is to create an intellectual multi-agent stock assessment information system. We can visually imagine this system as an autonomous virtual investment management and consultancy company wherein intellectual software agents perform operations which are usually performed by experts working in a traditional investment management company, i.e. economists, statisticians, finance brokers, analysts and others are replaced by software agents. Make a long story short, the multi-agent stock assessment system with regard to the situation in the financial market and the needs of an investor has to offer advice to an investor when and what stocks to buy/to keep/to sell, and to help to control an investment portfolio successfully. There are some reasons that encouraged creating such intellectual IT system.

Firstly, already available and commonly used new information technologies such as Internet throw down new challenges. When stock trading moved to electronic environment financial markets became global and dynamic. On the one hand, it is easy for an investor to make transactions in any world market, to give and get information, but on the other hand, on purpose to invest successfully it is necessary to comprehend the finer points of new financial markets, to keep a close watch on information, to be able to evaluate this information adequately and to take investment decisions quickly. It is unlikely that one man can do it while teamwork is hardly effective. So, investment management is a complex and integrated task including the solution of several tasks as well as the control and management of separate tasks [15], [23]. We and other scientists working in this field think that software agents are the most suitable to implement this system [11]. Actually, there are already such systems as, for example, WARREN, MASST [24].

Secondly, creating an intellectual multi-agent investment consulting IT system we tried to reach the additional underwritten goals:

- An intellectual IT Financial Management System as a new and additional service would allow increasing the profitability of consultancy and investment management companies.
- It would give a chance to render cheaper services for investors (for example, few commission fees for stock trade, reduced investment funds management expenses).
- It would quicken the investment decision-making.
- It would give greater confidentiality to an investor.
- An intellectual IT Financial Management System thanks to artificial intelligence would be able to adapt automatically to the changing needs of an investor.
• It is possible to transfer the experience of different experts to the IT system using expert systems, and thus to avoid human factor.

After reviewing in the introduction of the backgrounds of the origin of the intellectual multi-agent stock assessment system, in the second part of the work we briefly introduce the structure and the operating principle of our system. In the third and the forth parts of the work we review more precisely one part of the intellectual multi-agent stock assessment system, i.e. an interface agent. We pay a lot of attention to the question how an interface agent using fuzzy logic can assess an investor risk tolerance and how in the course of time the system will adapt to the changing investor risk tolerance.

2 The structure and operating principle of the information system

When creating the intellectual multi-agent stock assessment system we pondered that the system should be universal and include different kinds of investors, i.e. it should meet and correspond to the behavior of different investors in the market. To satisfy above mentioned expectations we are going to design the stock assessment information system using three types of agents: an interface agent, information agents and collaborating agents (task agents). These agents are notable for a number of features: autonomy, communicability, reactivity, sociability, mobility, pro-activity and others [11].

An interface agent is an intermediate between a user (investor) and a system [11]. The main functions of an interface agent are the following: to gather, analyze information about a user, to transmit user’s tasks to other elements of the system, to give generated responses of the system to a user, to inquire a user about additional information, to forecast the behavior of an investor, in case of the alteration of an investor risk tolerance to transform the structure of the system. The user of the system directly interacts only with an interface agent and does not see other agents. The main task of information agents is to provide tasks agents with information. For example, to find necessary information on the internet. Information agents are notable for strongly expressed autonomy, communicability, mobility and pro-activity features. Task agents perform a large amount of work to form investment proposal for an investor. Every task agent has got knowledge about one of stock assessment method. For example, technical analysis task agent is able to assess a stock using technical analysis means (knowledge); fundamental task agent assesses a stock using fundamental stock assessment theory and so on. At first out created system will include the following task agents: fundamental, technical analysis, risk, market and expert [11].

Besides, we know from the earliest times that investors according to their behavior can be divided into two main groups, i.e. conservative and aggressive investors. To meet the needs of different investors we offer two possible system structures, which thanks to the features of the agents can transform into each other. Both information system structures are showed in the Figure 1 and are called a) the horizontal structure of the information system, b) the vertical structure of the information system.

![Figure 1. Information system structure: a) horizontal structure, b) vertical structure](image)

Horizontal structure of the information system corresponds to the decision-making process of a conservative investor. Conservative investors are not inclined to risk, so they make investment decisions resting upon several stock assessment methods. The system with horizontal structure functions in the following way: with a help of an interface agent all necessary information about an investor is gathered and risk tolerance
tolerance consists of two constituents, i.e. an attitude towards risk and financial risk aspect [10], [17], [20].

client’s risk tolerance as the extent of the loss acceptable to an investor (i.e. financial attitude). Whereas risk
dominate either economical, or financial, or psychological? [12], [22]. Modern financial intermediaries see
assessment what is not good. In addition, there is a problem of questionnaire content, i.e. what questions should
recommendable questionnaire answering time should not exceed 15 min., questions should be interesting and not
so it is more difficult to assess it. It causes additional requirements when creating a questionnaire, for example,
boring to answer and so on.

Some state that not many [18], others on the contrary state that it should include many questions, because accuracy depends on it [17]. If there are few questions should a questionnaire of risk tolerance assessment include? The other problem is how many
questions should a questionnaire of risk tolerance assessment include? Some state that not many [18], others on the contrary state that it should include many questions, because accuracy depends on it [17]. If there are few questions in a questionnaire the weight of every one question becomes very significant for the final risk tolerance assessment what is not good. In addition, there is a problem of questionnaire content, i.e. what questions should dominate either economical, or financial, or psychological? [12], [22]. Modern financial intermediaries see client’s risk tolerance as the extent of the loss acceptable to an investor (i.e. financial attitude). Whereas risk tolerance consists of two constituents, i.e. an attitude towards risk and financial risk aspect [10], [17], [20].

It is necessary to take into consideration that a questionnaire can assess client risk tolerance inaccurately for a variety of reasons, for example, when a client consciously answers the questions loosely or lies; when he or she does not understand questions if they are long and difficult and require wide explanations, when questions are given with percentage, when complex financial and mathematical terms (for example, dispersion) are used and so on. Owing to this co-called risk tolerance assessment noise (error) is obtained. The inaccuracy of risk tolerance assessment appears because a questionnaire includes co-called “bad” questions that have nothing to do with risk tolerance assessment. The examples of bad questions are questions related to investment horizon, investment experience, investment goals that have nothing to do with risk tolerance assessment. These questions are more related to the investment strategy choice, to the selection of asset classes forming a part of investment portfolio.

Psychologists divide human behavior into two groups: intellectual behavior (when future sequence of actions is logically thought over in advance) and emotional behavior (when future actions take place on the spur of the moment, sometimes chaotically). Unfortunately, risk tolerance refers to the sphere of emotional behavior, so it is more difficult to assess it. It causes additional requirements when creating a questionnaire, for example, recommendable questionnaire answering time should not exceed 15 min., questions should be interesting and not boring to answer and so on.

3 Risk tolerance assessment problems

The success of the multi-agent stock assessment system depends on as accurate as possible investor risk
tolerance assessment. An interface agent transmits this parameter to other system agents, which decide what kind of information to look for. Unspecified investor (system user) risk tolerance will be responsible for false system recommendations (decisions) to an investor, i.e. inappropriate stocks will be offered to an investor and a client will have a loss or will not make desired profit. So, he will not be satisfied with the investment results, will not believe in the system and will stop using it.

Unfortunately, risk tolerance assessment is a complex task. This task has been discussed for a long time by different scientists, for example, by psychologists, economists [8], financial engineers, artificial intelligence creators [2]. There is no the only right and accurate way of risk tolerance assessment found. There are three ways of an inventor risk tolerance assessment: according to the investor autobiography, according to the reports about conversations and communication with an investor and according to special questionnaire. As our system is not so intelligent that it could read the biography of an investor or communicate with an investor we have chosen a questionnaire. At the moment the most popular means of risk tolerance assessment is also a questionnaire. The first questionnaire was published in 1984 [5].

Carried out and published scientific researches also showed that risk tolerance assessment is a complex
task. They state that a client can better assess risk tolerance himself than financial advisers can do it by means of questionnaire. Correlation coefficient of these assessments amounts 0.4 [18]. The other problem is how many
questions should a questionnaire of risk tolerance assessment include? Some state that not many [18], others on the contrary state that it should include many questions, because accuracy depends on it [17]. If there are few questions in a questionnaire the weight of every one question becomes very significant for the final risk tolerance assessment what is not good. In addition, there is a problem of questionnaire content, i.e. what questions should dominate either economical, or financial, or psychological? [12], [22]. Modern financial intermediaries see client’s risk tolerance as the extent of the loss acceptable to an investor (i.e. financial attitude). Whereas risk tolerance consists of two constituents, i.e. an attitude towards risk and financial risk aspect [10], [17], [20].

It is necessary to take into consideration that a questionnaire can assess client risk tolerance inaccurately for a variety of reasons, for example, when a client consciously answers the questions loosely or lies; when he or she does not understand questions if they are long and difficult and require wide explanations, when questions are given with percentage, when complex financial and mathematical terms (for example, dispersion) are used and so on. Owing to this co-called risk tolerance assessment noise (error) is obtained. The inaccuracy of risk tolerance assessment appears because a questionnaire includes co-called “bad” questions that have nothing to do with risk tolerance assessment. The examples of bad questions are questions related to investment horizon, investment experience, investment goals that have nothing to do with risk tolerance assessment. These questions are more related to the investment strategy choice, to the selection of asset classes forming a part of investment portfolio.

Psychologists divide human behavior into two groups: intellectual behavior (when future sequence of actions is logically thought over in advance) and emotional behavior (when future actions take place on the spur of the moment, sometimes chaotically). Unfortunately, risk tolerance refers to the sphere of emotional behavior, so it is more difficult to assess it. It causes additional requirements when creating a questionnaire, for example, recommendable questionnaire answering time should not exceed 15 min., questions should be interesting and not boring to answer and so on.
To evaluate the reliability and accuracy of the risk tolerance assessment questionnaire some methods are used. One of them is based on the correlation between risk tolerance value stated by a client himself and risk tolerance value stated by a financial intermediary. The second method is based on the measurement and evaluation of standard deviation of risk tolerance value \([3, 4]\). The third method is based on the evaluation of the correlation between answered pairs of questions of the questionnaire. These issues are widely researched by Roszkowski \([17]\).

Client risk tolerance assessment is not only the problem of client’s cognition; it is also client’s perception problem, so it is necessary to the extent possible to use several methods to establish a client risk group \([18]\).

To assess client risk tolerance Hanna \([7]\) singles out at least four methods, i.e. to evaluate risk tolerance it is necessary to question about the investment alternatives choice \([9]\) and the combinations of financial means in the portfolio, to ask subjective questions \([6, 13]\), to ask questions that can estimate current investor’s behavior \([1, 16]\), to ask hypothetical questions with clearly stated scenarios of financial market behavior. Roszkowski \([19]\) offered to combine all these methods, i.e. to ask an investor different questions. Later there were other methods offered based on the theory of utility, demographic research and others. \([18]\). However scientific researches showed that using of different methods often causes different investor risk tolerance assessments.

In summary, we want to share our practical notices. In practice the majority of financial intermediaries does not meet or place much emphasis on the usage of mentioned requirements for the assessment of risk tolerance. After the analysis of questionnaires offered to investors we can state that most of them are aimed at the identification of portfolio content and sales strategy. When an investor answers the questions at the end of the questionnaire he can find portfolio structure according to asset classes, whereas a financial intermediary cannot know his client and identify investor risk tolerance and is danger of losing a client in future. When we create our system we will try to avoid this. We also want to notice that a questionnaire includes questions with clearly stated 3 – 5 answer variants, each of these answers has an appraisal by points, whereas the final conclusion about risk tolerance is made after evaluation of scored points. In our opinion the accuracy of the questionnaire will suffer from this, because a limited in advance number of answer variants in principle defines which risk group an investor belongs to, i.e. an investor choosing one of limited answer versions attaches himself to a certain risk group, or in other words, an answer variant is closely linked with a corresponding risk group. Creating our system we offer to abandon a close (stated) link between an answer and a risk group. Therefore, we offer to use in the questionnaire the other question formulation principle, i.e. an answer to the question should not be determined (stated) in advance. An investor should answer the question in free format. In our opinion it would be innovative to include into the questionnaire some statements about an investor and investment, and an investor could answer on a scale of 0 to 100 to what extent this statement is true for him, to what degree he would use an offer and so on. We think that this principle would let an investor to open out better than answering to clichéd questions, so it is possible that risk tolerance assessment will be more precise.

Finally, we can draw some conclusions on the base of which we create (establish) our risk tolerance assessment method. Firstly, to measure risk tolerance we will use a questionnaire. Secondly, when creating our questionnaire we will avoid “bad” questions. Thirdly, after the evaluation of the conclusions of scientific researches and recommendation our questionnaire will consist of 17 questions-statements (15 questions, 2 statements). Fourthly, as there are still many unknowns in risk tolerance assessment we think that it is advisable to use fuzzy logic technique for the assessment of risk tolerance. Hereafter we will review our offered fuzzy logic system for the assessment of risk tolerance.

4 Fuzzy set logic system for risk tolerance assessment

Before starting to work out in detail how does an interface agent assesses investor risk tolerance using fuzzy logic we introduce an introduction which will help to understand the operating principle and structure of the system. An interface agent takes initial information about an investor from investor’s registration form and the questionnaire. The questionnaire consists of 15 questions and 2 statements. All these questions and statements are related only to the risk tolerance assessment. General data (for example, age, sex, yearly income and so on) about an investor is gathered during the logging on to the multi-agent stock assessment system. This data does not take part in the risk tolerance assessment, so we do not give much consideration to it in this article. A system’s user can review all the data further. We made the questionnaire on the ground of above mentioned conclusions. We also divided the questions into four groups, i.e. general questions, hypothetical questions, experience questions and risk questions. Later you will see that this division is important for the formation of system structure and the formulation of the final conclusion. Every group includes four questions – statements and we numbered them accordingly from \(Q_1\) to \(Q_{17}\). In the first questions \(Q_1\) an investor is asked what risk group he personally belongs to (very low risk tolerance, low risk tolerance, average risk tolerance, high risk tolerance, and very high risk tolerance). Later using fuzzy set logic and the answer to the first question the final conclusion
about an investor risk tolerance is drawn. As you can understand the answers to the questionnaire questions will be input data of fuzzy logic system for the risk tolerance assessment.

Theory implies that the construction of fuzzy set logic expert system consists of four steps: fuzziness introduction (fuzzification), the creation of expert assessment rules, aggregation (the summing up of the conclusions of assessment rules), and fuzziness elimination (defuzzification) [14]. We will discuss all there steps in detail, and we will start from the representation in the Figure 2 of the overall structure of our offered fuzzy set logic system for the risk tolerance assessment. As we can see in the Figure 2 the fuzzy logic system consists of two separate fuzzy logic systems, i.e. fuzzy logic system 1 and 2. The division is made in order that the system could adapt to the changing investor risk tolerance. We can record (fix) the change of the investor risk tolerance in two ways. In one case an investor voluntarily refills the questionnaire because he is not satisfied by the results given by the system or if some changes in his life appear (for example, marriage, the acquisition of new experience, knowledge), that change his attitude toward risk. Or the system can automatically ask an investor to refill the questionnaire because an investor has not overviewed it longer than two years. In this case the system fully renews risk tolerance assessment, i.e. both fuzzy logic systems 1 and 2 are activated. In another case the system observes what was the extent of risk of the financial means an investor bought/sold, and this signal goes only to the fuzzy logic system 2; as a result an iteration process is formed during which after each investor’s transaction its risk tolerance is reassessed. For this purpose a new input variable is brought in the fuzzy logic system 2, which shows last transaction’s stock risk group, whereas two other input parameters remain constant. So, in the first case the system quickly adapts to changes, whereas in the second case it adapts gradually. Later we will work out in detail the fuzzy logic system 1, because it will consist of some fuzzy logic systems with cascade structure. Meanwhile the fuzzy logic system 2 is implemented as a simple fuzzy logic system with three input parameters and one output parameter, so it is unnecessary to discuss it. Hereafter, let us proceed to the fuzziness introduction.

![Figure 2. Fuzzy set logic system structure for the risk tolerance assessment](image)

We have already mentioned that both fuzzy logic systems 1 and 2 (refer to Figure 2) receive data from the questionnaire. In the second section we mentioned that we would use another questions and answers formulation principle. The difference between the questionnaire offered by us and the questionnaire used in modern practice is graphically showed in the Figure 3.

![Figure 3. The comparison of questions and answers formation: a) usual formation, b) fuzzy formation](image)
A usual questions and answers formation principle (refer to Figure 3a) is different in that the answer to the questions are strictly determined (stated), i.e. the fixed number of clearly formulated answers is given, and a respondent should only choose a suitable to him answer which can be inaccurate expression of respondent’s opinion. Our fuzzy questions and answers formation principle is showed in Figure 3b. In this case a respondent does not choose an answer from given, he shows his answer on scale, where he indicates his attitude towards this question or statement. Getting such an answer form a respondent we not only find out his opinion, but also his state (we have a possibility to get a feel for a respondent, to measure his doubts). Later using membership function (denoted by symbol $\mu$ in fuzzy set) we find what determined (stated) answer (linguistic variable) or conclusion a fuzzy answer corresponds to. An example of membership function is showed in Figure 4. There you can see how to attach a risk tolerance linguistic variable to a fuzzy answer. We will use a trapezoidal membership function, because it naturally suits to solve the problem under consideration. In future we will try to use other functions. Hereafter, let us proceed to the creation of expert assessment rules.

![Figure 4. The example of risk group membership function](image)

Expert assessment rules are established by the experts in this field. We need experts who can write the rules: how should the fuzzy logic system 1 assess (evaluate) on the basis of 16 obtained fuzzy answers to what risk group an investor belongs to. In our situation we are going to be the experts ourselves. We have trust in our own 9-year experience in investor consultancy.

The form of the fuzzy logic expert system rules is showed by the Formula 1.

$$\text{IF } Q_2 \text{ is } X_2 \text{ AND } ... \text{ AND } Q_{17} \text{ is } X_{17} \text{ THEN } Q \text{ is } X$$

(1)

Where $Q$ – all questions set, $X$ – all possible risk groups set. Using the fuzzy logic theory [21], we can easily calculate, that having 16 questions and when it is possible to relate each question to one of five risk groups, a total number of rules necessary for the drawing of the final conclusion will be very high, i.e. some hundred million rules. Owing to this a rule written in such a form will be inefficient. To reduce the number of rules we offer to make (to form) from smaller fuzzy logic systems (with two input parameters and one output parameter, binary model) a cascade structure that is showed in Figure 5. In such a way a total number of rules will reduce up to 375 rules. We intuitively feel that there will be some rules that will repeat. This means that a real number of rules will be even fewer. This cascade structure lets to gain one more advantage. As you know we divided questions into four groups. In the Figure 5 we can see that at first conclusions about an investor risk tolerance are drawn on the ground of the pair of questions, later (the second layer) on the ground of the group of questions (in Figure 5 groups stand out against various background: rarely spotted background – general questions, densely spotted background – hypothetical questions, vertical lines in background – experience questions, horizontal lines – risk questions). Then conclusions are drawn on the ground of contiguous groups of questions. The groups of questions are arranged using some logic and providence, i.e. input parameters of the last fuzzy logic system should be the conclusions drawn on the ground of general-hypothetical and experience-risk questions. So, we think that it is logically that a conclusion drawn on the ground of experience-risk questions would determine “the rush” of the final conclusion towards extreme risk tolerance assessment, whereas the conclusion drawn on the ground of general-hypothetical questions would determine “the center” or “the average” of the final conclusion. Hereafter, some words about aggregation and fuzziness elimination.

Aggregation – for the summing up of the conclusions of assessment rules we will use Mamdami method. For fuzziness elimination we will use “center” method. These constructional elements of the fuzzy logic system remain unchanged, such s they described in fuzzy set theory [2], [14], [21].
5 Conclusions and future works

In this article we introduced one of the elements of the multi-agent stock assessment system – an interface agent – and worked out in detail how this element of the multi-agent system assesses an investor risk tolerance. The task of an investor risk tolerance assessment is complex and exceedingly important. The reliability and the accuracy of the whole system depend on how effectively we will complete this task. To complete this task we offer to use fuzzy logic. We also offer a new approach to this problem. First of all we offer to question an investor about his relation to risk on the ground of another principle, i.e. we offer to assess risk tolerance using fuzzy answers and questions-statements. This allows to know an investor better or even to find...
his doubts. On purpose to reduce the number of the rules of the expert system we offer to use a new cascade structure made up from fuzzy logic systems.

Risk tolerance assessment system is implemented with a help of “MatLab Fuzzy Tool Box”. At the moment we have gathered too little statistical data to draw presumptive conclusions about the reliability and accuracy of the system for risk tolerance assessment. Next researches and comprehensive system’s tests will be conducted to ascertain the accuracy of the system, to find optimal membership function forms. In respect that the system has to operate in internet environment we will seek for the means to implement this system in this environment, i.e. we will research the possibilities to implement this system by means of “FuzzyJ Toolkit”, “FuzzyTECH” or others.

References
DISCOVERY OF PERSONALIZED INFORMATION SYSTEMS USAGE PATTERNS

Inese Supulniece, Janis Grabis

Riga Technical University, Information Technology Institute, Meza 1/3, LV 1048, Riga, Latvia, 
Inese.Supulniece@rtu.lv, grabis@iti.rtu.lv

Abstract. Users use information systems to accomplish their tasks usually consisting of multiple steps. Each user or user group might have a preferred sequence of the steps. Adaptive information systems attempt to exploit such usage patterns, and it is expected that adaptivity can be improved through personalizing processes and accounting for temporal and sequential dependencies. However, there is a lack of empirical evidence of existence of personalized information systems usage patterns. The objective of this paper is to analyze empirical information systems usage data in order to confirm existence of personalized information systems usage patterns, which could be further used in developing adaptive information systems. Empirical data analyzed are derived from log files of customers service website of a telecommunication company and of university’s e-learning system. The Longest Common Subsequence algorithm is used to discover patterns in task execution. It is found that frequency of patterns observation varies significantly between the data sets although in both cases personalized patterns are observed more frequently than general patterns. The personalized patterns also are more precise than the general patterns.

Keywords: transaction logs, temporal data mining, patterns, process adaptation.

1 Introduction

User uses information systems to perform various tasks. Each task consists of multiple steps and often there are multiple alternative sequences of steps, which can be used to accomplish the same or closely related tasks. Sequences, which are used most frequently, can be referred to as task execution patterns. Various data mining [8] and process mining [12, 13] techniques are used to discover such patterns. Data mining is able to discover hidden patterns, unexpected trends or other subtle relationships in the data using a combination of techniques from machine learning, statistics and database technologies. Temporal data mining is concerned with data mining of large sequential data sets what is important in the case of analyzing task execution sequences (see Laxman and Sastry [8] for a recent overview of temporal data mining methods). Process mining techniques are mainly aimed at recovering process structure for process validation and performance improvement purposes.

Adaptive information systems attempt to exploit the task execution pattern to accommodate users in order to expedite execution of their tasks. The overall goal of the proposed research is to elaborate a process driven adaptive business information system, which is conceptually described in [17]. One of the main characteristics of the proposed adaptive information system is an ability to predict and to recommend future task execution steps on the basis of actions performed by the user and historically recorded behavior of this user or user groups. Some of available adaptive information systems (for example, [5, 11]) do not explore temporal and sequential dependencies and process mining techniques deal with general processes rather than user specific task execution sequences. It is expect in this research that efficiency of adaptivity can be improved through personalizing processes and accounting for temporal and sequential dependencies. However, there is a lack of empirical evidence of existence of user specific or personalized information systems usage patterns. Personalized patterns are patterns, which repeatedly occur in task execution sequences of individual users, while general patters are patters, which occur in task execution sequences regardless of users.

The objective of this paper is to analyze empirical information systems usage data in order to confirm existence of user specific information systems usage patterns, which could be further used in developing adaptive information systems. Two main hypotheses to be tested are that: 1) personalized patterns in user’s task execution sequences are observed more frequently than general patterns; and 2) personalized patterns have higher level of confidence than general patterns. Two empirical information systems usage data sets are used in the paper. These data sets are derived from log files of customers service website of telecommunication company and of university’s e-learning system. The Longest Common Subsequence algorithm [3] is used to discover patterns in task execution. In order to evaluate discovered patterns, two measures commonly used in data mining, namely, support and confidence [8] are used. The support measure indicates the frequency of pattern observation, and the confidence measure is a proxy for pattern precision measurement.

The rest of the paper is structured as follows. Section 2 introduces concepts and notation used in the paper. The data analysis and pattern evaluation methodology and empirical data used in the analysis are described in Section 3. Empirical data analysis is presented in Section 4. Section 5 contains related work and Section 6 concludes.
2 Information System Usage Patterns

It is assumed that a user performs various actions in the information system to accomplish her/his tasks. These actions are recorded in a log file as events. Each event is characterized by its name, time and user. Events sequentially involved within a specified timeframe form an information system usage events sequence or session \( S(u) \in S \), where subscript \( i \) indicates the session number in the log file and varies from 1 to the number of sessions \( N \), \( u \) indicates session’s user and \( S \) is the set of all sessions in the log file. It is expected that the sessions contain information system usage patterns. The pattern is defined as a sub-sequence of events what is present within at least two sessions. Given two sequences of events: \( S_1 = \{a_1, a_2, \ldots, a_n\} \) and \( S_2 = \{b_1, b_2, \ldots, b_m\} \), a pattern is such a sub-sequence \( P(S_1, S_2) = \{d_1, d_2, \ldots, d_y\} \), where \( P \subseteq S_i \) and exists integers \( i_1 < i_2 < \ldots < i_y \) such that \( d_i \subseteq a_i \) , \( d_2 \subseteq a_{i_1} \) , \( \ldots \) , \( d_y \subseteq a_{i_y} \). Patterns are denoted by \( P \subseteq S_2 \), where exists integers \( j_1 < j_2 < \ldots < j_y \) such that \( d_i \subseteq b_{j_1} \), \( d_2 \subseteq a_{j_1} \) , \( \ldots \) , \( d_y \subseteq a_{j_y} \). Patterns are denoted as \( P_k = \{d_1, \ldots, d_M\} \), where \( k \) is the number of pattern, \( d_n \) is the \( n \)th event in the pattern and \( M_k \) is the number of events in the pattern. Events in the pattern might not necessarily follow each other immediately, i.e. there could be other events between them.

Patterns are classified as general patterns and personalized or user specific patterns. Let \( S_u \) denotes all sessions performed by user \( u \), then general patterns are searched within the set \( S \setminus S_u \), while the personalized patterns are search within the set \( S_u \) for each user. Personalized patterns are denoted by \( P_{u_k} \), where \( k \) indicates the number of the pattern and \( u \) indicates pattern’s user.

Patterns are evaluated using their support and confidence measures. These measures are computed for each pattern separately. Support measures frequency of pattern occurrence within information system usage session. It is defined as

\[
F_k = \frac{\sum_{i=1}^{N} Z_i}{N}, \quad k = 1, \ldots, L, \tag{1}
\]

where \( Z_i \) indicates presence of the \( k \)th pattern in the \( i \)th session

\[
Z_i = \begin{cases}
1, & \text{if } P_k \subseteq S_i \\
0, & \text{if } P_k \not\subseteq S_i.
\end{cases}
\]

For example, given sessions \( S_1 = \{A, B, C\}, S_2 = \{A, C, D\}, S_3 = \{D, G\} \). The support for pattern \( P = \{A, C\} \) is \( 2/3 \) because sequence \( A \rightarrow C \) is presented in session \( S_1 \) and \( S_2 \), but not in \( S_3 \).

Confidence is perceived as a measurement of pattern precision and is calculated

\[
C_k = \frac{\sum_{i=1}^{N} Z_i}{\sum_{i=1}^{N} Y_i}, \quad k = 1, \ldots, L, \tag{2}
\]

where \( Y_i \) indicates presence of the right trimmed \( k \)th pattern \( P_k^* = P_k \setminus d_{M_k} \) in the \( i \)th session

\[
Y_i = \begin{cases}
1, & \text{if } P_k^* \subseteq S_i \\
0, & \text{if } P_k^* \not\subseteq S_i.
\end{cases}
\]

Essentially, the confidence measures a likelihood of a user completing the pattern once it has started to execute it. For example, given sessions \( S_1 = \{A, B, C\}, S_2 = \{A, C, D\}, S_3 = \{D, G\} \). The confidence for pattern \( P = \{A, C\} \) is calculated 1 because sequence \( A \rightarrow C \) is presented in two session \( S_1 \) and \( S_2 \), but not in \( S_3 \) and \( \sum_{i=1}^{N} Y_i = 2 \) because the pattern without the last element \( P^* = \{A\} \) is presented in 2 sessions, \( S_1 \) and \( S_2 \).

\( F_k(u) \) and \( C_k(u) \) for personalized patterns are computed using equations (1) and (2), respectively, where only sessions of the \( u \)th user are considered (i.e., \( S_u \)).

3 Analysis Methodology

The aim of the analysis is to confirm presence of personalized patterns in empirical information systems usage data. The presence of such patterns would suggest that it is possible to develop adaptive information systems, where processes are adapted according to individual preferences of users rather than according to some general rules for all users or user groups. In order to confirm presence of personalized patterns, empirical data are used to test two hypotheses:

H1. Personalized patterns are observed more frequently than general patterns;

- 26 -
H2. Personalized patterns have higher confidence (i.e., they are more precise) than general patterns. If both hypotheses are confirmed then individual patterns are considered more efficient than general patterns because users more frequently follow their personalized patterns and adaptation according to personalized patterns would be more accurate.

The five step methodology is developed for analysis purposes: (1) Data preparation (clean transaction logs from unnecessary information; group user activities into user sessions and transform data into format that is more convenient for analysis); (2) Finding general patterns (sequences of events that are similar to several or all users); (3) Finding individual patterns (sequences of events that are similar to each individual user); (4) Filtering of results; (5) Evaluation of general and individual patterns.

The main parts of the methodology are pattern discovery (using association rules identification algorithm) and evaluation of derived patterns. General and individual patterns are extracted from the same log files.

3.1 Data Description

Two data sources are used for analysis of information systems usage patterns, namely, log files of customers service website of telecommunication company (CS) and log files of university’s e-learning system (ES).

Description of CS

CS has two functions: a representative function (anonymous user) and a service portal function (identified user). In the representative part, there is information about company, about its products, contacts and, additionally, there are a lot of advertisements (special offers) and possibilities to apply for products. In the service portal part, there are possibilities to view and pay bills, view statements, etc. There is also functionality for a webpage administrator. Thus administration tasks are also included in the log file. The portal is more a flat application compared to workflow systems, so user navigation within the portal is not restricted by some predefined flows.

The analyzed log files contain data from 3 days including user activities and also system responses. The total number of entries in the log file is ~128000 transactions (rows) and ~600 unique users (determined according to IP addresses).

Description of ES

The second set of data comes from log files of ES. This portal is used only by identified users (students and course owners). Students can review information about their classes, homeworks, grading and other teaching resources, submit their homeworks and communicate with each other. Additionally, there is also functionality for course owners to add and update course related information. Similarly as in the case of CS, there are no predefined workflows and users have high degree of flexibility.

The analyzed log files contain data from 1 month. – They include only user activities. The total number of entries in the log file is ~10000 transactions (rows) and 44 unique users.

3.2 Data Preparation

Original logs of transactions from real systems require initial cleansing to adjust formatting. Described steps are similar for any web usage mining problem and are discussed in [2]. Additionally, content of logs from real systems might include also not relevant entries. For example, the HTTP protocol requires a separate connection for every file that is requested from the Web server. Therefore, a user’s request to view a particular page often results in several log entries since graphics and scripts are down-loaded in addition to the HTML file. In most cases, only the log entry of the HTML file request is relevant and should be kept for the user session file. This is because, in general, a user does not explicitly request all of the graphics that are on a Web page, they are automatically down-loaded due to the HTML tags. Since the main intent of pattern discovery is to get a picture of the user’s behavior, it does not make sense to include file requests that the user did not explicitly request.

CS Data Preparation

Data preparation includes the following activities:

- removing not relevant entries from log files;
- merging different events (URLs) that have the same result for user. For example, URL = ../admin/ means the same as URL = ../admins or URL = ../cat=1 means the same as URL = ../about_company;
- listing all possible unique events (events with the same meaning are listed as one item) and all possible unique users. Replacing events with identifiers;
• listing all user sessions (splitting all user events per user sessions. We defined new user session in the case there was more than 10 minutes interval between events of the same user). It is important to note that a user can perform several processes within one session;
• removing all user sessions that include only one activity, because these users have visited webpage accidentally and did not execute any process in the portal.

Before data preparation, there were ~2000 unique events and ~800 user sessions. After data preparation, there were 842 unique events. The result of data preparation is presented in Figure 1.

<table>
<thead>
<tr>
<th>Input data</th>
<th>Data preparation</th>
<th>Output data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log files</td>
<td>X remove not relevant transactions;</td>
<td>List of unique events</td>
</tr>
<tr>
<td></td>
<td>X delete all user sessions that include only single event</td>
<td>ID Event</td>
</tr>
<tr>
<td></td>
<td>+ add for every transaction meaning event and identifier;</td>
<td>1001 Start page</td>
</tr>
<tr>
<td></td>
<td>+ create list of user sessions;</td>
<td>1002 About company</td>
</tr>
<tr>
<td></td>
<td>X delete all user sessions that include only single event</td>
<td>1003 Save application</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1004 Information about product X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1005 Information about product Y</td>
</tr>
</tbody>
</table>
|           |                             | ... ... ...

Figure 1. The data preparation process for CS

**ES Data Preparation**

Data preparation includes the following activities:
• listing all possible unique events (events with the same meaning are listed as one item) and all possible unique users. Replacing events with identifiers.
• listing all user sessions (splitting all user events per user session). It is important to note that a user can perform several processes within one session.
• Removing all user sessions that includes only one event.

There are 29 unique events and 2325 user sessions. Result of data preparation is presented in Figure 2.

<table>
<thead>
<tr>
<th>Input data</th>
<th>Data preparation</th>
<th>Output data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log files</td>
<td>+ add for every transaction meaning—event and identifier;</td>
<td>List of unique events</td>
</tr>
<tr>
<td></td>
<td>+ create list of user sessions;</td>
<td>ID Event</td>
</tr>
<tr>
<td></td>
<td>X delete all user sessions that include only single event</td>
<td>1001 Start page</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1002 Course view</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1003 Resource view</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1004 Assignment view all</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1005 Assignment view submission</td>
</tr>
</tbody>
</table>
|           |                             | ... ... ...

Figure 2. The data preparation process for ES

### 3.3 Finding and Filtering of Patterns

The pattern is defined as a sequence of events what is present within at least two sessions. For example, if one session contains events \{A, B, C, D, E\} and other session contains events \{M, B, D, E, G\} then the following sequence of events is present on both sessions \{B, D, E\}, and it is referred to as a pattern. These event items might not follow each other immediately and there could be events between them.

Patterns can be identified using different temporal data mining algorithms [8]. The Longest Common Sub-sequence algorithm is one of the most simple but also efficient algorithms. The longest common subsequence (LCS) problem is to find the longest subsequence \(L\) common to all sequences in a set of sequences (often just two). For the general case of an arbitrary number of input sequences, the problem is NP-hard. For the case of two sequences of \(T_i\) and \(T_j\) elements, the running time of the dynamic programming approach is \(O(T_i \times T_j)\). \(L\) is determined using stepwise comparison of every two session or sequences of events. The following expression is evaluated on every step of the comparison:

\[
L(S_i^0, S_j^0) = \begin{cases} 
0 & \text{if } i = 0 \text{ or } j = 0 \\
L(S_i^0, S_j^0) \cup a_i & \text{if } a_i = b_j \\
L(S_i^0, S_j^0^{-1}) & \text{if } a_i \neq b_j \land |L(S_i^0, S_j^0^{-1})| > |L(S_i^0, S_j^0)| \\
L(S_i^0^{-1}, S_j^0) & \text{if } a_i \neq b_j \land |L(S_i^0^{-1}, S_j^0)| < |L(S_i^0, S_j^0^{-1})|
\end{cases}
\]

(3)
\(L(S_i, S_j)\) denotes an intermediate longest common sequence, and \(S_i\) and \(S_j\) denotes \(i\)th and \(j\)th prefixes of sessions \(S_1\) and \(S_2\), respectively. If two intermediate longest common sequences are of the same length but not identical, then both are retained. LCS does not necessarily yield unique results, for example the longest common sequence of \(\{A, B, C\}\) and \(\{A, C, B\}\) is both \(\{A, B\}\) and \(\{A, C\}\). Patterns are derived from all longest common sequences obtained during the stepwise comparison.

Patterns consisting of zero or one event are deleted (see figure 3). 0-item pattern means that there is nothing common between two sessions. 1-item pattern means that there is just one common event and that cannot be used as a basis for process oriented adaptation.

Patterns are evaluated using support and confidence measures as defined using Eq. 1 and Eq. 2, respectively. These measures are computed separately for general and individual patterns. Maximum possible support and confidence is 1 or 100%. If the support of pattern is closer to 1, then it is a popular pattern. If the confidence of pattern is closer to 1, then it is a precise pattern. The pattern is perceived as an efficient (i.e., it could be used efficiently for process adaptation) pattern if both support and confidence are high. A pattern efficiency chart is used to visualize pattern efficiency (Figure 4). The chart positions each discovered pattern according to its support and confidence levels. It is divided in four quadrants.

Each quadrant contains patterns with certain characteristics:

- **Quadrant I** – popular and precise patterns (confidence more than 50% and support more than 50%);
- **Quadrant II** – not so popular, but precise patterns (confidence more than 50% and support below 50%).
- **Quadrant III** – not so popular and not so precise patterns (confidence below 50% and support below 50%).
- **Quadrant IV** – popular, but not so precise patterns (confidence below 50% and support more than 50%).

It is obvious that the best patterns are within Quadrant 1 and not efficient patterns are within Quadrant 3. Therefore, patterns in each quadrant are counted separately for general and user patterns, and association between counts is used to check the overall hypothesis that personalized patterns are more efficient than general patterns.
4 Results

The methodology described above is used to analyze empirical information systems usage data obtained from CS and ES. Analysis results are presented for each information system separately.

Figure 5 shows the pattern efficiency charts for CS and ES data. One point in the charts can represent multiple patterns if their values of support and confidence numbers are the same. In the case of CS data, there is some contamination due to insufficient pre-processing of the log files. Visually, it can be observed that the general patterns are mainly located in the third quadrant while personalized patterns also frequent in the first and second quadrants.

In Figure 5a there is a point (personalized pattern) that has maximum support and confidence, in reality under this point are represented 75% of all personalized patterns. In this set are many patterns consisting of 2 events and some of these 2-item patterns are repeated by various users, but length of ‘ideal’ patterns are not limited only to 2 items. There are also patterns with length of 3, 4, 5 and 6 events. Some of these patterns are executed only by single user, others – by group of different users.

In Figure 5b someone could recognize that all personalized patterns are located on top of line where \( F_k = C_k \). There is no special interpretation, just for every single personalized pattern Confidence always seems to be higher than support or equal to it. This tendency was not observed with general patterns.

Table 1. Cross-tabulation results

<table>
<thead>
<tr>
<th></th>
<th>CS data</th>
<th></th>
<th>ES data</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>General patterns</td>
<td>Personalized patterns</td>
<td>General patterns</td>
<td>Personalized patterns</td>
</tr>
<tr>
<td>Quadrant I</td>
<td>0.2</td>
<td>91.3</td>
<td>0.0</td>
<td>0.9</td>
</tr>
<tr>
<td>Quadrant II</td>
<td>22.0</td>
<td>7.4</td>
<td>65.5</td>
<td>79.2</td>
</tr>
<tr>
<td>Quadrant III</td>
<td>75.2</td>
<td>1.3</td>
<td>34.5</td>
<td>20.0</td>
</tr>
<tr>
<td>Quadrant IV</td>
<td>2.7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Pearson Chi-Square = 11853, DF = 3, P-Value = 0.000

Numerical evaluation of pattern efficiency is obtained by cross-tabulation (Table 1) showing percentage of patterns falling in each quadrant for general patterns and for personalized patterns. It can be observed that 0.2% of general patterns and 91.3% of individual patterns are precise and popular at the same time for CS data (Quadrant I). At the same time 75.2% of general patterns and only 1.3% of individual patterns fall in Quadrant III (least desirable situation). It means that data from these log files confirm our assumption that individual patterns are more effective. Additionally, personalized patterns have very high efficiency and almost all patterns are very precise and popular. That means that if users return to webpage, their activities are the same almost in all sessions. For the ES data, it can be observed that 0% of general patterns and 0.9% of individual patterns are precise and popular at the same time. The majority of both general and personalized patterns belong to Quadrant II while fewer personalized patterns belong to Quadrant III. That means that also personalized patterns in discovered from the ES data are slightly more efficient than general patterns. The Chi-square test of cross-tabulation data confirms that the position of patterns in the pattern efficiency chart depends upon pattern personalization. Given that majority of personalized patterns fall in Quadrant I, personalized patterns are more efficient than general patterns at least in the case of CS data.
Table 2. Descriptive statistics of the support and confidence measures

<table>
<thead>
<tr>
<th></th>
<th>$F$</th>
<th></th>
<th>$C$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>General patterns</td>
<td>Personal patterns</td>
<td>General patterns</td>
<td>Personal patterns</td>
</tr>
<tr>
<td>CS data</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>0.03</td>
<td>0.85</td>
<td>0.39</td>
<td>0.96</td>
</tr>
<tr>
<td>St. Dev.</td>
<td>0.06</td>
<td>0.27</td>
<td>0.34</td>
<td>0.14</td>
</tr>
<tr>
<td>Median</td>
<td>0.01</td>
<td>1.00</td>
<td>0.30</td>
<td>1.00</td>
</tr>
<tr>
<td>ES data</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>0.01</td>
<td>0.06</td>
<td>0.65</td>
<td>0.75</td>
</tr>
<tr>
<td>St. Dev.</td>
<td>0.02</td>
<td>0.08</td>
<td>0.29</td>
<td>0.31</td>
</tr>
<tr>
<td>Median</td>
<td>0.00</td>
<td>0.04</td>
<td>0.67</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Non-parametric Mann-Whitney test was used to compare support and confidence values for personalized and general patterns to test H1 and H2 presented in Section 2. Probability that the median value for personalized patterns is larger than median value for generalized patterns approaches zero for both measures. Therefore, both hypotheses are accepted implying that the personalized patterns are observed more frequently and their predictive power is stronger than that of general patterns. It has to be noted more frequent observation of personalized patterns could have been expected because of size differences between $S$ and $S_u$. However, the magnitude of differences suggests that using general patterns as a basis of adaptation could be misleading. The finding that personalized patterns have higher confidence is not so obvious and suggests that individual users have strong tendency to use information systems in a predictable manner what could be exploited in design of adaptive information systems.

There is a large difference of pattern support level between CS and ES data. That could be explained by using process groups in CS data while process groups have not been used in ES data. It appears that users use only a limited number of processes. Therefore, patterns belonging to one process group are not likely to occur in sessions during which user primarily considers processes from unrelated process groups.

5 Related work

Analysis of transaction logs had been widely used to understand user behavior within the system, e.g. to characterize user search behavior in information retrieval systems [7, 9, 15], to do process mining to improve quality of services and processes [12, 13] or for extracting business rules from information systems [14]. Most of the studies concentrate on mining usage of web systems. The overall process of web usage mining is generally divided into two main tasks: data preprocessing and pattern discovery. Data mining methods employed during pattern discovery include association rule mining (e.g. [2]), sequential pattern discovery (e.g. [6]), clustering (e.g. [5]) and classification (e.g. [1, 10]) or mix of the aforementioned methods (e.g. [18]). Usage patterns extracted from web data can be applied to a wide range of applications such as web personalization, system improvement, site modification, business intelligence and usage characterization [16].

According to Gerry [4] and Wang [19], the accuracy and coverage rate of association mining techniques is usually quite low. They applied the association mining over all users' navigation sessions to establish a knowledge model that predicts users’ next request. Their results also show that the sequence mining method produces higher accuracy than association mining. The association patterns do not perform well in prediction of future navigation patterns due to the low matching rate of prediction rules. Some improvements are made, when instead of performing the association mining task over all users’ navigation sessions, users are first clustered so that users in each cluster demonstrate shared navigation characteristics [18].

However, none of the approaches analyzes personal dimension of discovered patterns. The navigational behavior of the current user is matched with the ones of other users with similar profiles, so the adaptation is done according to the profiles of other similar users, not according to patterns of a particular user.

6 Conclusion

The analysis of empirical information systems usage data has been performed to highlight differences between general system usage and behavior of individual users who might prefer their personalized ways information system usage. The empirical data obtained from log files of CS and ES confirm the hypotheses that personalized pattern are observed more frequently than general patterns and that personalized patterns have higher confidence or predictive power. From the perspective of the proposed development of process-oriented adaptive information systems, these findings imply that users are more likely to accept adaptive systems that are based on personalized patterns and that the usage of personalized patterns could result in more accurate adaptive behavior of the system (e.g., system’s recommendations about further steps of the process execution would be
more accurate). Acceptance of both hypothesis and cross-tabulation results show that overall efficiency of personalized patterns is better than that of general patterns.

However, it can be observed that efficiency of patterns varies significantly even between just two empirical data sets. That implies that expected gains from adaptivity also would vary significantly. Low level of pattern support in the case of ES implies that an adaptive system would have to support very large number of different process variations what could be technically challenging. It has been identified that process groups might have significant role in prediction of pattern efficiency, it is stated also in [10], where term ‘content’ refer to the same meaning as ‘process group’ in this paper.

References


SENSITIVITY ANALYSIS FOR QUANTITATIVE DECISION MAKING METHODS: TOPSIS AND SAW

Ruta Simanaviciene¹, Leonas Ustinovichius²

¹Vilnius Gediminas Technical University, Department of Building Technology and Management, Sauletekio al. 11, LT-10223 Vilnius, Lithuania, leonasu@st.vtu.lt
²Vilnius Gediminas Technical University, Department of Information systems, Sauletekio al. 11, LT-10223 Vilnius, Lithuania, ruta.simanaviciene@vgtu.lt

Abstract The quantitative multiple criteria decision making methods are practical to use in decision support systems. In this paper we analyses the quantitative multiple criteria decision making methods and sensitivity analysis methods usage in decision support systems. Both species of these methods are strongly mathematically based. We take notice of sensitivity of these methods to initial data. Monte Carlo method is applied for the generation initial data. It is submitted analysis sensitivity of methods TOPSIS and SAW.

Keywords: decision support systems, decision making, quantitative methods, multiple criteria evaluation, TOPSIS, SAW, sensitivity analysis, Monte Carlo method.

1 Introduction

The various activity fields’ database has the data of quantitative form. Accordingly, quantitative multiple criteria decision making methods (MCDM) are practical to use in decision support systems. In this paper the quantitative multiple criteria decision making methods are proposed to use in decision support systems. These methods are strongly mathematical based. Quantitative multiple criteria decision making methods are these: Linear assignment method, Simple additive weighting method, Hierarchical additive weighting method, ELECTRE methods, TOPSIS method.

Often data in multiple criteria decision making (MCDM) problems are imprecise and changeable. Therefore, an important step in many applications of MCDM is to perform a sensitivity analysis on the input data. The significance of quantitative criteria is usually determined with some errors. If measurements are not accurate, the result obtained is not accurate either, but sensitivity of the result may be checked by varying the parameters. [1]

If we scrutinize the standard decisions relevant to construction technology and management, we shall become certain that deficiency of information is very often ignored. Experts make use of unfavourable initial data, their values applied are exaggerated, work is executed with poor quality models which, in case of need, are a bit corrected on the basis of practical experience, however reflect the actual situation insufficiently. Acting in such a way, experts make allowable decisions, but most often these decisions are unfavourable. While researching into regularities, deficiency of information is attempted to evade. Application of regularities enables to evaluate results of necessary actions and to present the direction of their selection. Simple evaluation of all possible actions is not always sufficient. Each action may cause several sometimes contradicting each other results. As the actual result is not known, solutions criteria are necessary, which could take into consideration the totality of possible results. [12] If we want to take more accurate decision results, so it must make sensitivity analysis to a method, which is used in decision making.

A possible definition of sensitivity analysis (SA) is the following: The study of how uncertainty in the output of a model (numerical or otherwise) can be apportioned to different sources of uncertainty in the model input [8].

Sensitivity analyses are commonly used and the reader is referred to Clemen (1996), French and Rios Insua (1999), or Rios Insua and French (1991) for a perspective on different approaches to sensitivity analyses in multicriteria decision making. The goal of sensitivity analyses is to investigate the impacts of the uncertainties to the model. Sensitivity analyses can be used for a wide range of purposes. Pannell (1997) grouped these into four categories:

1. Decision making (identifying critical values/parameters, testing robustness, overall riskiness of decision);
2. Communication (increasing commitment/confidence/credibility, explicitly showing critical assumptions);
3. Increased understanding (understanding relationship between input/output variables);
4. Model development (identifying needs for more accurate measurements/more information).

In this paper we analyses the quantitative multiple criteria decision making methods and sensitivity analysis methods usage in decision support systems. Both species of these methods are strongly mathematically
based. We take notice of sensitivity of these methods to initial data. Monte Carlo method is applied for the
generation initial data. It is submitted analysis sensitivity of methods TOPSIS and SAW.

2 Related works

A number of essential disadvantages of the criteria were given in the paper (E. K. Zavadskas et al, 2007). These are: a definition of the attributes significances is complicated and in some cases, their relationship
with the efficiency parameters is weak; the small size of one parameter is compensated by a superfluous size of
another; the average success criterion sensitivity of the made decision to the changes of the size of separate
efficiency parameters is low, especially, if the attribute number is great. The authors gave sensitivity analysis of
a SAW method in point of the values of the attributes significances. They used two methods of normalization

In mathematical programming sensitivity analysis is typically understood as analysis of changes of an
optimal solution caused by change of the data in the model. A traditional approach of such analysis is based on
properties of an optimal solution. It typically consists of calculations of ranges of changes of parameters for
which an optimal solution does not change and on using a dual solution for calculations of changes of value of a
goal function for changes of some parameters that are small enough for allowing such a simple evaluation
procedure [7].

The Fondazione Eni Enrico Mattei (FEEM) research institution developed decision support system
mDSS in the years 2008. This system utilises two approaches for SA: (i) most critical criterion: identifying the
criterion for which the smallest change of current weight may alter the existing ranking of options; and (ii)
tornado diagram: graphically comparing the chosen option with any other one and showing ranges within which
the parameters may vary.

Most critical criterion

This method developed by (Triantaphyllou 2000) considers the most critical criterion to be the one
which requires the amount of change in the current value of its weight in order to change the options’
rank order. Using this method, the user may directly test if the minimal change in criteria weights which leads to
the final rank disturbance is within or outside his confidence range. The method distinguishes two rank order
changes:

• The top critical criterion is that one which changes the best ranked option.
• Any critical criterion is that one which changes ranking of any options.

Tornado diagram

The tornado diagram is a graphical method of sensitivity analysis. The advantage of this approach is its
visual representation of sensitivity that compares two options (a basic and a challenging one) at a time. The
horizontal bars represent the ranges of the options’ total performance obtained by the variation of each weight.
Bars are arranged from widest to narrowest and thus produce a “tornado” shape. On the x - axis the difference in
total performance between compared solutions is shown. Notice that the zero points – with equal performance of
both options – correspond to the weights obtained from the critical criterion approach [4].

Some decision methods (for instance, the AHP, SAW, TOPSIS, …) require that the aij values represent
relative importance. Given the below data and a decision making methods, the objective of the decision maker is
to find the best (i.e., the most preferred) alternative or to rank the entire set of alternatives [2].

Table 1. Decision matrix

| Alt. | C1   | C2   | C3   | ... | CN
| Alt. | W1   | W2   | W3   | ... | WN |
| A1   | a11  | a12  | a13  | ... | a1N
| A2   | a21  | a22  | a23  | ... | a2N
| A3   | a31  | a32  | a33  | ... | a3N
| ...  | ...  | ...  | ...  | ... | ...
| AM   | aM1  | aM2  | aM3  | ... | aMN

In a simple MCDM situation, all criteria are expressed in terms of the same unit (e.g., euros). However,
in many real life MCDM problems different criteria may be expressed in different dimensions (units). Examples
of such dimensions include dollar figures, weight, time, political impact, environmental impact, etc. It is this
issue of multiple dimensions which makes the typical MCDM problem to be a complex one [2].

In own paper (1997) Triantaphyllous described, that a change of 0.03 is very different if the original
value was 0.08 or 0.80. That is, it is more meaningful to use relative changes. As Dantzig (1963, p. 32) stated it:
"Sensitivity analysis is a fundamental concept in the effective use and implementation of quantitative decision
models, whose purpose is to assess the stability of an optimal solution under changes in the parameters.” By knowing which data are more critical, the decision maker can more effectively focus his/her attention to the most critical parts of a given MCDM problem [2].

3 Proposed approach

When the requirements to the initial data to be used in multiple criteria evaluation methods are not defined, it is hardly possible to say that the result obtained in using these methods is reliable. In the present work, it is offered to carry out sensitivity analysis for quantitative decision-making methods, and produce decision-making result with its probability of occurrence. Only then, the methods discussed may be used in decision support system.

When slight variation of the initial data parameter results in the considerable changes in the final results, it is stated that the latter are sensitive to this parameter. This usually implies that either the parameter should be determined very accurately or that the alternative should be redesigned for lower sensitivity.

This paper analyzes the sensitivity of SAW and TOPSIS methods.

SAW method

Mathematically, simple additive weighting (SAW) method can be stated as follows: Suppose the decision maker assigns a set of importance weights to the attributes, \( \bar{q} = \{\bar{q}_1, \bar{q}_2, \ldots, \bar{q}_n\} \). Then the most preferred alternative \( A^* \), is selected such that

\[
A^* = \left\{ A_i \left| \max_j \left( \sum_{j=1}^{n} \bar{q}_j \frac{x_{ij}}{\sum_{j=1}^{n} \bar{q}_j} \right) \right. \right\}
\]

(1)

Where \( x_{ij} \) is the outcome of the \( i \)-th alternative about the \( j \)-th attribute with a numerically comparable scale. Usually the weights are normalized so that

\[
\sum_{j=1}^{n} \bar{q}_j = 1
\]

(2)

Simple additive weighting method requires a comparable scale for all elements in the decision matrix. The comparable scale is obtained by using equality

\[
\bar{x}_{ij} = \frac{x_{ij}}{x_j^{\text{max}}}
\]

(3)

for benefit criteria and by equality

\[
\bar{x}_{ij} = \frac{x_j^{\text{min}}}{x_{ij}}
\]

(4)

for cost criteria [5].

TOPSIS method

Yoon and Hwang (1981) developed a technique based on the idea that the optimal alternative is most similar to an ideal solution (being closest to it and at the longest distance from the negatively ideal solution). This method is known as TOPSIS – Technique for Order Preference by Similarity to Ideal Solution.

A relative distance of any \( i \)-th variant from an ideal one is obtained by the formula:

\[
K_{\text{BRT}} = \frac{L_i^+}{L_i^+ + L_i^-}, i = 1, m
\]

where \( L_i^+ \) is a distance between the compared \( i \)-th variant and the ideal one; \( L_i^- \) is a distance between the compared \( i \)-th variant and the negatively ideal alternative. The nearer to one is \( K_{\text{BRT}} \) value, the closer is the \( i \)-th variant to \( a^+ \), i.e. the optimal variant is the one which has the highest value of \( K_{\text{BRT}} \) [6].

Relative standard deviation

The relative standard deviation (RSD) is often times more convenient. It is expressed in percent and is obtained by multiplying the standard deviation (s) by 100 and dividing this product by the average (\( \bar{x} \)).

- 35 -
\[ RSD = \frac{S}{\bar{x}} \times 100 \]  

(6)

In our work, we propose to carry out sensitivity analysis for quantitative multiple criteria decision making methods, before using them for developing the decision support systems.

First, we use Monte Carlo simulation for initial data generation. We generate \( n \) samples from a normal distribution. The mean represents the value of the criterion value; the standard deviation represents 10% from the criterion value. We carry out the alternatives evaluation by SAW and by TOPSIS with these generated initial data. We get \( n \) decisions by SAW and by TOPSIS, calculate relative standard deviations to each alternative evaluation.

We illustrate the sensitivity analysis by graph, and the final decision provide alongside the results of sensitivity analysis, by relative standard deviation.

The main approach is presented, compared and demonstrated by means of analytical and practical examples.

4 Case study

To illustrate our proposed approach, some alternatives of purchasing an office building for a company are considered. Suppose that the clients (DMs) need to purchase office premises. There are four variants (A1 – A4) of office location. Four criteria are considered: \( X_1 \) – price (10,000 $); \( X_2 \) – office area (m²); \( X_3 \) – distance from home to work (km); \( X_4 \) – office location (in points). The criteria \( X_2 \) and \( X_4 \) are maximized, while \( X_1 \) and \( X_3 \) are minimized. The data concerning office purchasing for a firm is presented in Table 2.

<table>
<thead>
<tr>
<th>Alternatives</th>
<th>Criteria ( X_1 )</th>
<th>( X_2 )</th>
<th>( X_3 )</th>
<th>( X_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>3.0</td>
<td>100</td>
<td>10</td>
<td>7</td>
</tr>
<tr>
<td>A2</td>
<td>2.5</td>
<td>80</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>A3</td>
<td>1.8</td>
<td>50</td>
<td>20</td>
<td>11</td>
</tr>
<tr>
<td>A4</td>
<td>2.2</td>
<td>70</td>
<td>12</td>
<td>9</td>
</tr>
</tbody>
</table>

Having made the calculations, we obtained the following significance values of the criteria. They are presented in Table 3.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>( X_1 )</th>
<th>( X_2 )</th>
<th>( X_3 )</th>
<th>( X_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subjective values of criteria</td>
<td>0.092</td>
<td>0.264</td>
<td>0.184</td>
<td>0.460</td>
</tr>
</tbody>
</table>

In Table 4 (below), we see the data, obtained by calculating the effectiveness of the alternatives by TOPSIS.

<table>
<thead>
<tr>
<th>Methods</th>
<th>( A_1 )</th>
<th>( A_2 )</th>
<th>( A_3 )</th>
<th>( A_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alternative evaluation by TOPSIS</td>
<td>0.514</td>
<td>0.391</td>
<td>0.583</td>
<td>0.638</td>
</tr>
<tr>
<td>Rank of alternative by TOPSIS</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Alternative evaluation by SAW</td>
<td>0.75912</td>
<td>0.671</td>
<td>0.758</td>
<td>0.75910</td>
</tr>
<tr>
<td>Rank of alternative by SAW</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

According to the results of evaluation, a decision may be made that the alternative A4 is optimal by TOPSIS method and that the alternative A1 is optimal by SAW method.

We performed the generation of the initial data. 20 values of each criterion \( x_{ij} \) were generated with standard deviation of 10% from the average value of each criterion. This result given in Figure 1
The alternatives are evaluated according to SAW and TOPSIS methods. The graphs show the results of alternative evaluations in Figure 2 and Figure 3 accordingly, by changing the initial data.

The graphs (Figure 4 and Figure 5) demonstrate that the ranking for alternatives is sensitive to changes of initial data. On purpose to produce effective decision, the decision maker needs to carry out analysis of initial data repeatedly.

We calculate the relative standard deviations to the alternative rationality values (effectiveness) (Table 5) and make the conclusions about the final decision.

The final decision can be summed up as follows, after alternatives evaluation and sensitivity analysis:

- TOPSIS method is more sensitive than SAW method;
- When initial data vary by 10%, the decision results by SAW have the relative standard deviation 14.26%; the decision results by TOPSIS have the relative standard deviation 5.48%.
- We offer to give the final decision, (it using data from the table 4 and table 5) as follows: the most effective alternative by SAW A1, with the bias ± 5.937%; the most effective alternative by TOPSIS A4, with the bias ± 13.521%. 

Figure 1. Initial data after generation

The alternatives are evaluated according to SAW and TOPSIS methods. The graphs show the results of alternative evaluations in Figure 2 and Figure 3 accordingly, by changing the initial data.

Figure 2. Estimates of alternatives by SAW

Figure 3. Estimates of alternatives by TOPSIS

Figure 4. Rang of alternatives by SAW

Figure 5. Rang of alternatives by TOPSIS
### Table 5. RSD of each alternative

<table>
<thead>
<tr>
<th>Methods</th>
<th>Alternative</th>
<th>$A_1$</th>
<th>$A_2$</th>
<th>$A_3$</th>
<th>$A_4$</th>
<th>Average of RSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSD of alternative evaluations given by TOPSIS</td>
<td></td>
<td>15,568</td>
<td>12,371</td>
<td>15,589</td>
<td>13,521</td>
<td>5.48</td>
</tr>
<tr>
<td>RSD of alternative evaluations given by SAW</td>
<td></td>
<td>5,937</td>
<td>5,156</td>
<td>4,756</td>
<td>6,074</td>
<td>14.26</td>
</tr>
</tbody>
</table>

5 **Conclusions and Future works**

The paper addresses the problem of determining sensitivity of quantitative methods SAW and TOPSIS. The performed sensitivity analysis with respect to initial data allowed the authors to draw the following conclusions:

1. If the initial data (values of the criteria) differ by 10% from the average criterion values, TOPSIS method is more sensitive, than SAW method;
2. The authors suggest performing sensitivity analysis of decision making methods with respect to the initial data, which may be not sufficiently accurate. This applies both to the values and weights of the criteria used. The final decision should be provided alongside the results of sensitivity analysis;
3. Using quantitative multiple criteria decision making methods, it is importantly to do sensitivity analysis for these methods. It must be taken into account, when developing decision support systems.

In future, we want to do the analysis of the multiple criteria decision making problems, to determine which methods (more or less sensitive) for the chosen problems are most effective.

6 **Acknowledgements**

The authors would like to thank two anonymous reviewers for their helpful comments.

**References**


[3] Friedl L. Decision support systems: Methods to extend science to decision making. American Geophysical Union, Fall Meeting. 2006


A LANDSCAPE OF LEGAL TELOELOGY: FORMALIZATION THROUGH VISUALIZATION

Vytautas Cyras
Vilnius University, Faculty of Mathematics and Informatics, Naugarduko 24, Vilnius, Lithuania, vytautas.cyras@mif.vu.lt

Abstract. The paper advocates a view that visualization and symbolization precede formalization on the way to legal knowledge representation. Hence, an attempt is made to formalize the semantics of a fundamental legal concept. The visual metaphor of a continuous landscape with a path is used to convey the landscape of legal teleology; it is called a sinnlandscape. The metaphor is fit with the discrete path definition in a graph. In the survey, our earlier notation, which consists of a two-element relation (action-goal or means-telos), is extended to three elements, such as an initial situation, a path and a goal. Thus the visualization can have a favorable effect on the modeling of legal teleology and formal representation. The examined issues comprise the entities of the sinnlandscape, subjective and objective teleological interpretation, and teleological and causal relations. Positive and negative evaluations of means and goals are combined in the analysis of the statement “The end justifies the means.”

Keywords: Legal informatics, teleological interpretation, visual metaphor, spatialization, formalizing semantics.

1 Introduction

There are many approaches to formalizations in law. Here various formalisms, notations, logics and modeling techniques are used. Symbol level representations of legal teleology are extensively used in artificial intelligence (AI) and law; see e.g. [2]. We find the knowledge level representation [10] a true challenge for legal informatics. In this way, a knowledge engineer meets a problem of understanding the contents of fundamental legal concepts, such as the nature of law, a legal method, legal interpretation, justice, value, etc. Though, jurists know the contents. The concepts are studied at different levels of abstraction in specific branches of law, legal dogmatics, legal theory, jurisprudence, and legal philosophy.

The paper presents a kind of a cognitive approach to legal teleology. A new metaphor for visualization is proposed. The path metaphor consists of three elements:

initial situation – path – goal

This metaphor supplements the earlier proposed notation which consists of a two-element relation (action-goal or means-telos) [5]:

$A \rightarrow B$ (1)

For example, $A = \text{open\_the\_door}$ and $B = \text{fresh\_air}$, or $A = \text{close\_the\_door}$ and $B = \text{security}$.

Using this notation, the purpose of a street as defined in Austrian law $^1$ is represented as follows:

$\text{Straße} = (\text{Landfläche} \rightarrow (\text{Fußgängerverkehr} \lor \text{Fahrzeugverkehr})$

$\& (\text{Anlagen} \rightarrow \text{Verkehr}) )$ or in English:

$\text{street} = (\text{land\_area} \rightarrow (\text{pedestrian\_traffic} \lor \text{vehicle\_traffic})$

$\& (\text{facilities} \rightarrow \text{traffic}) )$

2 The Entities of the Sinnlandscape

In the paper, legal teleology is comprehended as a landscape, sinnlandscape (Sinnlandschaft) of the legal domain. $^2$ Different subjects (individuals, companies, associations, political parties, states, etc.) have different purposes. Their actions are ruled by very different laws. The entities of the sinnlandscape are depicted in Figure 1. There is no uniform list of values which are equally preferred by all the subjects. Therefore triangle arrows show different directions. A statement about the teleological relation $A \rightarrow B$ is denoted by $\text{stmt}(A \rightarrow B)$. For instance, “John says that opening a door serves to let the fresh air in.” A reflection about te-structures is

$^1$ Straßenververkehrsgesetzbuch 1960 (StVO), §2, para. 1. Straße: eine für den Fußgänger- oder Fahrzeugverkehr bestimmte Landfläche samt dem in ihrem Zuge befindlichen und diesem Verkehr dienenden baulichen Anlagen.

$^2$ Eppler and Burkhard emphasize the role of visual cognition and perception and note that a majority of our brain’s activity deals with processing and analyzing visual images: “If visualization is applied correctly, it dramatically increases our ability to think and communicate” [6, p. 553].
expressed by meta-statements. A kind of them claims that goal entities are part of individual and collective sinnlandscapes. This is a primary consideration of the paper.

The visual metaphor of a path in a landscape is a kind of a continuous paradigm in legal visualization. Thus, the discrete nature of logic is supplemented. Continuous and discrete models are not opposites but emphasize different viewpoints to the same system. Formal (mathematical) logic has certain limitations, especially when applied to law, legal reasoning and modeling of legal argumentation. Stability which is characteristic of a continuous system might be a certain advantage when comparing with a discrete system. Small changes in the input of a continuous system usually cause small changes in its output. A discrete system may cause big changes, for instance, a slightly wrong initial choice can lead to a seriously wrong destination.

![Diagram](image)

**Figure 1.** The goal entities (Zweckentitäten) of the sinnlandscape (Sinnlandschaft).

A reflection about teleological structures, a meta-statement, claims that the goal entities are part of individual and collective sinnlandscapes

### 3 Why Legal Teleology?

Why is teleology important? Several reasons can be listed:

1. Teleology is innate in normative legal systems.
2. The people reason primarily by goals and roles then by rules.
3. Teleological structures are mostly implicit and rarely explicit.
4. Teleological statements are extensively used in legal drafting.

In argument analysis, Woods emphasizes agent’s targets and their implicit character [13]:

“Argument and reasoning succeed or fail in relation to an agent’s targets. Such arguments are cognitive in nature, typified by a desire to know what to believe and what to do. Targets are usually contextually cued and implicit in an agent’s behavior. They are for the most part presumed rather than declared.”

The nature of legal teleology is complex and multi-level. One may speak about the purpose of the statute (ratio legis), the purpose of a norm, the purpose of an action, the purpose of an entity (e.g., a street lane and a pavement), a promoted/demoted value, etc. Therefore, different formalizations can be built.

Modeling of legal argument is a central but not the only concern in AI and law. Legal theory classifies the functions of law into three categories: the **judicial** function, the **legislative** function, and the **executive** function. The research in the paper concerns applications in law making.

We focus on formalizing the teleological interpretation method. Peczenik [11, p. 329-339] examines teleological construction of statutes. To assure the logical character of the basic structure of the interpretation of a statute in view of its purpose, Peczenik added the third premise in the structure below [11, p. 329]:

| Premise 1:    | Obtaining of the situation Z is prescribed |
| Premise 2:    | If one had not do H, then Z would not be obtained |
| Premise 3:    | If |
|              | 1) obtaining of the situation Z is prescribed; and |
|              | 2) if one had not do H, then Z would not be obtained; |
|              | then one should do H |

Conclusion: One should do H
Peczenik distinguishes between the *subjective* and *objective* teleological interpretation of statutes. The limitations of logic are noted in [11, p. 330]:

“...The purpose of the statute (*ratio legis*) as regards hard cases differs from the will of the persons that participated in the process of legislation. Neither the *ratio* nor the proposed construction of statutes follows logically (emphasis added) from the description of the will alone.”

As the purpose does not follow logically, the linguistic (literal) and logical methods of interpretation are not enough in goal-based reasoning. “The step from the text of the statute and data concerning the will of its ‘authors’ to the *ratio legis* is a jump” [11, p. 331]. This jump constitutes a qualitatively new step which can be depicted as the semantic graph in Figure 2.

![Figure 2](image_url)

**Figure 2. A jump from the text of the statute and data concerning the will of its “authors” to the purpose of the statute**

We hold that the jump serves to get over the limitations of formal logic, too. The use of the teleological method requires an invention, heuristics and discovery. This is important in hard cases, whereas logic copes well in ordinary cases. The nature of logic lies more in explanation than in discovery. Finiteness of assertion sets is essential in computationally effective models based on formal logic.

The objective-teleological construction of statutes is expressed in [11, p. 331]:

Premise 1: According to an interpretation, supported by various juristic substantive and authority reasons, the provision, L, is a means to fulfill the goal, Z

Premise 2: If one had not interpreted L as containing the rule R, then Z would not be obtained

Conclusion: One should interpret L as containing the rule R

We think that listing the purposes of a statute in its preamble is too little. In order to fully understand the purposes, more elaborate structures should be developed. For example, a first step might be to annotate the structural parts of the statute giving the purposes.

The statute has to be treated as a system [4]. The purposes of the statute have to form a coherent system. A formalization in the form of a list of atomic goals is too little. The purposes have to be attributed richer semantics. They are subject to multi-dimensionality, ordering and different evaluation by social groups.

4 Goal Modeling in Legal Argument

This section illustrates how goals are addressed in modeling legal argument. First we choose an approach to practical reasoning with agents, which is proposed by Atkinson, Bench-Capon and McBurney, see e.g. [1]. They consider the *sufficient condition scheme* from [12] and unpack Walton’s notion of a goal into three elements. We depict the three elements as the semantic graph in Figure 3.

![Figure 3](image_url)

**Figure 3. Unpacking Walton’s goal notion into three elements [1]**

Representation of the goal model as a semantic graph can be suitable for both computer scientists and legal scholars. We depict the argument scheme AS1 from [1] as in Figure 4.
Figure 4. The argument scheme AS1 from Atkinson and Bench-Capon [1]

Goals are concerned in Bench-Capon and Prakken’s logical (hence, discrete) formalization of an argument-based reasoning [3]. They formalize goals as propositional literals in defeasible modal logic. The formalism is illustrated with an example of a judge who must determine the best way to punish \( (pu) \) a criminal found guilty. He has three options: imprisonment \( (pr) \), a fine \( (fi) \) and community services \( (cs) \). Besides punishment there are three more goals: deterring the general public \( (de) \), rehabilitating the offender \( (re) \) and protecting society from crime \( (pt) \). Punishment, \( (pu) \), is the most important goal, but the method of punishment chosen will depend on the other goals. The argument language is comprised of the following formulas. Imprisonment promotes punishment, \( pr \Rightarrow pu \), imprisonment demotes rehabilitation, \( pr \Rightarrow \neg re \), etc. The judge’s goal base is the set \( G = \{ Dpu, Dpt, Dde, Dre \} \), where \( D \) is the modality operator. Arguments are represented as trees. They are the most interesting part of the example and the reader is directed to [3].

We hold that the spirit of legal teleology is in the “sea” of concrete goals which are innate in substantive law. This is in accordance with the idea that the essence of a knowledge-based system is in the knowledge base (comprised of a vast amount of knowledge chunks), not in the inference engine. Further the landscape metaphor is used to visualize this “sea”.

5 The Landscape as a Surface

The notion of a landscape can be formalized as a surface in mathematics, namely, a mapping \( f \) from \( n \)-dimensional Euclidean space \( \mathbb{R}^n \) to real numbers set \( \mathbb{R} \):

\[
f: x \rightarrow y \text{ where } x \in \mathbb{R}^n, \ y \in \mathbb{R}
\]

This mapping can be viewed as a surface in \( n+1 \)-dimensional Euclidean space: \( n \) dimensions for the argument \( x \) and one for \( y = f(x) \). In 3-dimensional space the surface serves as the graphical representation of a certain mapping from 2-dimensional space to reals, \( y = f(x_1, x_2) \). A sample surface in 3D is depicted in Figure 5. The higher a surface point is situated the higher value it represents.

The concept of legal teleology is substantially more complex than can be visualized with the surface above. However, our intuition attempts the landscape metaphor as the first iteration.

The initial situation can be comprehended as a point on a surface. It represents a state of the \textit{Is world}. The goal situation is a point on the same surface. This is a state of the \textit{Ought world}. The path metaphor fits to lead from the initial situation to the goal. The surface serves to visualize all possible paths from any initial situation to any goal.

The goal entities of the legal sinnlandscape are more complex than the metaphors above. Certain regions of the landscape shall form “holes”. They represent tabu-negative values and shall not be approached. Certain regions form “mountain peaks” which can hardly be reached. They represent tabu-positive values; achieving them requires too expensive means.

A means can be formalized as a vector of factors (multiple criteria). For example, a destination can be reached by a cheap and slow train, \( \text{means}_1 \), or by an expensive but fast train, \( \text{means}_2 \). Here the formalization is a 2-dimensional vector \( \text{means} = (\text{price}, \text{time}) \). In order to weigh \( \text{means}_1 \) and \( \text{means}_2 \), a tradeoff of the criteria – price and time – has to be formalized. Linear models are extensively used in mathematics, especially game theory and mathematical economics. The weight (also called utility, gain) \( w \) of the means can be defined as the linear function of criteria:

\[
w(\text{means}) = \alpha_1 \cdot \text{price} + \alpha_2 \cdot \text{time}
\]

Here both the coefficients \( \alpha_1, \alpha_2 \) and the criteria are real numbers. The means with a greater weight is preferred:

\[3 \ldots \text{the power} \ldots \text{does not reside in the inference method; almost any inference method will do. The power resides in the knowledge” [7, p.101]}\]
means₁ \succ means₂ if and only if w(means₁) > w(means₂), where \( \succ \) denotes the preference relation.

The path metaphor suits with the notion of path in a graph, which is used in discrete mathematics. Suppose a graph \( G = (V, E) \), where \( V \) is the set of vertices and \( E \subseteq V \times V \) is the set of edges. A path \( L \) is defined as a sequence of nodes

\[
L = \langle g_0, g_1, g_2, \ldots, g_M \rangle \quad \text{where} \quad g_i \in V
\]

such that each pair \( e_i = (g_{i-1}, g_i) \) is an edge, i.e. \( e_i \in E \). Thus the path can also be represented as the sequence of edges:

\[
\langle (g_0, g_1), (g_1, g_2), \ldots, (g_{M-1}, g_M) \rangle \quad \text{or} \quad \langle e_1, e_2, \ldots, e_M \rangle
\]

In the path metaphor, each node \( g_i \) plays the role of a subgoal. The goal situation in a sinnlandscape is a certain point on the surface, usually the top of a hill. The path from the initial situation to the goal is a sequence of intermediate subgoals

\[
\langle g_0 = \text{initial\_situation}, g_1, g_2, \ldots, g_M = \text{goal\_situation} \rangle
\]

Each step, i.e. the edge from the subgoal \( g_{i-1} \) to the subsequent \( g_i \), is realized by a certain means \( m_i \). Thus the goal situation is achieved by the sequence of means

\[
\langle m_1, m_2, \ldots, m_M \rangle
\]

Several paths can lead from the initial situation \( g_0 \) to the goal \( g_M \) in the given graph, i.e. in the agreed sinnlandscape. A path \( \text{path}_1 \) is preferred to \( \text{path}_2 \) if its weight is larger (i.e. the cost is smaller). For example, \( \text{path}_1 = \langle \text{up}, \text{up} \rangle \) is preferred to \( \text{path}_2 = \langle \text{up}, \text{down}, \text{up}, \text{up} \rangle \) as the cost \( \kappa(\text{path}_1) = 2 \) is smaller than \( \kappa(\text{path}_2) = 4 \). Here the path cost is calculated as the number of edges. This preference is denoted

\[
\text{path}_1 \succ \text{path}_2 \quad (\text{where cost}(\text{path}_1) < \text{cost}(\text{path}_2), \text{i.e. the cheaper – the better; cost = −weight})
\]

More elaborate formalizations should take into account the cost of each means \( m_i \), which leads from one situation, \( g_{i-1} \), to another, \( g_i \). The cost of an edge \( e_i \) is defined as the cost of this means \( m_i \), i.e. \( \kappa(e_i) = \text{def} \kappa(m_i) \), which realizes the transition. The cost of a path \( L \) is defined as it is generally accepted in graph theory, namely, the sum of edge costs:

\[
\kappa(L) = \sum_{i=1}^{M} \kappa(e_i)
\]

A good path is short and climbs up. It does not lead to deep holes nor does it circle around the goal’s hill like a serpentine.
The cost of a continuous path \( L \) on the landscape \( f \) can be calculated as the thread length from the initial situation \( g_0 \) to the goal \( g_M \). This length is calculated with the mathematical function which is called variation.

Another continuous formalization can be defined as the curvilinear integral of the cost \( \kappa \) on the path \( L \):

\[
\int_L \kappa = \sum_{i=1}^M \kappa(m_i)
\]

This definition follows the intuitive interpretation that the path cost sums up the costs of all the steps.

6 The Landscape Metaphor in Means-Ends Analysis

The landscape metaphor can explain the weight of a means. In this section the proverb “The end justifies the means” (Der Zweck heiligt das Mittel) is analyzed.

Hans Kelsen devotes a whole section [9, p. 13-15] to logical relation between willing an end and willing a means. He starts with the examination of Kant’s imperative “Who is willing the end, must be willing the means” (Wer den Zweck will, muss das Mittel wollen). Kelsen is concerned with teleological and normative necessity. In his logical analysis, both the end and the means are treated as one-dimensional (positive-negative), even Boolean: 1, right (heiliges), or 0, wrong (unheiliges). Such formalization suits well the logical analysis of relations between norm, the act of will (Willensact) and thought operation (Denkoperation).

In this analysis, the means is also evaluated as one-dimensional, even Boolean: true-false (1-0, right-wrong, positive-negative). We extract from “The end justifies the means” the following meaning: only a positive (right) means is justified. The constraint for the means to be positive is essential. In other words, the statement’s idea is to seek a means \( m \), such that (1) brings about the end, and (2) is evaluated positively:

\[
eval(m) = 1 \quad \text{(or } eval(m) > 0 \text{ in continuous mathematics)}
\]  

Though the means war realizes the end peace (Si vis pacem, para bellum), but it is evaluated negatively, \( eval(\text{war}) = 0 \). Therefore war does not satisfy the search criteria. Hence it does not suit to achieve peace.

![Diagram](image_url)

**Figure 6.** A representation of three means, \( m_{\text{wrong}} \), \( m_{\text{weak}} \) and \( m_{\text{right}} \), in the form of two-dimensional vectors. (1) \( m_{\text{wrong}} = (1,0) \) realizes the end, but it is evaluated negatively. (2) \( m_{\text{weak}} = (0,1) \) does not realize the end, though it is evaluated positively. (3) \( m_{\text{right}} = (1,1) \) both realizes the end and is evaluated positively.

As noted earlier, in multiple criteria interpretation the means is treated as a vector (of reals, integers or otherwise ordered values). A simple way to expand one-dimensional Boolean representation is to add the second dimension. To illustrate this, the dimension \( \text{bringsAboutTheEnd} \) is added. This dimension is assigned the value true (1) in the case the means realizes the end; otherwise, it is assigned false (0). For example, a disproportionate means \( m_{\text{wrong}} \) realizes the end (represented \( \text{bringsAboutTheEnd} = 1 \)), but is evaluated negatively (\( \text{evaluation} = 0 \)); see Figure 6:

\[
m_{\text{wrong}} = \langle \text{bringsAboutTheEnd} = 1, \text{evaluation} = 0 \rangle
\]  

A weak means \( m_{\text{weak}} \) does not realize the end (\( \text{bringsAboutTheEnd} = 0 \)), though it is evaluated positively (\( \text{evaluation} = 1 \)); see Figure 6:

\[
m_{\text{weak}} = \langle \text{bringsAboutTheEnd} = 0, \text{evaluation} = 1 \rangle
\]  

A right means \( m_{\text{right}} \) both realizes the end (\( \text{bringsAboutTheEnd} = 1 \)) and is evaluated positively (\( \text{evaluation} = 1 \)); see Figure 6:

\[
m_{\text{right}} = \langle \text{bringsAboutTheEnd} = 1, \text{evaluation} = 1 \rangle
\]
7 Teleological Relations in the Theory of Rudolf von Jhering

This section provides analysis of statements about legal teleology in Rudof von Jhering’s “Law as a Means to an End” [8]. Quotations below are provided according to the English translation.

In the Editorial preface [8, p. xxii], Joseph H. Drake writes about Bentham’s idea that “legislation must be shaped with reference to the greatest good for the greatest number”. Thus Bentham’s conception of the purpose of law can be represented in the notation of 2-elements relation:

\[ \text{law} \leftrightarrow \text{the_greatest_good_for_the_greatest_number} \]

The assertion “law is not an end in itself, but a means” can be inferred from the book’s title. Further in the preface [8, p. xxiii], Drake reminds not to forget “that law is not an end in itself and as such to be brought to a state of formal and static perfection, but that the end is the good of society”.

From the above we extract two assertions. The first is “Law is a means.” It is interpreted as “Law is an instance of the class means.” It can be represented as the relation

\[ \text{law} \leftrightarrow \text{instance-of(means)} \]

The second assertion is the following: “This means teleologically serves to achieve a certain end.” Both assertions are represented as the semantic graph in Figure 7.

![Figure 7. A semantic graph which represents two assertions: (1) “Law is a means” (more precisely, “Law is an instance of the class means”) and (2) “This means teleologically serves a certain end”](image)

Jhering’s Law of Purpose is: no volition, or, which is the same thing, no action, without purpose [8, p. 2]. We visualize the law of purpose in Figure 8. In other words, Jhering’s theory claims that “no rule which does not owe its origin to a purpose” [8, p. xxviii-ix].

![Figure 8. Visualization of Jhering’s law of purpose](image)

Jhering notes that the purpose belongs to the future. Therefore in our notation the arrow leads from the action to the purpose: \( A \leftrightarrow B \), i.e. action \( A \leftrightarrow \) purpose. The term “motive” can be treated as a synonym of purpose [8, p. 3].

Drake refers to the philosophy that knowledge is not wisdom: “Knowledge comes, but wisdom lingers” [8, p. xxviii]. Thus wisdom is rated above knowledge. In this way an ordering or preference is introduced: knowledge \( \prec \) wisdom. This preference is important from the standpoint of knowledge representation, whereas the latter is modeled in computing and AI. Knowledge is subject to representation in computers, whereas law is subject to wisdom. The latter is attributed to human beings and the humanity on the whole.

8 Mixed Systems: Systems with Teleological and Causal Relations

This section concerns supplementing a teleological relation \( A \leftrightarrow B \) with a causal relation \( A \rightarrow B \). Hence, we aim at a notation for normative systems which include two kinds of relations: teleological and causal. Then we analyze different combinations of evaluation of \( A \) and \( B \). It should be noted that evaluation and weighing is within the nature of law. (The author is indebted to F. Lachmayer for the examples below.)

Consider a causal relation \( A \rightarrow B \). Consider that the action \( A \) is evaluated positively, for instance, “wandering”. Hence, generally, the consequence \( B \) has to be evaluated neutrally:

\[ A^{\text{positive}} \rightarrow B \]

But in the derived causal relationship the consequence \( B \) is generally evaluated positively, i.e.

\[ A^{\text{positive}} \rightarrow B^{\text{positive}} \]

- 45 -
Now consider that $A$ is evaluated negatively. Hence, generally, the consequence $B$ has to be evaluated neutrally: $A^{\text{negative}} \rightarrow B$. But in the derived causal relationship $B$ is usually evaluated negatively:

$$A^{\text{negative}} \rightarrow B^{\text{negative}}$$

(19)

A problem of formalization (multiple criteria) arises when the action $A$ is evaluated positively, while its consequence $B$—negatively (the so-called conflicting evaluation):

$$A^{\text{positive}} \rightarrow B^{\text{negative}}$$

(20)

For example, a negative consequence of wandering is wasting time. Other examples are: “give a chocolate”, “reach a toy-arrow from the roof”, etc. The consequences are evaluated negatively because of a side effect, e.g., teeth caries, or high risk to fall down from the roof, etc.

Conflicting evaluations are dominant in our lives. Here we derive a metanorm “not to do $A$”:

$$A^{\text{positive}} \rightarrow B^{\text{negative}} \implies \text{metanorm}\left(\neg A\right)$$

(21)

9 Conclusions

The paper emerges as a reflection on legal teleology. The law can be viewed as a whole. Such a view is required by a cognitive approach to the law. The result is a metaphorical view to legal teleology.

The proposed action-goal model is of declarative nature. It describes one step; another question is whether this leap is a little or big. The path metaphor is of more procedural nature. The reason to mix a continuous and a discrete paradigms is as follows. Open-texture is innate in the legal domain. The fuzziness of the justice concept is innate in the law. Discrete (mathematical) formalizations bring their pro and contra. An advantage is mathematical strictness. However, a disadvantage is a risk to lose the spirit of law.

We think that formalizing semantics can contribute to the open-texture problem in the law. This can be illustrated by Hart’s example of “Vehicles are forbidden in the park” and its analysis [2].

10 Acknowledgements

The author profited greatly from collaboration with Friedrich Lachmayer. The paper resulted after fruitful discussions with him. His visualizations reveal the spirit of law.

The anonymous referees deserve special thanks. They indicated some open issues, in particular, those of mathematical logic, which must be addressed. The author postpones the issues for the future research.

References


EXPERIMENTS WITH LOCAL SEARCH HEURISTICS FOR THE TRAVELING SALESMAN PROBLEM

Alfonsas Misevicius\textsuperscript{1}, Andrius Blazinskas\textsuperscript{2}

\textsuperscript{1}Kaunas University of Technology, Department of Multimedia Engineering, Studentu st. 50–4004416a, LT–51366, Kaunas, Lithuania, alfonsas.misevicius@ktu.lt

\textsuperscript{2}Kaunas University of Technology, Department of Multimedia Engineering, Studentu st. 67–504, LT–51392, Kaunas, Lithuania, andrius.blazinskas@ktu.lt

Abstract. In this paper, the experiments with local search heuristic algorithms for the traveling salesman problem (TSP) are described. Since the ordinary local search heuristics very seldom yield solutions of high quality, we investigate the enhanced local search algorithms using the extended neighbourhood structures. We also examine the performance of the local search heuristics in an iterated local search paradigm based on the deterministic descent-random ascent methodology. Our new heuristic algorithms are tested on the symmetric TSP instances taken from the publicly available electronic library of the TSP instances (TSPLIB). The results from the experiments demonstrate that our heuristics appear to be superior to traditional types of local search algorithms.

Keywords: artificial intelligence, optimization, local search, heuristics, traveling salesman problem.

1 Introduction

The traveling salesman problem (TSP) \cite{2,7} can be formulated as follows. Given the matrix $D = (d_{ij})_{n \times n}$ and the set $\Pi_n$ of all possible permutations of the integers from 1 to $n$, find a permutation $p = (p(1), p(2), \ldots, p(n)) \in \Pi_n$ that minimizes the following function:

$$z(p) = \sum_{i=1}^{n} d_{p(i)p(i+1)} + d_{p(n)p(1)}.$$  \hfill (1)

Each permutation corresponds to a tour through $n$ cities such that each city is visited exactly once. A particular element of the permutation, $j_i = p(i)$ ($i = 1, 2, \ldots, n$), denotes city $j_i$ to visit at step $i$. The pairs $(p(1), p(2)), \ldots, (p(i), p(i+1)), \ldots, (p(n), p(1))$ are called edges. The matrix $D$ contains distances between all the pairs of cities\textsuperscript{1}; the entries $d_{p(i), p(i \text{ mod } n + 1)}$ ($i = 1, 2, \ldots, n$) then denote the corresponding edge length and $z(p)$ is the total length of the tour $p$. So, solving the TSP means searching for the shortest tour so that every city is visited once and only once and the salesman returns back to his starting city at the end of the trip.

The traveling salesman problem is one of the most famous and probably the most intensively studied problems in combinatorial optimization. The TSP and its variants have various important practical applications \cite{13}. At the same time, the TSP is known to be NP-hard and is an excellent experimenting platform for the investigation of both exact (approximation) algorithms and heuristics (metaheuristics) \cite{2,7,10,11,12,19}.

In this paper, we describe the experiments with several local search (LS) heuristics for the TSP. The origins of the LS algorithms go back to \cite{5,15,16}; still, this class of heuristics is a quite active domain of research \cite{3,4,6,8,9,14,18,21}. The LS heuristics are distinguished for their simplicity and ease of implementation; but, sometimes, they also hide some features which can significantly improve the efficiency of the search. In this work, we demonstrate how such improvement is achieved by extending the traditional local search paradigm.

The remaining part of this paper is organized as follows. In Section 1, we outline the general local search framework. The proposed extensions of the basic LS framework are also described. In Section 2, we present the results of the computational experiments with the proposed heuristics on the benchmark problems taken from the TSP library (TSPLIB). Section 3 completes the paper with concluding remarks.

2 Local search heuristics for the traveling salesman problem

2.1 Preliminaries

The Hamming distance between two tours (permutations) $p$ and $p'$ is declared as

$$\rho(p, p') = |\Omega|;$$  \hfill (2)

where $\Omega$ is the set that consists of all possible pairs $(p(i), p(i \text{ mod } n + 1))$ ($i = 1, 2, \ldots, n$) such that $\neg \exists j$:

---

\textsuperscript{1} Only symmetric TSP is considered here, i.e. $d_{ij} = d_{ji}$, $i = 1, 2, \ldots, n$. 
Briefly speaking, the Hamming distance is the number of edges that are contained in one tour, but not the other.

A neighbourhood function \( \Theta : \Pi_n \to 2^{\Pi_n} \) assigns for every \( p \) from \( \Pi_n \) a set \( \Theta(p) \subseteq \Pi_n \) — the set of neighbouring solutions of \( p \) (see Figure 1). An example of the neighbourhood function is the \( \tau \)-edge-exchange neighbourhood \( \Theta_\tau \), which is defined in the following way:

\[
\{ (p'(j), p'(j + 1)), 1 \leq j < n \} \quad \text{or} \quad \{ (p(i), p(i \mod n + 1)) \} = \{ (p'(j), p'(j - 1)), 1 < j \leq n \} \quad \{ (p'(j), p'(n)), j = 1 \}.
\]

(3)

Figure 1. Graphical interpretation of a hypothetical neighbourhood

The tour \( p' \in \Theta(p) \) can be obtained from \( p \) by an operation called a move, and \( p \) is said to move to \( p' \) when such an operation is performed. In case of the 2-edge-exchange neighbourhood, the move may be formally described as an operator \( \phi(p, i, j) : \Pi \times \mathbb{N} \times \mathbb{N} \to \Pi \), which gives \( p' \in \Theta(p) \) such that \( p'(i) = p(i), p'(i + 1) = p(j), p'(j) = p(i + 1), p'(j \mod n + 1) = p(j \mod n + 1) \), where \( 1 \leq i, j \leq n \) and \( 1 < j - i < n - 1 \); in addition, if \( j - i - 2 \geq 1 \), then \( p'(i + k + 1) = p(j - k) \) for every \( k \in \{1, ..., j - i - 2\} \). Roughly speaking, two edges at the positions \( i \) and \( j \) are removed and two different edges are added (see Figure 2).

For the 2-edge-exchange move, a shorter notation of the form \( \phi_{ij} \) may be used, such that \( p' = p \diamond \phi_{ij} \) means that \( p' \) is obtained from \( p \) by applying the move \( \phi(p, i, j) \).

Figure 2. Graphical illustration of the 2-edge-exchange move

The solution (tour) \( p^* \) is 2-optimal (imal) solution, i.e. it is locally optimal with respect to the neighbourhood \( \Theta_2 \) if \( z(p^*) \leq z(p) \) for any \( p \in \Theta_2(p^*) \), i.e. \( z(p^*) \leq z(p^* \diamond \phi_{ij}) \) for every \( i \in \{1, ..., n - 2\}, j \in \{i + 2, ..., n - 1 + \text{sign}(i - 1)\} \).
The 2-opt solution may be achieved by the 2-opt procedure, which can also be viewed as a steepest
descent (SD) algorithm using the neighbourhood $\Theta_2$, i.e. a sequence of the descending (improving) 2-edge-
exchange moves (2-opt moves) starting from an arbitrary initial solution (see Figures 3, 4). (The initial solution
may be generated randomly or constructed in a pre-determined way.) The description of the template of the 2-opt
algorithm (steepest descent in $\Theta_2$) in a high-level pseudo-code form is given in Figure 5. In a similar way, higher
order procedures may be described: 3-opt, 4-opt, ..., r-opt. However, in this case, the run-time complexity
increases exponentially.

In the next section, we propose several extended local search strategies without substantially increasing
computational complexity.

```
function steepest_descent_in_\Theta_2(p);
    // input: p – initial (starting) solution
    // output: p’ – resulting solution (locally optimal solution in $\Theta_2$)
begin
    p’ := p;
    repeat
        p := p’;  \Delta_{\text{min}} := 0;  \//\Delta_{\text{min}} denotes the minimum difference in the objective function values
        for i := 1 to n - 2 do
            for j := i + 2 to n - 1 + \text{sign}(i-1) do begin
                \Delta := z(p \oplus \phi_{ij}) - z(p);
                if \Delta < \Delta_{\text{min}} then begin
                    \Delta_{\text{min}} := \Delta;  k := i;  l := j
                end;  \// for
            end;
        if \Delta_{\text{min}} < 0 then p’ := p \oplus \phi_{kl}  \//move from the current solution to a new one
        until \Delta_{\text{min}} = 0;
    return p’
end.
```

Figure 5. Pseudo-code of the steepest descent algorithm using the neighbourhood $\Theta_2$

2.2 Extensions of the local search

We can construct new more complex neighbourhood structures by allowing a composition of simpler
neighbourhoods (like $\Theta_2$). The extended (compounded) 2-edge-exchange neighbourhoods may be
formally defined as follows. Let $p$ be a feasible tour (permutation) from $\Pi_n$; also, let $\Theta_2(p)$ be the 2-edge-
exchange neighbourhood as defined above in Section 2.1. Then, the extended neighbourhood $\Theta_2$ (denoted as
$\Theta_{2\Theta_2}$) can be described in the following way (also see Figure 6):

$$\Theta_{2\Theta_2}(p) = \Theta_2(p) \cup \{ p^* | p^* \in \Pi_n, p^* \neq p, \rho(p^*, p^*) = 2 \};$$

(4)

where $p^* = \arg\min_{p^* \in \Theta_2(p)} z(p^*)$, $\Theta_2(p) = \Theta_2(p) \setminus \{ \arg\min_{p \in \Theta_2(p)} z(p) \}$.

Figure 6. Graphical interpretation of the neighbourhood $\Theta_{2\Theta_2}$

Further, two new additional neighbourhoods ($\Theta_{2\Theta_2\Theta_2}$, $\Theta_{2\Theta_{2\Theta_2}}$) may be introduced (also see
Figures 7, 8):

$$\Theta_{2\Theta_2\Theta_2}(p) = \Theta_{2\Theta_2}(p) \cup \{ p^{\sim\sim} | p^{\sim\sim} \in \Pi_n, p^{\sim\sim} \neq p, \rho(p^{\sim\sim}, p^{\sim\sim}) = 2 \};$$

(5)

where $p^{\sim\sim} = \arg\min_{p^{\sim\sim} \in \Theta_{2\Theta_2}(p)} z(p^{\sim\sim})$, $\Theta_{2\Theta_2}(p^*) = \Theta_2(p^*) \setminus \{ \arg\min_{p^* \in \Theta_2(p^*)} z(p^*) \}$;

$$\Theta_{2\Theta_{2\Theta_2}}(p) = \Theta_{2\Theta_2}(p) \cup \{ p^{\sim\sim\sim} | p^{\sim\sim\sim} \in \Pi_n, p^{\sim\sim\sim} \neq p, \rho(p^{\sim\sim\sim}, p^{\sim\sim\sim}) = 2 \};$$

(6)
where \( p^* = \operatorname{arg\,min}_{p \in \Theta_2} z(p^*) \), \( \Theta_2 = \Theta_2(p) \setminus \{ \operatorname{arg\,min}_{p \in \Theta_2} z(p^*) \} \).

Similarly, the algorithms using the neighbourhood algorithms is proportional to \( O(n^2) \). The complexity of these algorithms is proportional to \( O(n^2) \), where \( n \) is the problem size. These algorithms may be thought of as ”forward-looking” strategies; they also conceptually resemble the ”depth-first search” and ”breadth-first search” strategies used for searching in tree or graph structures.

The pseudo-code of the steepest descent algorithm in the neighbourhood \( \Theta_{2028} \) is presented in Figure 9. Similarly, the algorithms using the neighbourhoods \( \Theta_{2028} \), \( \Theta_{2028} \) could be described. The complexity of these algorithms is proportional to \( O(n^3) \), where \( n \) is the problem size. These algorithms may be thought of as ”forward-looking” strategies; they also conceptually resemble the ”depth-first search” and ”breadth-first search” strategies used for searching in tree or graph structures.

```plaintext
function steepest_descent_in_\Theta_{2028}(p);
    // input: p - initial (starting) solution
    // output: p' - resulting solution (locally optimal solution in \( \Theta_{2028} \))
    begin
        p' := p;
        repeat
            for i := 1 to n - 2 do
                for j := i + 2 to n - 1 + sign(i - 1) do begin
                    \( \Delta := z(p \ominus \phi_i) - z(p) \);
                    if \( \Delta < \Delta_{\min}^{(2)} \) then begin
                        \( \Delta_{\min}^{(2)} = \Delta_{\min}^{(2)} ; \Delta_{\max}^{(1)} = k^{(1)} ; f^{(1)} = f^{(1)} + 1 \);
                        \( i_j = i_j + 1 \) end else if \( \Delta < \Delta_{\min}^{(1)} \) then begin
                        \( \Delta_{\min}^{(2)} = \Delta_{\max}^{(1)} ; \Delta_{\max}^{(1)} = k^{(1)} ; f^{(1)} = f^{(1)} + 1 \);
                        \( i_j = i_j + 1 \) end
            end;
        until \( z(p') \geq z(p) \);
        return p';
    end.
```

Figure 9. Pseudo-code of the steepest descent algorithm using the neighbourhood \( \Theta_{2028} \)

The other way of extending the classical local search framework is to integrate (combine) the deterministic and stochastic search processes. This policy is known as an iterated local search (ILS) [17]. There are many different variations of the ILS approach depending on the actual deterministic improving algorithm and randomization (perturbation) technique. In this work, we applied the steepest descent procedure using the neighbourhood \( \Theta_{2028} \) in the role of the improving (descending) algorithm; while, for the random perturbations (random ascents), we used a special type of perturbations — the so-called nearest neighbour reconnection (NNR) procedure (see [18]).

\[2\] Note that \( \Theta_{2028}(p) = \Theta_{2028}(p) \).
The resulting algorithm (called the iterated steepest descent-random ascent (ISD-RA)) is not deterministic any more; however, it enables to significantly increase the search efficacy and leads to the superior-quality solutions (see Section 3). The NNR perturbations take $O(n^2)$ time, so the complexity of ISD-RA remains proportional to $O(n^5)$. The high-level pseudo-code of this algorithm is shown in Figure 10. The limit of run time of ISD-RA is predetermined by the number of iterations, $\lambda$, which, in turn, can be flexibly tuned by the user.

\begin{verbatim}
function iterated_steepest_descent_random_ascent(p, \lambda):
  // input: p – initial (starting) solution, \lambda – number of iterations (\lambda \geq 1)
  // output: p’ – resulting solution
  begin
    p’ := steepest_descent(p); // perform steepest descent in a given neighbourhood (starting from p’)
    p’ := p’;
    current_number_of_iterations := 0;
    repeat
      current_number_of_iterations := current_number_of_iterations + 1;
      p’ := random_ascent(p’); // perform random ascending perturbation (starting from p’)
      p’ := steepest_descent(p’); // perform steepest descent in a given neighbourhood (starting from p’)
      if z(p’) < z(p’) then p := p’
    until current_number_of_iterations = \lambda
    return p’
  end.
\end{verbatim}

Figure 10. Pseudo-code of the iterated steepest descent-random ascent algorithm

3 Results of computational experiments

We have tested our local search algorithms on the benchmark problems taken from the publicly available electronic library of the TSP – TSPLIB [20]. The experiments were carried out on a personal computer with an Intel Pentium IV 3 GHz single-core processor.

The following heuristics were used in the experiments: 2-opt; 3-opt; 4-opt; steepest descent in $\Theta_{282}$ (SD-$\Theta_{282}$); steepest descent in $\Theta_{20282}$ (SD-$\Theta_{20282}$); steepest descent in $\Theta_{20282}$ (SD-$\Theta_{20282}$); iterated steepest descent-random ascent (ISD-RA). In order to allow fair comparison, the experimentation was designed as follows. Let $W$ be the pre-defined number of runs. At every run, the algorithm is applied to a given instance, each time starting from a new random initial solution. The current run is interrupted as soon as the local optimum is found (or the maximum number of iterations is performed). The next run is then started, and so on. The process stops when $W$ runs have been carried out. The best solution obtained during these runs serves as a resulting solution of the algorithm. This is repeated for each examined instance. In particular, we used 1 run of 4-opt, 100 runs of 3-opt, and 10000 runs of 2-opt. Similarly, we applied 5000 runs of SD-$\Theta_{20282}$ and 2500 runs of SD-$\Theta_{20282}$ and SD-$\Theta_{282}$. In the last three cases, the number of runs is smaller than in 2-opt; this is due to the fact that SD-$\Theta_{20282}$, SD-$\Theta_{20282}$, and SD-$\Theta_{282}$ consume some more computation (CPU) time. The run time of ISD-RA is controlled by the number of iterations, $\lambda$. We used $\lambda = 2000$. In this way, all the algorithms utilize approximately similar CPU times (note that we experimented with the problems, where the number of cities is more or less equal to 100).

The results of these experiments are presented in Table 1 (additionally, in Figure 11, we graphically illustrate the obtained optimal tours for several TSP instances). Note that, in Table 1, $\delta$ denotes the relative deviation of the solutions from the provably optimal solution; it is defined by the following formula:

$$\delta = 100(\tilde{z} - z^*)/z^* [\%],$$

where $\tilde{z}$ is the obtained value of the objective function (tour length) and $z^*$ denotes the provably optimal objective function value (these values can be found in TSPLIB [20]).

The results from Table 1 show that the new proposed neighbourhoods $\Theta_{282}$, $\Theta_{20282}$, $\Theta_{282}$ are superior to the ordinary 2-edge-exchange neighbourhood $\Theta_2$. Also, it could be seen that the neighbourhood $\Theta_{20282}$ ("breadth-first search") is preferable to the neighbourhood $\Theta_{282}$ ("depth-first search") (this is despite the fact that the sizes of the neighbourhoods $\Theta_{282}$ and $\Theta_{20282}$ are identical). Finally, it is easy to observe that the steepest descent in the neighbourhood $\Theta_{20282}$ combined with the random ascending perturbations is clearly better than all remaining algorithms without random perturbations. It should be stressed that the results of ISD-RA may possibly be improved even more by incorporating more elaborated perturbation procedures.

The other criterion of the efficiency of the algorithms is so-called time-to-target plots [1]. In this case, for any given target value of the objective function (target solution) and the time to obtain this value, the time-to-target plot shows the probability that the target value will be obtained. So, for a given target value, the run time of the algorithm to achieve this value is recorded. This is repeated multiple times and the recorded times are then
sorted. With the ith time, a probability \( p_i = \frac{i-0.5}{m} \) (\( i = 1, 2, \ldots \)) is associated, where m is the number of trials (we used \( m = 30 \)).

Table 1. Results of the experiments with the different local search heuristics on TSPLIB instances

<table>
<thead>
<tr>
<th>Instance name (^\dagger)</th>
<th>( z^0 )</th>
<th>2-opt</th>
<th>3-opt</th>
<th>4-opt</th>
<th>SD-(\Theta_{20202})</th>
<th>SD-(\Theta_{20202})</th>
<th>SD-(\Theta_{20202})</th>
<th>ISD-RA</th>
<th>Average CPU time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>bier127</td>
<td>118282</td>
<td>0.649</td>
<td>0.853</td>
<td>1.596</td>
<td>0.517</td>
<td>0.340</td>
<td>0.331</td>
<td>0.000</td>
<td>3.7</td>
</tr>
<tr>
<td>ch130</td>
<td>6110</td>
<td>0.898</td>
<td>0.977</td>
<td>0.953</td>
<td>0.418</td>
<td>0.401</td>
<td>0.356</td>
<td>0.016</td>
<td>3.9</td>
</tr>
<tr>
<td>ell101</td>
<td>629</td>
<td>1.455</td>
<td>1.934</td>
<td>2.657</td>
<td>1.119</td>
<td>0.996</td>
<td>0.855</td>
<td>0.007</td>
<td>3.1</td>
</tr>
<tr>
<td>gr96</td>
<td>55209</td>
<td>0.446</td>
<td>0.397</td>
<td>0.406</td>
<td>0.321</td>
<td>0.311</td>
<td>0.302</td>
<td>0.000</td>
<td>2.9</td>
</tr>
<tr>
<td>gr120</td>
<td>6942</td>
<td>1.786</td>
<td>1.468</td>
<td>2.579</td>
<td>1.199</td>
<td>0.939</td>
<td>0.843</td>
<td>0.034</td>
<td>3.6</td>
</tr>
<tr>
<td>gr137</td>
<td>69853</td>
<td>0.819</td>
<td>0.888</td>
<td>1.718</td>
<td>0.768</td>
<td>0.779</td>
<td>0.600</td>
<td>0.011</td>
<td>4.5</td>
</tr>
<tr>
<td>kroa100</td>
<td>21282</td>
<td>0.043</td>
<td>0.045</td>
<td>0.108</td>
<td>0.040</td>
<td>0.035</td>
<td>0.035</td>
<td>0.000</td>
<td>3.0</td>
</tr>
<tr>
<td>krob100</td>
<td>22141</td>
<td>0.377</td>
<td>0.575</td>
<td>2.527</td>
<td>0.387</td>
<td>0.352</td>
<td>0.347</td>
<td>0.000</td>
<td>3.0</td>
</tr>
<tr>
<td>kroc100</td>
<td>20749</td>
<td>0.536</td>
<td>0.539</td>
<td>0.530</td>
<td>0.403</td>
<td>0.375</td>
<td>0.335</td>
<td>0.000</td>
<td>3.0</td>
</tr>
<tr>
<td>krod100</td>
<td>21294</td>
<td>1.485</td>
<td>1.548</td>
<td>1.619</td>
<td>1.002</td>
<td>0.940</td>
<td>0.904</td>
<td>0.000</td>
<td>3.0</td>
</tr>
<tr>
<td>kroe100</td>
<td>22068</td>
<td>0.738</td>
<td>1.392</td>
<td>2.035</td>
<td>0.830</td>
<td>0.694</td>
<td>0.657</td>
<td>0.000</td>
<td>3.0</td>
</tr>
<tr>
<td>lin105</td>
<td>14379</td>
<td>0.441</td>
<td>0.437</td>
<td>2.324</td>
<td>0.438</td>
<td>0.416</td>
<td>0.403</td>
<td>0.000</td>
<td>3.2</td>
</tr>
<tr>
<td>pr124</td>
<td>59030</td>
<td>0.088</td>
<td>0.183</td>
<td>1.429</td>
<td>0.095</td>
<td>0.086</td>
<td>0.071</td>
<td>0.010</td>
<td>3.2</td>
</tr>
<tr>
<td>pr124</td>
<td>44303</td>
<td>0.088</td>
<td>0.183</td>
<td>1.429</td>
<td>0.095</td>
<td>0.086</td>
<td>0.071</td>
<td>0.010</td>
<td>3.2</td>
</tr>
<tr>
<td>rat99</td>
<td>1211</td>
<td>0.424</td>
<td>0.401</td>
<td>1.007</td>
<td>0.365</td>
<td>0.329</td>
<td>0.296</td>
<td>0.000</td>
<td>3.0</td>
</tr>
<tr>
<td>rd100</td>
<td>7910</td>
<td>0.921</td>
<td>0.682</td>
<td>0.445</td>
<td>0.537</td>
<td>0.462</td>
<td>0.391</td>
<td>0.009</td>
<td>3.0</td>
</tr>
</tbody>
</table>

\( \dagger \) the numeral in the instance name indicates the size of the problem, i.e. the number of cities.

Figure 11. Illustration of provably optimal TSP tours for the TSPLIB instances

kroa100 (a), krob100 (b), kroc100 (c), krod100 (d), kroe100 (e)

In Figure 12, we present the time-to-target plots for two best algorithms, SD-\(\Theta_{20202}\) and ISD-RA. The instances examined are kroa100 and krob100 and the target values are 21282 and 22141, respectively. These values are exactly equal to the provably optimal objective function values (see TSPLIB [20]).

Figure 12. Probabilities of obtaining target values of 21282 (a) and 22141 (b) versus time for the instances kroa100 (a) and krob100 (b)

The performance improvement factor, PIF, of ISD-RA to SD-\(\Theta_{20202}\) can formally be defined by the formula:

\[
\text{PIF} = \frac{t_{0.5}(\text{SD-}\Theta_{20202})}{t_{0.5}(\text{ISD-RA})}
\]

where \( t_{0.5} \) denotes the time needed to obtain the given target solution with probability 0.5. From Figure 12, it could be viewed that the performance improvement factor of ISD-RA to SD-\(\Theta_{20202}\) is equal to approximately 15.5 (\( \frac{4.65}{0.3} \)) and 11.1 (\( \frac{5.55}{0.5} \)) for the instances kroa100 and krob100b.
respectively. We observed similar performance improvement also for remaining TSPLIB instances used in the experimentation.

4 Concluding remarks

The local search (LS) heuristics remain popular among researchers due to their reasonably good efficiency and easy implementation. In this work, several new modifications of the local search heuristics for solving the traveling salesman problem are proposed. These heuristics are based on the use of the extended neighbourhoods, which enable to seek high quality solutions without significantly increasing computational complexity. The efficiency of the search can be improved even more by integrating these extended heuristics into the iterated local search (ILS) framework. The computational experiments confirm the promising performance of the resulting ILS algorithm (iterated steepest descent-random ascent (ISD-RA)) from both the solution quality and computation time point of view.

Our proposed strategies for extending the neighbourhoods and enhancing the local search heuristics are of quite general character. They appear to be problem-independent, so may be applicable for other combinatorial type optimization problems. Also, there is still a potential for constructing further new neighbourhood structures following the proposed approach. In addition, the improved local search procedures can by used as powerful subroutines within more complex metaheuristic methods (like genetic or evolutionary algorithms). This might be one of the hopeful future research directions.

References

QUADRATIC-ASSIGNMENT-LIKE PROBLEM FORMULATION IN FEATURE SELECTION

Erinija Pranckeviciene¹, Alfonsas Misevicius²,

¹University of Ottawa, Biology Department, 30 Marie Curie, Ottawa ON, Canada, erinija.pranckeviciene@gmail.com
²Kaunas University of Technology, Department of Multimedia Engineering, Studentu 50, Kaunas, Lithuania, alfonsas.misevicius@ktu.lt

Abstract. Quadratic-assignment-like problem formulation in feature selection is proposed. We select a group of m most similar features out of n features by solving quadratic-assignment-like problem using an effective genetic algorithm. As feature similarity metrics the correlation coefficient of individual features is used. The proposed method is validated on NIPS2003 FS benchmark data-sets. The method finds stable feature subsets in which the identities of the discovered features do not vary much. Our procedure can easily be applied in multi-class problems, including other measures of feature similarity.

Keywords: Feature selection, classification, validation, model selection, quadratic assignment problem.

1 Introduction

Data mining is a process of finding the meaningful patterns in the data. Feature selection/extraction is an inherent part of this process [4]. Massive data-sets are accumulated in many domains. Various feature selection techniques are emerging in response to the increasing demand for more efficient ways of data analysis. Feature selection methods are divided into filter, wrapper and embedded [5], [14]. All have certain advantages and limitations. Filter methods are easy to apply, but do not take into account the interactions between features. Wrapper methods produce feature subsets optimal for specific classifier, but suffer from over-fitting when data-dimensionality is high compared to the training sample size. Embedded feature selection integrates feature selection into the classifier, but is limited in classifier choices. Advanced general feature selection strategies such as Floating Forward Feature Selection (FFFS) [13] with monitoring option are time consuming and inefficient in high-dimensional problems. Either in predictive modeling or exploratory data analysis the feature selection procedure is subject to validation. In a single validation step the feature selection method usually produces different feature subsets, due to the adaptation to the peculiarities of the particular training set. Classifier independent feature selection aims at identifying groups of similar features by clustering [10]. In clustering, the methods and formulations of operations research can be used [11]. The problem of selection of a group of m similar objects out of n objects, can be formulated and solved as quadratic assignment problem [2]. In our study we focus on application of quadratic-assignment-like problem (QALP) formulation and solution in feature selection.

1.1 Feature selection algorithm

We aim at the selection of the group of m features out of n features according to some measure of their similarity. We use a correlation between the individual features as their similarity metrics. The selected group comprises most mutually correlated m features. The ith and jth feature are represented by vectors

\[ X_i = [x_{i1}, \ldots, x_{iN_1}, x_{i(N_1+1)}, \ldots, x_{iN_2}] \] and \[ X_j = [x_{j1}, \ldots, x_{jN_1}, x_{j(N_1+1)}, \ldots, x_{jN_2}] \], in which \( N_1 \) and \( N_2 \) denote the number of samples in class 1 and class 2. The feature similarity metrics is defined as:

\[ c_{ij} = \frac{X_i^T X_j}{\|X_i\| \|X_j\|} \] (1)

Motivation for using correlation as feature similarity metrics is, that noisy features are less correlated than the features, relevant for class separation (in general, the number of classes are not restricted to two, and similarity metrics can be arbitrarily chosen). Given n by n matrix of the entries computed by (1), we select the m features, maximizing the objective function:

\[ f(m) = \sum_{i=1}^{m} \sum_{j=1}^{m} c_{ij} \] (2)

For all permutations of \( i \in m \) and \( j \in m \), the objective function (2) is evaluated. The feature group with maximum value of the objective is selected. This problem can be formulated and solved in terms of the quadratic
assignment problem [2]. Many heuristic approaches can be applied for solving QAP. Recently, genetic algorithms are among the advanced heuristic techniques for solving QAP [1],[8]. In particular we use the genetic algorithm crite(alfa) for the solution of (our) feature selection problem. An effective multiple parent crossover [9] is applied in the current genetic algorithm to obtain high quality solutions. Our primary interest lies in the stability and robustness of the feature selection procedure, in which the identities of the selected features would not vary much in cross-validation, and the noisy features would be discarded.

2 Experimental setup

We used datasets of NIPS2003 Feature Selection (NIPS2003 FS) challenge [5], [6] in our study. More about NIPS2003 FS challenge can be found on the challenge website www.nipsfsc.ecs.soton.ac.uk. Two class datasets of the challenge represent real life problems - mass spectra classification (ARCENE dataset), handwritten digit recognition (GISETTE dataset), text classification (DEXTER dataset), drug discovery (DOROTHEA dataset) and the artificial problem (MADELON dataset). The true features in the datasets are augmented with probes- the fake features. Ground truth about the feature identity: useful or probe is available, enabling to test the feature selection methods. All datasets are high-dimensional and represent various levels of difficulty. Training set and Validation set with the class labels, and Test set without the class labels were provided in the benchmark. Performance estimate in NIPS2003 FS challenge is the Balanced Error Rate (BER), computed from the confusion matrix $\text{confmat}$ of the classifier:

$$
\text{confmat} = \begin{pmatrix}
TP & FP \\
FN & TN
\end{pmatrix}
$$

$$
\text{BER} = \frac{1}{2} \left( \frac{TP}{TP + FP} + \frac{TN}{TN + FN} \right)
$$

in which the TP is the true positive, FP is false positive, FN is false negative and TN is true negative. We validated our feature selection procedure in k-folds of the Training set. Simple feature pre-filtering was very effective in building good classifiers on NIPS2003 FS data [6]. In every fold we initially performed a univariate feature filtering using the absolute class difference as the criterion:

$$
a = \left| \sum_{p=1}^{N_1} x_p - \sum_{q=1}^{N_2} x_q \right|
$$

The indices $p$ and $q$ denote the samples of the class 1 and class 2 respectively. $N_1$ and $N_2$ are sample sizes of the classes. The features are ranked by the decreasing value $a$. Low-$a$-value features are discarded, retaining only 255 features, as constrained by the current version of the experimental software for solving quadratic-assignment-like problem (QALP) [9]. Several feature groups of different sizes (40, 60,80, 100, 120) out of the retained features are selected by solving QALP.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>ARCENE</th>
<th>GISETTE</th>
<th>DEXTER</th>
<th>DOROTHEA</th>
<th>MADELON</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dim</td>
<td>10000</td>
<td>5000</td>
<td>20000</td>
<td>100000</td>
<td>500</td>
</tr>
<tr>
<td>Total probes (%)</td>
<td>30</td>
<td>50</td>
<td>50.3</td>
<td>50</td>
<td>96</td>
</tr>
<tr>
<td>Fold Train</td>
<td>35+45</td>
<td>2400+2400</td>
<td>135+135</td>
<td>62+578</td>
<td>800+800</td>
</tr>
<tr>
<td>Fold Test</td>
<td>9+11</td>
<td>600+600</td>
<td>15+15</td>
<td>16+144</td>
<td>200+200</td>
</tr>
<tr>
<td>Validation</td>
<td>44+56</td>
<td>500+500</td>
<td>150+150</td>
<td>34+316</td>
<td>300+300</td>
</tr>
<tr>
<td>Test</td>
<td>310+390</td>
<td>3250+3250</td>
<td>1000+1000</td>
<td>78+722</td>
<td>600+600</td>
</tr>
<tr>
<td>Features (%)</td>
<td>2.6</td>
<td>5.1</td>
<td>1.3</td>
<td>0.255</td>
<td>51</td>
</tr>
<tr>
<td>Probes (%)</td>
<td>0.2</td>
<td>0</td>
<td>12</td>
<td>10.8</td>
<td>93.2</td>
</tr>
<tr>
<td># Folds k</td>
<td>5</td>
<td>5</td>
<td>10</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

In the following we refer to the selected subsets as QALP features. For MADELON we tested the feature groups of smaller sizes (5:2:25). The stability and robustness to noise were measured by the average percentage of common feature identities and probes, selected in k-fold cross-validation. In every fold, the selected feature groups were used to train several state-of-the-art classifiers in PRTools [3]. A performance of a combination QALP features + classifier was assessed on the Fold Test set. The performance estimates determine the final model number of QALP features + classifier to train on all available data. The non-linear and
linear classification rules were used: nearest neighbor 3 and 5 (NN3 and NN5) classifiers, linear support vector machine (SVC), subspace classifier (SUBSC) and logistic linear classifier (LOGLC). The number of folds k, data dimensionality Dim, total percentage of probes in the original dataset, number of samples in both classes $N_1 + N_2$ in Fold Train, Fold Test, independent Validation and Test sets, percentages of the retained features and probes in the filtered feature set are summarized in Table 1.

3 Results

Due to different training samples in cross-validation, the different feature subsets are selected. In order to assess the stability and robustness of the QALP features, we calculated a percentage of the same feature identities, occurring more than 0.5*k times in k-fold cross-validation. This was applied for each subset size. Larger percentages indicate more stable procedure. The average percentage of probes, selected for each subset size, shows the robustness of the procedure. Our results are presented in Table 2.

Table 2. Stability and robustness of QALP feature selection.

<table>
<thead>
<tr>
<th>Subset size</th>
<th>Dataset</th>
<th>40</th>
<th>60</th>
<th>80</th>
<th>100</th>
<th>120</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>common feature identity (%) / probes (%) in k-fold cross-validation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARCENE</td>
<td>47/0</td>
<td>71/0</td>
<td>68/0</td>
<td>37/0</td>
<td>29/0</td>
<td></td>
</tr>
<tr>
<td>GISETTE</td>
<td>76/0</td>
<td>92/0</td>
<td>95/0</td>
<td>94/0</td>
<td>90/0</td>
<td></td>
</tr>
<tr>
<td>DEXTER</td>
<td>87/16</td>
<td>70/21</td>
<td>70/17</td>
<td>86/17</td>
<td>80/19</td>
<td></td>
</tr>
<tr>
<td>DOROTHEA</td>
<td>54/0</td>
<td>44/0</td>
<td>51/0</td>
<td>48/0</td>
<td>57/0</td>
<td></td>
</tr>
<tr>
<td>Subset size</td>
<td>5</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>MADELON %</td>
<td>56/0</td>
<td>64/0</td>
<td>69/0</td>
<td>69/19</td>
<td>36/78</td>
<td></td>
</tr>
</tbody>
</table>

The most stable QALP features were for GISETTE and DEXTER datasets. In the second place were MADELON and DOROTHEA datasets. Feature selection stability in ARCENE decreased with the increase of the feature subset size. This can be explained by the relatively small sample size of Fold Train. The feature subsets of ARCENE, GISETTE, DOROTHEA do not contain probes. GISETTE was probe-free after filtering. More probes were included into MADELON feature subsets as subset size increased. This was an expected result, because MADELON had only 20 useful features. Largest percentages of probes appeared in the DEXTER dataset. We explain this as the correlation metrics not suitable for DEXTER features. DEXTER is text classification problem, where the features represent frequencies of occurrence of word stems in the text. The

![Figure 1. ARCENE. Distribution of the performances of 3 nearest neighbor (NN3), linear support vector machine (SVC) and subspace (SUBSC) classifiers in 5-fold cross-validation of QALP features. The high variation in the BER estimates in the top-panel is due to small sizes of Fold Test and independent Validation sets. NN3 classifier performs better than other classifiers in subsets of size 100, as can be seen in the bottom-panel.](image)

stable and probe-free feature subsets were obtained for the 3 datasets. In real life problems we do not have information about the feature identity. In order to find out indirectly whether the feature subset contains noisy features, we refer to the estimates of classification performance. If the feature subset is noise-free, then we expect better classification performance of the classifiers, trained with this subset. Several methods, which were highly ranked by the independent Test Balanced Error Rate in NIPS2003 FS benchmark, used feature subsets
with probes. Feature selection and classifier learning are intrinsically combined. The same classification rule performs differently with the different feature subsets and the data samples. If the test set size is small, then the performance estimate has high variance. It gives the over-optimistic or over-pessimistic assessment and the selection of the classification model may be biased. However, the systematic cross-validation leads to the informed choices. QALP features are classifier-independent, therefore the performances of several classification models QALP features + classifier would reveal the proper classification models. The performances of the different classification models in the ARCENE, DOROTHEA and MADELON datasets are summarized in Figures 1, 2 and 3. The top-panels of the Figures show the distribution of the BER of every classification model, estimated in the $k$-th-fold. The x-axis and the y-axis represent the Fold Test BER and the independent Validation BER respectively. The bottom-panels show the dependency between the QALP feature’s subset size and the Fold Test BER. The classifiers are denoted by the different symbols and colors.

Figure 2. DOROTHEA. Distribution of the performances of linear support vector machine (SVC), subspace (SUBSC) and linear logistic (LOGLC) classifiers in 5-fold cross-validation of QALP features. The BER estimates of SUBSC classifier is much less than other classifiers for all subset sizes.

Figure 3. MADELON. Distribution of the performances of nearest neighbor 5 and 3 classifiers (NN3 and NN5), subspace (SUBSC) and linear logistic (LOGLC) classifiers in 5-fold cross-validation of QALP feature subsets. Obvious linear relationship between Fold Test and Validation BER is observed in the top-panel. The BER estimates of the nearest neighbor classifiers are very different from the other two for the features subset sizes larger than 5.

The estimates of the Fold Test BER and independent Validation BER are linearly related. This relationship allows to infer which classifier is superior. Different classifiers have different performance on the same data, depending of how well the classifier matches the complexity of the data, represented by the feature subset of the certain size. Figure 1 shows that for ARCENE the NN3 classifier with feature subsets of size 80 and 100, is superior. A high variation of the BER in subsets of size 120 (bottom-panel) indicates that the larger subsets share less common features identities (confirmed also by Table 2). For ARCENE the recommended classification model would be 100 QALP features + NN3. It is known, that ARCENE dataset has a nonlinear character [5], thus, the nonlinear rule performed better. For DOROTHEA the subspace classifier is the best choice (see Figure 2). The relatively small variation of the BER in the subsets of the size 120 suggests,
that 120 is an optimal feature subset size for DOROTHEA out of the tested feature set sizes. The recommended classification model would be **120 QALP features + SUBSC**. The bottom-panel of Figure 3 shows that the subsets containing only 5 features do not provide sufficient information about the class separation. All classifiers perform poorly. The improvement begins with the subsets of size 11. Performance degrades gradually as the subset size increases. In MADELON more probes occur in the larger subsets and degrades the performance. NN3 and NN5 are superior classifiers for this dataset. The BER of the NN3 classifier is less variable, which justifies the **11 QALP features + NN3** model for the MADELON. Similar analysis suggests the **100 QALP features + NN5** classification model for the GISETTE and **120 QALP features + SVC** for the DEXTER datasets. Classifier independent QALP feature selection was compared to the embedded feature selection method Liknon in the context of NIPS2003 FS benchmark. Liknon, as a “black box” classification model, was characterized by a good performance in Agnostic Learning versus Prior Knowledge challenge [7], [12]. Liknon generates a feature profile optimal for linear separation. Liknon feature profile was used as input to the NN3 and SUBSC classifiers in our study. We compared our results and those of the other probe-free methods in the benchmark. The Test BER of the QALP and Liknon based classification models, the Test BER of the best probe-free method, median and mean BER of all probe-free methods in the benchmark are presented in Table 3. The number of probe-free methods is given in the last column. The QALP and Liknon feature subsets and used classifiers are summarized in Table 4.

### Table 3. Comparison of the performances in NIPS2003 FS benchmark in terms of the Test BER.

<table>
<thead>
<tr>
<th>NIPS2003 FS Entries</th>
<th>QALP features</th>
<th>Best no probes</th>
<th>Liknon embedded</th>
<th>Median no probes</th>
<th>Mean no probes</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARCENE</td>
<td>0.2199</td>
<td>0.0720</td>
<td>&amp; 0.1711</td>
<td>&amp; 0.2017</td>
<td>0.2260</td>
<td>60</td>
</tr>
<tr>
<td>GISETTE</td>
<td>0.0532</td>
<td>0.0086</td>
<td>0.0402</td>
<td>0.0258</td>
<td>0.0724</td>
<td>105</td>
</tr>
<tr>
<td>DEXTER</td>
<td>0.1740</td>
<td>0.0565</td>
<td>0.0820</td>
<td>0.0870</td>
<td>0.1537</td>
<td>29</td>
</tr>
<tr>
<td>DOROTHEA</td>
<td>0.1628</td>
<td>0.1065</td>
<td>0.1996</td>
<td>0.1989</td>
<td>0.2106</td>
<td>77</td>
</tr>
<tr>
<td>MADELON</td>
<td>0.1250</td>
<td>0.0622</td>
<td>0.1133</td>
<td>0.0892</td>
<td>0.1340</td>
<td>172</td>
</tr>
</tbody>
</table>

The worst performance of QALP features was in DEXTER, the best was in DOROTHEA. In general the QALP features performed worse than Liknon features, but better than average, except for DOROTHEA. Table 4 compares the subsets of features. Neither Liknon and QALP features are probe-free in ARCENE and MADELON. Liknon's 41-feature subset is considerably better than QALP 100-feature subset. The two subsets share only 1 common feature identity. In MADELON, the difference between the probe-free 11-feature subsets comprises 5 features, leading to very different performances as well. This suggests that the feature identities are as important for improved performances, as are the sizes of the subsets and the fraction of noisy features in them. QALP features, coupled with the subspace classifier, proved to be the most effective for DOROTHEA. DOROTHEA is highly unbalanced dataset with binary attributes.

### Table 4. Comparison of QALP and Liknon feature subsets and classification models.

<table>
<thead>
<tr>
<th>Method</th>
<th>Subset size / # probes (classifier)</th>
<th>QALP</th>
<th>Liknon</th>
<th>Common features</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARCENE</td>
<td>100 / 0 (NN3)</td>
<td>41 / 0 (texttt{NN3})</td>
<td>1 / 0</td>
<td></td>
</tr>
<tr>
<td>GISETTE</td>
<td>100 / 0 (NN5)</td>
<td>56 / 12 (texttt{NN3})</td>
<td>17 / 0</td>
<td></td>
</tr>
<tr>
<td>DEXTER</td>
<td>120 / 17 (texttt{SVC})</td>
<td>158 / 15 (texttt{SUBSC})</td>
<td>&amp; 37 / 8</td>
<td></td>
</tr>
<tr>
<td>DOROTHEA</td>
<td>120 / 0 (texttt{SUBSC})</td>
<td>7 / 0 (texttt{SUBSC})</td>
<td>3 / 0</td>
<td></td>
</tr>
<tr>
<td>MADELON</td>
<td>11 / 0 (texttt{NN3})</td>
<td>11 / 0 (texttt{NN3})</td>
<td>6 / 0</td>
<td></td>
</tr>
</tbody>
</table>

### 4 Conclusions

We have studied a novel classifier-independent feature selection, using the quadratic assignment like problem formulation. We solved our problem by means of genetic algorithm. The stability and robustness of the feature selection procedure were tested using NIPS2003 FS benchmark datasets in k-fold cross-validation. The QALP feature selection was preceded by the univariate filtering of features, which proved to be a successful strategy. The performances of QALP features, coupled with several classifiers, favorably compared to the average performance of the probe-free benchmark methods. The main advantages of QALP feature selection procedure are simplicity of use and the capability of generating stable and probe-free feature subsets. We used the two class classification problems and the correlation metrics to estimate the feature similarity. This approach can easily be applied in multi-class problems, including other measures of feature similarity, which is subject of
our future work. QALP feature selection using correlation as feature similarity metrics coupled with the subspace classifier was successful for very unbalanced DOROTHEA dataset. The further study of this phenomenon will be useful for solving highly unbalanced classification problems. We believe that our approach of classifier-independent feature and classification model selection will become a useful tool for the practitioners.

References

TIME-EFFICIENT NURBS CURVE EVALUATION ALGORITHMS

Kestutis Jankauskas

Kaunas University of Technology, Department of Multimedia Engineering, Studentu st. 50, LT-51368 Kaunas, Lithuania, kestutis.jankauskas@ktu.lt

Abstract: This paper analyses time-efficiency of existing NURBS evaluation algorithms. The most competitive computation methods are modified to achieve even better performance. Performance tests indicate that NURBS curve evaluation time-efficiency can be improved in uniform and non-rational B-spline cases. Suggested optimizations are very effective in the evaluation of higher degree splines with a larger number of control points.

Keywords: NURBS, curve evaluation, inverted triangular scheme

1 Introduction

NURBS stands for Non-Uniform Rational B-Spline. It is the most popular spline representation in today’s commercial CAD packages [1, 4, 5, 8, 10]. NURBS is able to represent large variety of shapes, like circles, hyperbolas, parabolas, and still preserves mathematical exactness [5].

Generally, a spline is a smooth curve interpolated among given control points. Unfortunately, a spline cannot be constructed in the model space directly. Each point on NURBS curve or surface must be calculated from the set of control points, knot vector, and basis function of specific degree. This process is called NURBS evaluation [1, 5, 8].

In the following sections we will discuss theoretical aspects of NURBS as well as existing evaluation algorithms. Moreover, we will introduce certain modified evaluation algorithms and strategies for uniform and non-rational cases that improve evaluation performance. Finally we will compare actual time-efficiency of suggested method implementations.

2 NURBS in Theory

Acronym NURBS defines special properties of this particular spline: (1) Non-Uniform, (2) Rational, (3) B-Spline. Let us clarify those properties by starting from the last one. It has the basis function of B-spline, which ensures smooth blending [2] of control point influence over the curve. All theory regarding a regular B-spline is covered in Section 2.1. Rational property gives more flexibility to a spline [4, 5, 10], but also increases complexity. It is achieved by adding weights to control points. Rational B-spline is presented in Section 2.2. Finally, introducing the editable knot vector to the spline concept, allows usage of non-uniform piece-wise features [1, 10]. They are covered in Section 2.3.

2.1 B-Spline

Regular B-spline is defined by a set of control points \(P_i\), a knot vector \(U = \{u_j\}\), and degree \(p\), where \(i = 0..n-1\), \(j = 0..m\) and \(m = n + p + 1\) [1, 4, 6, 9]. Control points are located in the multi-dimensional space we refer to as the model space. The spline interpolates between control points with the help of the basis function:

\[
C(u) = \sum_{i=0}^{lcn} N_{i,p}(u)P_i,
\]

where \(p\) is the degree of the basis function, \(u\) is a coordinate in the parametric spline space and \(C(u)\) is point on curve in the model space. Accordingly, any point on the curve is obtained by summing multiplications of control points \(P_i\) and basis functions \(N_{i,p}(u)\). The basis function is calculated from the expression:

\[
N_{i,0}(u) = \begin{cases} 
1 & \text{if } u_i \leq u < u_{i+1} \\
0 & \text{otherwise}
\end{cases} 
\]

\[
N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u),
\]

where \(u\) is the coordinate in the parametric spline space and \(u_j\) are values from the knot vector \(U\). The last expression is referred to as Cox-de Boor recursion formula [1, 3]. It denotes that basis function domains are divided by elements of the knot vector, i.e. knots [1, 9]. It is also known that the sum of all basis functions \(N_{i,p}(u)\) equals one [7]. The sum of \(N_{i,p}(u)\) in the interval \(k-p \leq i \leq k\) equals one as well, where \(u_k \leq u < u_{k+1}\) (the partition of unity [2, 6]):
\[
\sum_{i=k-p}^{i=k} N_{i,p}(u) = 1. \tag{4}
\]

As the basis function is non-negative, it means that all basis functions \( N_{i,p}(u) \) outside the interval \( k - p \leq i \leq k \) are zero. Consequently, all multiplications \( N_{i,p}(u) P_i \) outside the same interval are zero. Thus, such control points have no effect on the portion of the curve within \( u_k \leq u < u_{k+1} \). So any point on the curve \( C(u) \) of the uniform B-spline is affected by \( p + 1 \) control points, with the exception of \( C(0) \) and \( C(1) \). These special cases can be explained through analysis of the knot vector.

The knot vector is a set of non-decreasing values \( u_j \leq u_{j+1} \), where \( j = 0, 1, \ldots, m \). In this paper we use a normalized knot vector form, so the parametric space of the B-spline and knot vector values are bounded by 0 and 1. Also, it is a common practice to use the clamped knot vector, where the first \( p + 1 \) values equal 0 and the last \( p + 1 \) values equal 1 [1, 3, 10]:

\[
U = \{u_0 = u_1 = \ldots = u_p = 0 \leq u_{p+1} \leq u_{p+2} \leq \ldots \leq u_{n-1} \leq u_n = \ldots = u_{n+p} = u_{n+p+1} = 1\}. \tag{5}
\]

Let us examine the example of a cubic B-spline (\( p = 3 \)), defined by eight control points (\( n = 8 \)) and \( m = 12 \) uniform knots: \( U = \{0, 0, 0, 0, 0.2, 0.4, 0.6, 0.8, 1, 1, 1, 1\} \). Basis functions are given in Figure 2. For \( u = 0.3 \) functions \( N_{1,3}(0.3), N_{2,3}(0.3), N_{3,3}(0.3), \) and \( N_{4,3}(0.3) \) are greater than zero, other functions are zero. This means that a point on the curve \( C(0.3) \) is affected by the position of four control points: \( P_1, P_2, P_3, \) and \( P_4 \). Also function \( N_{2,3}(0.3) \) and \( N_{3,3}(0.3) \) values are significantly greater than values of \( N_{1,3}(0.3) \) and \( N_{4,3}(0.3) \). This suggests that the point \( C(0.3) \) is closer to \( P_2 \) and \( P_3 \) than to \( P_1 \) and \( P_4 \).

In the case when \( u \) is in the position of the knot \( u = 0.4 \) we have only three non-zero functions: \( N_{2,3}(0.3) \neq 0, N_{3,3}(0.3) \neq 0, \) and \( N_{4,3}(0.3) \neq 0 \). Consequently \( C(0.4) \) is affected by three control points: \( P_2, P_3, \) and \( P_4 \). So, a point on the curve \( C(u) \) is affected by \( p - s + 1 \) control points, where \( s \) is knot multiplicity at \( u \). Therefore the curve becomes \( C^{p-s} \) continuous at this point (here the symbol \( C \) refers to spline continuity and has a different meaning than \( C(u) \), see Section 2.3 for more details) [1, 9].

Because of the clamped knot vector the first curve point \( C(0) \) is affected only by one control point \( P_0 \). The last curve point \( C(1) \) is affected by the control point \( P_{n-1} \) respectively:

\[
C(0) = P_0, \tag{6}
\]
\[
C(1) = P_{n-1}. \tag{7}
\]

2.2 Rational B-Spline

Regular B-spline is quite powerful interpolation tool, but it lacks flexibility. B-spline can not represent conic sections, like circles [4, 5, 9]. Therefore a rational form is used to cover these cases [4, 10]:

\[
C(u) = \frac{1}{\sum_{i=0}^{i=n} w_i N_{i,p}(u) P_i}, \tag{8}
\]

where the weight \( w_i > 0 \) is attached to every control point.

Figure 1. A circle represented by four rational B-splines
Let us take a look at the example of a circle represented as four rational B-splines in Figure 1. Each quarter of the circle is constructed from separate rational quadratic B-spline defined by control point sequences: \( \{P_0, P_1, P_2\} \), \( \{P_2, P_3, P_4\} \), \( \{P_4, P_5, P_6\} \), \( \{P_6, P_7, P_0\} \). The weights of the first and the last control points in each sequence are 1. The weight of the middle control point is \( \frac{\sqrt{2}}{2} \) \[4\]. Greater weights pull the curve towards the control point and lesser weights push the curve away \[1, 4, 10\]. Naturally, regular B-spline is a special case of rational B-spline when all weights are equal to 1.

### 2.3 Non-uniform Rational B-Spline

The term of uniformity is used to define a relation between the sequence of control points and the parametric spline space. As mentioned in Section 2.1, control point influence over the curve is defined by basis functions and function domains are divided by knots \[9\]. This means that property of uniformity is embedded into the knot vector \[1\]. Until now we considered a knot vector to be clamped and uniform:

\[
U = \{u_0 = \ldots = u_p = 0, \ldots, u_i = \frac{i - p}{n - p}, \ldots, u_n = \ldots = u_{n + p + 1} = 1\},
\]

where \( p + 1 \leq i < n \). Such a knot sequence divides whole parametric space into uniform intervals. Each of intervals contains \( p + 1 \) non-zero basis functions, thus the curve is affected by \( p + 1 \) control points in this interval (see Section 2.1). In general case, knots can be distributed in non-uniform manner. However a knot sequence must be non-decreasing, as shown in the expression (5).

Let us take the example of the knot vector \( U = \{0, 0, 0, 0, 0.2, 0.4, 0.6, 0.8, 1, 1, 1, 1\} \) and modify it by setting \( \{u_6 = u_5 = u_4 = 0.2\} : \quad U = \{0, 0, 0, 0, 0.2, 0.2, 0.2, 0.8, 1, 1, 1, 1\} \). Knot multiplicity of \( s = p \) at \( u = 0.2 \) leaves only one non-zero function at this point (see Figure 3), which suggest that \( C(0.2) \) is affected by single control point. Therefore, the curve goes through this control point: \( C(0.2) = P_5 \). In other words, the knot of multiplicity \( s \) reduces curve continuity at that knot by \( s \) \[3, 10\]. In this example the curve becomes \( C^{P-s} = C^0 \) continuous at \( u = 0.2 \). Further increment of multiplicity is pointless, because it excludes control points from affecting the curve.

NURBS is powerful enough to compose any shape. Recall the example of the circle in Figure 1. It was represented by four uniform rational B-splines. Knot multiplication in the knot vector allows the construction of such a shape from single quadratic NURBS curve. The same control points with weights are used in the sequence \( \{P_0, P_1, P_3, P_4, P_5, P_6, P_7, P_0\} \). The last control point is the same as the first to close the curve. Instead of multiple control points, multiple knots are employed: \( U = \{0, 0, 0, 0.25, 0.25, 0.5, 0.5, 0.75, 0.75, 1, 1, 1\} \) \[4\]. Behavior of basis functions is depicted in Figure 4. Notice that every quarter of the circle is represented by single non-zero knot interval and each quarter is independent.

![Figure 2. Basis functions of uniform cubic B-spline defined by eight control points](image)

![Figure 3. Basis functions of cubic B-spline with knot multiplicity of three at 0.2](image)
Figure 4. Basis functions of quadratic NURBS defined by the knot vector \( U = \{0, 0, 0, 1/4, 1/4, 2/4, 2/4, 3/4, 3/4, 1, 1, 1\} \)

3 NURBS evaluation algorithms

To represent NURBS in the model space (Cartesian multi-dimensional space) as a curve, the spline must be evaluated at multiple \( u \), where \( 0 \leq u \leq 1 \). According to the expression (8) basis functions are necessary in order to do so. Several basis function calculation methods are covered in Section 3.1. Once basis functions are known, they can be used to determine a point on the curve. The description of single point evaluation algorithms can be found in Section 3.2. Finally, entire NURBS curve evaluation strategies are presented in Section 3.3.

3.1 Basis function

As we already discussed in Section 2.1, calculation of all basis functions is not necessary. There are only \( p - s + 1 \) non-zero basis functions at any \( u \), where \( p \) is the degree of the basis function and \( s \) is knot multiplicity at \( u \). So we are to obtain all basis functions from \( N_{k-p:p}(u) \) to \( N_{k:p}(u) \), where \( u_k \leq u < u_{k+1} \).

3.1.1 Cox-de Boor recursion

The most obvious solution is to use a standard Cox-de Boor recursion formula, given in the expression (2) and (3). Although this formula is simple to understand and easy to implement, [6] and [9] sources state that it involves many unnecessary calculations. Figure 5 illustrates how \( N_{k:p}(u) \) is obtained.

Figure 5. Computation of non-zero basis functions

Zero functions are marked in blue. They have no effect on higher degree functions in successive iterations, because multiplication by zero is zero (blue arrows). In the example of \( U = \{0, 0, 0, 0.2, 0.4, 0.6, 0.8, 1, 1, 1\} \), where \( p = 3 \) and \( k = 4 \) (\( u \) is in the interval \( u_4 \leq u < u_5 \)), the recursive formula returns non-zero values of \( N_{1:3}(u) \), \( N_{2:3}(u) \), \( N_{3:3}(u) \), and \( N_{4:3}(u) \). Accordingly to the expression (3), to obtain \( N_{1:3}(u) \) the algorithm calculates \( N_{0:2}(u) \) and \( N_{1:2}(u) \). To acquire second degree functions, the recursion must obtain first degree functions \( N_{1:1}(u) \) and \( N_{2:1}(u) \). Finally, first degree functions is calculated from zero degree functions: \( N_{1:1}(u) \) is acquired from \( N_{0:1}(u) \) and \( N_{1:0}(u) \), \( N_{2:1}(u) \) is acquired from \( N_{2:0}(u) \) and \( N_{3:0}(u) \), \( N_{3:1}(u) \) is acquired from \( N_{3:0}(u) \) and \( N_{4:0}(u) \). Notice that \( N_{2:1}(u) \) is calculated twice, so \( N_{2:0}(u) \) as well as \( N_{3:0}(u) \) is actually calculated three times. Moreover, only \( N_{4:0}(u) \) is non-zero among all zero degree functions.

This example illustrates how Cox-de Boor recursion formula is overloaded with unnecessary calculations. Naturally, the evaluation of higher degree B-spline basis functions yields even more unnecessary iterations. Also the expression (3) is numerically unstable, because of \( 0/0 \) cases [5]. Another drawback is noted
in [2]. The recursion formula gives an incorrect result when \( u = 1 \). The last point on the curve is always \( C(1) = [0, 0, 0] \). To overcome this problem, we simply use expressions (6) and (7) as special cases, so \( C(0) \) and \( C(1) \) can be found without the calculation of basis functions.

### 3.1.2 Inverted Triangular Scheme

To avoid unnecessary calculations, authors in [6] present the algorithm based ITS (inverted triangular scheme). It is given as Basis_ITS0 function in pseudo code. It calculates functions from lower to higher degree in contrast to the recursive algorithm. Also, they suggest rearrangement of the expression (3) to remove operation duplications:

\[
N_{k-j-p}(u) = \frac{L_{j+1}}{R_j + L_{j+1}} N_{k-j-p-1}(u) + \frac{R_{j+1}}{R_{j+1} + L_j} N_{k-j-p-1}(u),
\]

where \( L_j = u - u_{k+j} \) and \( R_j = u_{k+j} - u \).

#### Basis_ITS0(k, p, u)

1. \( N[0] = 1 \)
2. \( \text{for } (j = 1; j <= p; j++) \)
   1. saved = 0
   1.1. \( L[j] = u - \text{knots}[k + 1 - j] \)
   1.2. \( R[j] = \text{knots}[k + j] - u \)
   1.3. \( \text{for } (r = 0; r < j; r++) \)
      1.3.1. \( \text{tmp} = N[r] / (R[r + 1] + L[j - r]) \)
      1.3.2. \( N[r] = \text{saved} + R[r + 1] * \text{tmp} \)
      1.3.3. \( \text{saved} = L[j - r] * \text{tmp} \)
3. \( N[j] = \text{saved} \)
4. return \( N \)

Note that \( k \) should already be known, where \( k \) defines the knot interval in which \( u \) resides. Therefore, the method FindKnotSpan (available in [6]) must be applied to determine \( k \) before the implementation of Basis_ITS0.

#### Modified Inverted Triangular Scheme

We noticed another relation. Let the right part of the sum in \( N_{i,p}(u) \) be equal \( A \), then the left part of the sum in \( N_{i,p}(u) \) is always \( A - 1 \). Based on this observation, we propose another modification of the expression (3):

\[
N_{i,p}(u) = A_i(u) \cdot N_{i,p-1}(u) + (1 - A_{i+1}) \cdot N_{i+1,p-1}(u),
\]

where \( A_{i,p}(u) = (u - u_i)/(u_{i+1} - u_i) \) and \( k - p \leq i \leq k \).

The example of non-zero cubic basis function calculation is given in Table 1, followed by modified ITS algorithms. As \( u \) value is fixed we omit the notation of \( (u) \).

| Table 1. Non-zero basis function calculations for cubic B-spline, using a modified ITS |
|---------------------|---------------------|---------------------|---------------------|---------------------|
| \( i \)         | \( p = 0 \) | \( p = 1 \) | \( p = 2 \) | \( p = 3 \) |
| \( k - 3 \) | \( N_{k-3,1} = 1 \) | \( N_{k-3,2} = N_{k-3,1} \) | \( N_{k-3,3} = (1 - A_{k-3,2}) N_{k-3,2} \) | |
| \( k - 2 \) | \( N_{k-2,1} = (1 - A_{k-2,1}) N_{k-2,0} \) | \( N_{k-2,2} = A_{k-2,1} N_{k-2,1} + (1 - A_{k-2,2}) N_{k-2,2} \) | \( N_{k-2,3} = A_{k-2,2} N_{k-2,2} + (1 - A_{k-2,3}) N_{k-2,3} \) | |
| \( k - 1 \) | \( N_{k-1,0} = 1 \) | \( N_{k-1,1} = A_{k-1,0} N_{k-1,0} \) | \( N_{k-1,2} = A_{k-1,2} N_{k-1,2} + (1 - A_{k-1,3}) N_{k-1,3} \) | \( N_{k-1,3} = A_{k-1,3} N_{k-1,3} + (1 - A_{k-1,4}) N_{k-1,4} \) |
| \( k \) | \( N_{k,0} = 1 \) | \( N_{k,1} = A_{k,0} N_{k,0} \) | \( N_{k,2} = A_{k,2} N_{k,2} + (1 - A_{k,3}) N_{k,3} \) | \( N_{k,3} = A_{k,3} N_{k,3} \) |

#### Basis_ITS0(k, p, u)

1. \( N[0] = 1 \)
2. \( \text{for } (i = 1; i <= p; i++) \)
   1. \( A = (u - \text{knots}[k]) / (\text{knots}[k + 1] - \text{knots}[k]) \)
   1.1. \( \text{for } (j = i - 1; j >= 0; j--) \)
      1.1.1. \( \text{tmp} = N[j] * A \)
   1.2. \( \text{for } (j = i - 1; j >= 0; j--) \)
      1.2.1. \( \text{tmp} = N[j + 1] + N[j] - \text{tmp} \)
      1.2.2. \( N[j] = \text{tmp} \)
3. return \( N \)

#### Basis_ITSU(k, p, u)

1. \( N[0] = 1 \)
2. \( M = (u - \text{knots}[k]) / (\text{knots}[k+1] - \text{knots}[k]) \)
3. \( \text{for } (i = 1; i <= p; i++) \)
   1. \( \text{for } (j = i - 1; j >= 0; j--) \)
      1.1. \( \text{tmp} = N[j] * M \)
      1.1.1. \( \text{tmp} = N[j] * (M + j)/i \)
      1.1.2. \( N[j + 1] = N[j] - \text{tmp} \)
      1.1.3. \( N[j] = \text{tmp} \)
3. return \( N \)

These algorithms return basis functions in reversed order: from \( N_{k,p}(u) \) to \( N_{k-p,p}(u) \). Basis_ITS0 and Basis_ITS1 algorithms are suitable for any NURBS. Only few CAD and CAM applications allow editing
the knot vector, because such modification is not intuitive [10]. Hence, in many cases NURBS stays uniform. From the expression (9) it is obvious that every non-zero interval in the knot vector equals \( 1/(n-p) \). Let us presume that \( M = A_{k,1} = (u-u_k)/(a_{k+1}-u_k) \). It is easy to calculate that \( A_{k,p} = M/p \), \( A_{k-1,p} = (M+1)/p \), \( A_{k-2,p} = (M+2)/p \). So, in case of the uniform knot vector, the expression (13) can be simplified:

\[
A_{k-j,p} = \frac{M+j}{p}.
\]

(14)

Plugging the expression (13) into the last row of Table 1 indicates that calculation of a non-zero function set uses knots from \( u_{k+p+1} \) to \( u_{k+p} \). So the equation (14) is valid when all knot intervals from \( u_{k+p+1} \) to \( u_{k+p} \) are equal. In the case of the clamped knot vector, the first \( p \) and last \( p \) knot intervals are zero. As the first uniform interval begins at \( u_p \) and the last uniform interval ends at \( u_n \), the expression (14) can be used for all intervals from \( u_{p+p-1} \) to \( u_{n-p} \). This means that the ITS algorithm can be written as Basis_ITSU for all \( u_k \leq u < u_{k+1} \), where:

\[
2p-1 \leq k \leq n-p.
\]

(15)

### 3.2 Single point on curve

Each of non-zero functions defines how strongly a certain control point affects a curve (see Section 2.1). According to the expression (8), the strength of the effect is also modified by weights of control points (see Section 2.2). In order to calculate \( C(u) \), we require a sum of all \( N_{i,p}(u)w_i \) divided by the sum of \( N_{i,p}(u)w_i \), where \( k-p \leq i \leq k \) and \( u_k \leq u < u_{k+1} \). Following algorithms calculate a point on the curve, when basis functions are known. Thus GetPoint0 should be used after Basis_ITS0. Because of the inverted function order in Basis_ITS1 and in Basis_ITSU, those algorithms should be followed by GetPoint1.

#### 3.2.1 De Boor’s algorithm

There are several B-spline evaluation techniques that do not need basis functions to determine a point on the curve, like de Boor’s algorithm [9]. De Boor’s algorithm is based on observation that \( C(u) \) is positioned at the location of the control point \( P_{k-p} \), when \( u = u_k \) and knot multiplicity at \( u \) equals \( p \) (see section 2.3). How do we make desired knot multiplicity at any \( u \)? The author in [9] suggests a multiple insertion of a knot at \( u \). The insertion of an additional knot also means the insertion of a new control point, thus after \( p \) iterations the last control point is exactly at the position of \( C(u) \). In case when \( u \) is already at the position of the knot \( u_k \), with multiplicity \( s \), only \( p-s \) iterations of the insertion are required. The position of every new control point can be found from expressions [3, 9]:

\[
Q_i^w = (1-a_i)P_i^w + a_iP_{i+1}^w,
\]

(16)

where

\[
a_i = \frac{u-u_i}{u_{i+p} - u_i} \quad \text{for all } k-p+1 \leq i \leq k.
\]

(17)

However, the actual insertion of knots is not performed, because this would lead to the modification of the control point sequence during the evaluation. Thus the sequence of new control points is processed in a
GetPoint_DeBoor(k, u)
1. s = 0
2. while (k >= s && knots[k - s] == u)
3. Q = new ControlPoint[p - s + 1]
4. for (i = k - p; i <= k - s; i++)
5. for (r = 1; r <= p - s; r++)
6. return Q[p-s].ConvertTo3D().To3D()

3.3 Multiple Points on Curve
Generally, evaluation of multiple points can be done using single point evaluation several times. But several optimizations can be made. To evaluate entire NURBS curve, we must obtain multiple points C(u), where \( u = 0, \Delta u, 2\Delta u, \ldots, (steps-1) \Delta u = 1 \) is the step in the parametric spline space. Under these conditions the initial knot interval is \( u_0 \leq u \leq u_{p+1} \), thus initial \( k = p \). Successive \( k \) values can be traced easily, so the procedure FindKnotSpan in not needed. Also \( u = 0 \) and \( u = 1 \) are handled as special cases (see Section 3.1) and calculated from expressions (6) and (7). The following algorithm evaluates the number of points equal to \( steps \) on any NURBS curve.

NURBS_ITS0(steps)
1. step = 1 / (steps - 1)
2. Cu = new Point[steps]
3. Cu[0] = P[0].To3D()
4. iter = 1
5. u = knots[p] + step
6. for (k = p; k < n; k++)
   6.1. while (knots[k] == knots[k + 1] && knots[k] < 1)
   6.1.1. k++
   6.2. while (u < knots[k + 1])
   6.2.1. N = Basis_ITS0(k, p, u)
   6.2.2. Cu[iter] = GetPoint0(N, k)
   6.2.3. iter++
   6.2.4. u += step
7. C[steps - 1] = P[n - 1].To3D()
8. return Cu

Algorithms in steps 6.2.1 and 6.2.2 can be replaced by modified Basis_ITS1 and GetPoint1 respectively. If a spline is known to be non-rational then GetPoint_NR1 can be used in step 6.2.2. If a spline is uniform it is possible to optimize this algorithm even further.

3.3.1 Evaluation of Uniform B-spline Curve
Recall Section 3.1.3 and expressions (15), which states that Basis_ITSU can be used instead of Basis_ITS1 within bounds of \( 2p - 1 \leq k \leq n - p \). Figure 6 illustrates the basis functions of the cubic uniform B-spline defined by \( m = 17 \) knots. Notice that \( N_{6,1}(0) = N_{12,3}(1) \), \( N_{13,0}(0.05) = N_{11,3}(0.95) \), \( N_{2,3}(0.1) = N_{10,3}(0.9) \) and so on. Clearly, certain basis functions of the uniform B-spline are symmetrical to each other. Actually, any function \( N_{i,p}(u) \) can be reflected to \( N_{n-i-1,p}(1-u) \) at the middle point of the parametric space. We refer to this operation as to \( ref \) :

\[
ref : N_{i,p}(u) \rightarrow N_{n-i-1,p}(1-u),
\]

where \( k - p \leq i \leq k \).
The set of non-zero functions \( N_{k-p,p}(u) \ldots N_{k,p}(u) \) at \( 0 \leq u_k \leq u < u_{k+1} < 0.5 \) can be cloned to \( N_{n-k-1,p}(1-u) \ldots N_{n-k+p-1,p}(1-u) \). In other words, there is no need to calculate non-zero functions for the second half of the parametric space, because they can be obtained from the first one.

Figure 6 also depicts another important property of uniform B-spline. Pay attention to functions marked as red, they are identical: \( N_{3,3}(0.2) = N_{3,3}(0.3) = N_{5,3}(0.4) = \ldots = N_{9,3}(0.8) \). The set of non-zero functions at \( u = 0.22 \) consists of four functions: \( N_{2,3}(0.22), N_{3,3}(0.22), N_{4,3}(0.22), \) and \( N_{5,3}(0.22) \). There is a set of functions with the same values at each interval \( u_k \), where \( 5 \leq k \leq 10: N_{2,3}(0.22) = N_{3,3}(0.32) = \ldots = N_{7,3}(0.72), N_{3,3}(0.22) = N_{4,3}(0.32) = \ldots = N_{8,3}(0.72), N_{4,3}(0.22) = N_{5,3}(0.32) = \ldots = N_{9,3}(0.72), \) and \( N_{5,3}(0.22) = N_{6,3}(0.32) = \ldots = N_{10,3}(0.72) \). Obviously, non-zero functions at arbitrary \( u_{2p-1} \leq u < u_{2p} \) can be repeated at \( u + j/(n-p) \), where \( 1 \leq j \leq (n-p)-(2p-1) \). In this paper we refer to this operation as to \( rep: \)

\[
\text{rep: } N_{i,p}(u) \rightarrow N_{i+j,p}(u + j/(n-p)),
\]

where \( 1 \leq j \leq n-3p+1 \) for all \( p-1 \leq i < 2p \).

However, \( u \) values must be distributed in specific manner, in order to hit a required \( 1-u \) or \( u + j/(n-p) \). This means that the chosen step \( \Delta u \) must divide each non-zero knot interval into the same number of equal subintervals. If we consider that \( \kappa \in N \) is a natural number, then step \( \Delta u \) must satisfy:

\[
\Delta u = \frac{1}{(n-p)\kappa}.
\]

### 3.3.2 Multiple Point on Curve Evaluation Strategies

We suggest several NURBS evaluations strategies regarding given observations in Table 2. The strategy name corresponds to the case of the spline and to the basis function algorithm. The interval row indicates the bounds of the parametric space for basis clone operations that are given in the last table row (see expressions \((18), (19), (20), \) and \((21))\).

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Case</th>
<th>Interval</th>
<th>Basis method</th>
<th>Get point method</th>
<th>Basis clone operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>NURBS_ITS0</td>
<td>General</td>
<td>( 0 \leq u \leq 1 )</td>
<td>Basis_ITS0</td>
<td>GetPoint0</td>
<td></td>
</tr>
<tr>
<td>NURBS_ITS1</td>
<td>General</td>
<td>( 0 \leq u \leq 1 )</td>
<td>Basis_ITS1</td>
<td>GetPoint1</td>
<td></td>
</tr>
<tr>
<td>URBS_ITS1</td>
<td>Uniform</td>
<td>( 0 \leq u \leq 0.5 )</td>
<td>Basis_ITS1</td>
<td>GetPoint1</td>
<td>( N_{n-i-1,p}(1-u) = \text{ref}(N_{i,p}(u)) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( u = 0.5 )</td>
<td>Basis_ITS1</td>
<td>GetPoint1</td>
<td></td>
</tr>
<tr>
<td>URBS_ITS1+U</td>
<td>Uniform</td>
<td>( n \geq 3p-1 )</td>
<td>Basis_ITS1</td>
<td>GetPoint1</td>
<td>( N_{n-i-1,p}(1-u) = \text{ref}(N_{i,p}(u)) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( u_{2p-1} \leq u &lt; u_{2p} )</td>
<td>Basis_ITSU</td>
<td>GetPoint1</td>
<td>( N_{i+j,p}(u+j/(n-p)) = \text{rep}(N_{i,p}(u)) )</td>
</tr>
<tr>
<td>UBS_ITS1+U</td>
<td>Uniform</td>
<td>( n \geq 3p-1 )</td>
<td>Basis_ITS1</td>
<td>GetPoint_NR1</td>
<td>( N_{n-i-1,p}(1-u) = \text{ref}(N_{i,p}(u)) )</td>
</tr>
<tr>
<td></td>
<td>Non-rational</td>
<td>( u_{2p-1} \leq u &lt; u_{2p} )</td>
<td>Basis_ITSU</td>
<td>GetPoint_NR1</td>
<td>( N_{i+j,p}(u+j/(n-p)) = \text{rep}(N_{i,p}(u)) )</td>
</tr>
</tbody>
</table>

Note that in order to apply uniform case optimizations, the step size \( \Delta u \) must be set accordingly to the expression \((22)\) before evaluation. An uniform non-rational spline is evaluated using the simplified method \texttt{GetPoint\_NR1} instead of \texttt{GetPoint1} (expression \((1)\) instead of \((8)\)).
4 Results

Algorithms given in Section 3 were implemented using C# programming language and .NET framework. The performance tests were acquired on Intel Core2 Duo 1.86 GHz x2 CPU, 3.0 GB RAM machine. The evaluation of single point or function takes only few nanoseconds. This makes the comparison of evaluation time-effectiveness hardly possible. Therefore all evaluation algorithms were applied $10^5$ times at different $u$. This procedure was performed several times and average calculation times were recorded.

Recursive Cox-de Boor, Basis_ITS0, Basis_ITS1, and Basis_ITSU basis function evaluation algorithms were tested on the uniform 27 control point B-spline. The same algorithms and de Boor’s knot insertion method were employed to determine a single point on the curve. Calculation times are given in Figure 7 and Figure 8 respectively.

![Figure 7. NURBS basis function calculation times in milliseconds](image)

<table>
<thead>
<tr>
<th>Degree</th>
<th>Recursive</th>
<th>ITS0</th>
<th>ITS1</th>
<th>ITSU</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>206</td>
<td>251</td>
<td>206</td>
<td>187</td>
</tr>
<tr>
<td>2</td>
<td>231</td>
<td>247</td>
<td>233</td>
<td>225</td>
</tr>
<tr>
<td>3</td>
<td>393</td>
<td>447</td>
<td>370</td>
<td>310</td>
</tr>
<tr>
<td>4</td>
<td>1128</td>
<td>320</td>
<td>311</td>
<td>279</td>
</tr>
<tr>
<td>5</td>
<td>4927</td>
<td>429</td>
<td>431</td>
<td>360</td>
</tr>
<tr>
<td>6</td>
<td>10976</td>
<td>481</td>
<td>496</td>
<td>402</td>
</tr>
<tr>
<td>7</td>
<td>25855</td>
<td>543</td>
<td>574</td>
<td>460</td>
</tr>
</tbody>
</table>

Figure 8. NURBS single point on curve evaluation times in milliseconds

Calculation time of the recursive Cox-de Boor algorithm grows rapidly for every successive degree of B-spline. However, the ITS is noticeably less affected by the degree increment. De Boor’s knot insertion should be fast, because it has no basis function calculation phase. According to Figure 8, GetPoint_DeBoor overtakes recursive algorithm only when $p \geq 4$, but is left far behind by the ITS. This happens because of a large number of scalar multiplications and conversions from 3D to 4D and back.

The ITS takes less time in the basis function determination phase than in the position acquisition phase even when $p = 8$. Therefore performances of single point evaluation using Basis_ITS0, Basis_ITS1 or Basis_ITSU are very similar. Due to poor performance of recursive Cox-de Boor and de Boor’s knot insertion algorithms they were not included in multiple point evaluation. The evaluation of multiple points over entire 27 control point NURBS curve was carried out using strategies given in Section 3.3.2. Results are given in Figure 9.

The same strategies were applied to $n = 18$ control point and $n = 9$ control point curves. Performance patterns remain the same as in Figure 9, but URBS_ITS1+U and UBS_ITS1+U provided less time economy. In these cases, less control points mean fewer intervals where the operation rep can be applied (see expressions (20) and (21)).
The implementation of ref and rep operations in evaluation of uniform rational B-spline saved from 3.7% to 24.6% of calculation time (compare URBS_ITS1+U and NURBS_ITS1). The algorithm designed for uniform non-rational B-spline saved from 12.4% to 32.8% of calculation time. Also, URBS_ITS1+U and UBS_ITS1+U were respectively up to 18.7% and 27.3% more time-efficient in comparison to NURBS_ITS0.

The evaluation of higher degree basis functions takes longer. In these cases ref and rep operations can save more time (compare URBS_ITS1+U and NURBS_ITS1 in Figure 9). There is one more fact to be taken into consideration. The percentage of saved calculation time depends on the number of NURBS control points. Accordingly to the expression (21), there are \( n - 3p + 1 \) knot intervals where rep operation can be applied. If \( n < 3p - 1 \), this optimization can not be implemented even if B-spline is uniform.

## 5 Conclusions

In this paper we analyzed three already known NURBS evaluation algorithms. Test results showed that the recursive Cox-de Boor formula is highly ineffective especially in the evaluation of higher degree splines. Although de Boor’s knot insertion method performed better while evaluating splines of the fourth and higher degree, it was significantly overtaken by inverted triangular scheme in all cases.

Due to this discovery we composed several modifications of the inverted triangular scheme and few evaluation strategies designed for special cases of NURBS. Accordingly to the test results, the presented strategies saved up to 24.6% of evaluation time in the case of uniform B-spline, and up to 32.8% in the case of uniform non-rational B-spline. A significant gain of performance was observed during NURBS evaluation of the degree \( p > 4 \) with the number of control points greater or equal to \( 3p - 1 \).

The stated facts lead to a conclusion that time-efficiency of NURBS curve evaluation based on the inverted triangular scheme can be improved. This is achieved by recognizing uniform and non-rational cases and implementing evaluation strategies presented in this paper. Optimizations are especially effective in the evaluation of higher degree splines with a larger number of control points.

## References


A DATA MINING APPROACH FOR THE ANALYSIS OF “STOCK-TOUTING” SPAM EMAILS

Mohamed Zaki, Babis Theodoulidis, David Diaz Solis

University of Manchester, Manchester Business School, Manchester M13 9SS, United Kingdom, Mohamed.Zaki@postgrad.mbs.ac.uk, B.Theodoulidis@mbs.ac.uk, David.DiazSolis@postgrad.mbs.ac.uk

Abstract. Although the financial markets are regulated by robust systems and rules that control their efficiency and try to protect investors from various manipulation schemes, markets still suffer from many attempts to mislead or misinform investors in order to gain illegal profits. The impetus to effectively and systematically address such schemes is presenting many challenges to academia, industry and relevant authorities. This paper proposes the use of data mining techniques to detect abuses in the stock market and in particular, Information-based manipulations. These are manipulations that allow rogue traders to gain illegal profits from disseminating false or vague information to investors through spam emails. The paper proposes a spam fraud analysis and detection framework using data mining techniques that helps analysts to identify possible touting cases based on spam emails. The framework employs different techniques such as classification, neural networks and linear regressions. The application of the framework is demonstrated using data from the Pink Sheets market and the results strongly suggest that data mining techniques can be used to facilitate fraud investigations originating from spam emails. The proposed framework and findings of the paper could be used in a retroactive mode to help the relevant authorities and organisations to identify abnormal behaviors in the stock market. It could also be used in a proactive mode to warn analysts and stockbrokers of possible cases of market abuse.

Keywords: Business Intelligence, Data Mining, Fraud Detection, Spam Emails.

1 Introduction

Financial markets continue to suffer from attempts to mislead or misinform investors to impact the decision making process to gain illegal profits despite the increased regulations and control systems. The impetus to effectively and systematically address such schemes presents a very important challenge to academia, industry and relevant authorities.

The rapid development and penetration of technology and communications in our society has strongly affected the flow of information within the stock market. In fact, spam e-mails have been used extensively and creatively to target specific stocks in order for fraudsters to gain illegal benefits. Most commonly, spammers state that they have ascertained private information about stocks. The e-mails contain fine print messages claiming valuable information such as investment advice, stimulation about specific investment decision disclosed with financial terms and recent price quotes. Thus, stock spammers speculated on positive price models of the traded stocks and sent thousands of e-mails to possible investors to tempt investors, driving the price up/down upon the touted spam.

Earlier work mentions that over 80% of all e-mail traffic is classified as spam e-mails, with 15% of these messages related to stock touts [3]. Various studies have manually classified and categorized spam e-mails to evaluate the effect of spam volume tout on the market efficiency. They concluded that touted spam functionally works in the stock market and there are recipients who act upon it [1][3].

Fraud detection systems play an important role in investigating, and detecting fraudulent behaviours and ascertain possible evidence of various financial frauds within different financial markets [7]. Furthermore, these systems could support financial organizations to proactively detect market abuse (MA) transactions, thus minimizing the circulation of rumors or illegal information [2].

This paper introduces a data mining approach for the analysis of “stock-touting” spam emails by proposing a number of financial indicators to use for modelling the problem and applying a data mining approach to analyse the problem. Different algorithms are examined and their performance is compared in terms of their classification accuracy.

2 Previous work

Many research areas have delivered the spam e-mails as one of the main problems that need further investigations. Such as, Boehme and Holz [1] found that there is evidence of harmful effect of the spam messages on the financial market. They employed a multiplicative multivariate regression model and classical event study methodology, focusing on effects of spam e-mails on the return and volume of the target stocks. They found some indications of an increase in trading activity of the advertised stock. Moreover, there were an evidence of abnormal returns occurred after the messages have been disseminated. Stock prices respond
positively to spam e-mails, with a positive relationship between the amount of spam mails per day and the size of returns [5]. Lastly, they linked this kind of manipulation in prices and volumes to three groups of individuals (spammers, Naïve recipient, Smart recipient).

Frieder and Zittrain [3] evaluated and analyzed the impact of touted spam on the trading activity of specific stocks using a big sample from the Pink Sheets market. They compared the touted stock with another control sample (not touted) during the same period [5]. Evidence was found of a significant positive return on days that have a heavy spam touting. Furthermore, the volume of trading was responding positively to the heavy spam touting. For example, the stock that was touted through spam, it was probably the most active stock in trading operations. It was jumping from 4% to 70% daily return on a day that has touting activity. However, the returns of the following days are negative; which is firmly confirming the assumption that spammers “buy low and spam high”.

Hanke and Hauser [5] started their analysis by describing the common characteristics of advertised stock such as price level and average turnover. In spite of investigating the effect of touting spam on returns and volume only, they measured the effect of spam email on other variables like excess returns, turnover and intra-day volatility. They proved that there is a significant (positively) impact on the securities prices when the touting occur. Furthermore, they found that liquidity is the one of the foremost factor in the spamming campaign success. Lastly, the repeated spam on consecutive days continued to rise up the demand on the targeted stock which strength the spammer position and enlarge their time window for liquidation. This paper has some limitations such as, the unavailability of intra-day data and bid-ask quotes. These would give the research the ability to evaluate the potential profits from trading strategies.

All the aforementioned studies proofed that touted spam is functionally working within the stock market and there are recipients read it and also act upon it. However, the analysis and evaluation of the previous research only used the normal financial, statistical equations and models to proof their assumptions. At the same time, the amount of data in the stock market has been increased exponentially during the last ten years, and it has reached unprecedented levels of data generation, scarcely compared with any other fields. This paper will continue the effort that have been done by the previous research by finding the appropriate data mining techniques to analyze and detect fraudulent activities of the spam scenario.

Through the previous literature there are different data mining techniques that have been utilized to detect fraudulent activities in different domains such as telecommunications stock exchanges, credit-card and insurance companies, banks and securities firms. Such as, Donoho [2] utilized some data mining techniques to detect the early insider trading manipulation scheme before news breaks within the option market. He studied two cases for two companies released some news which indicated negative trading behaviour and mislead investor decision. As well as his research compared the results of the algorithms that have been used in his experiment like C4.5, decision tree, neural network, K-Mean clustering, and logistic regressions. Furthermore, the research concludes that the early detection of insider trading can avoid the market and investors from potential losses.

Kirkland [6] described how the National Association of Securities Dealers (NASD) Inc. used a fraud detection system called Advanced Detection System (ADS) to monitor trades and quotations in the NASDAQ Stock Market. The main objective of the system as explained was to detect and identify any suspected trading behaviour for further investigation. Furthermore, raising the level of surveillance from issue based to firm based patterns and practices. The system used many artificial intelligence (AI) techniques like visualization, data mining techniques (association rules, decision trees), time sequence pattern, and pattern recognition to cover the market surveillance, trading violation and help the regulatory parties to protect the market from any breaches. Furthermore, the system is keeping up to date with modern patterns and new data mining algorithms to adapt with any changes in the market behaviour or any new trading violation.

Goldberg [4] described the Securities Observation, News, Analysis and Regulation system (SONAR) that developed by NASD. The main objective of the system is monitoring the NASDAQ transactions in the stock market to detect and identify any potential insider trading and any falsification of fraud news. The system uses many techniques like data mining, text mining, statistical regression, fuzzy matching, and rule based inference that are used to mine various news wires stories and U.S. Securities and Exchange Commission (SEC) filings. Furthermore, the system evaluates price and volume models for various securities within the market, and generates flag alerts daily for further investigation that could support the prosecution of those who are involved in the manipulation. [2] Commented that SONAR system is a good approach for the regulatory purposes, a reliable detector of “inside trading” before the news break.

The analysis introduced in this paper is based on specific sample data that has been collected and analyzed by [3]. They used a specific touted stock which is listed on the Pink Sheets quotation system and a sample of spam e-mails. Their initial dataset consisted of a database of 1,802,016 unsorted spam messages, most of which were downloaded from the internet usenet newsgroup “news.admin.net-abuse.sightings (NANAS)”. They extracted the stock tout messages by selecting the ones that contain the word ‘stock’ and a symbol ticker.
This extraction process filtered 75,415 messages with 28,803 different stock symbols between 22/08/00 and 02/08/05. In addition, it filtered down 3,669 symbol date groups with 500 distinct stocks that were touted.

This study utilizes the following web site which contains the raw data compiled by [3] “http://cyber.law.harvard.edu/stockspam”. The paper uses the China World Trade Corporation “CWTD” stock as a case study to detect the highest possibilities of fraudulent activities. All spam messages that have the symbol “CWTD” (the target sample stock in this research) received on 13/02/04 were placed into a cluster; and representative messages were sampled to check if the symbol reflected an actual tout or not.

3 Spam Manipulation Indicators

Before performing the analysis, the main symptoms of spam manipulation in the stock market are analysed. This allows to build a suitable set of financial indicators to detect the fraudulent behaviors.

The primary symptom is a jump in the volume of traded stocks, which normally indicates good news for investors. However, it may make the stock prices fall sharply after artificial stock pumping. Figure 1 shows a visual analysis of the average daily trading volume of “CWTD”. In this representation, normal days present smooth trends with dark blue dots. The dot size is reflected by the number of touting. Small dark blue dots are expected on a normal trading day. In contrast, larger dots represented by different colours indicate a significant increase in volume and touting. This can be associated with high probabilities of trading violations and possible pump and dump manipulations. The most clear suspicious behaviour is seen in the days prior to 03/01/2004 in which the “CWTD” stock was highly touted. This significantly affected the trading volume which reached 1,400,000 transactions (red dots).

Another symptom is the jump in the prices of the touted stocks. The quotation transactions available for this study include the closing prices which represent the last price at the closing time. Unfortunately, this paper does not have access to the intra-day price transactions of “CWTD”. Figure 2 represents the average daily trading prices of “CWTD”. Normal movements are expressed using dark blue dots. Any variation in the size of the dots is linked to the number of touting occurring in that trading day. It is clearly seen that small dark blue dots are indicative of a normal trading day. Larger and lighter coloured dots indicate a significant increase in prices and volume. In the absence of any new information in the market, related to the CWTD company, in the form of financial news, or fillings sent to the authorities, these jumps could be related with high probability to fraudulent activities. For example, there is a clear suspicious behavioural transaction around the 02/01/2004 for the “CWTD”. This date shows the stock was highly touted and therefore significantly affecting the stock price and the trading volume as well.
Imbalance between the intraday bid and ask number of offers could be another symptom that could be used as an indicator for detecting fraudulent activities. Stock prices should always be affected by any information swirl in the market which could make stock prices fluctuate. If the amounts of bidding offers are greater than the asking number of offers then this indicates good news. However, more asking offers than bidding offers may signify bad news. If these imbalances are not driven by news, either good or bad, then this could indicate the presence of some form of manipulation. Due to this fact, spammers could try to deceive people by stating that they have some kind of insider information and that they should trade and act upon it. Unfortunately, this study only has access to the last bid and ask price of the day. In order to overcome this limitation we build an indicator based on the average of the bid and ask price. This will be further explained in the following sections.

The mentioned data is publicly available in the authors website, from which the whole stock touts table, was downloaded and then filtered down to match the selected tout stock “CWTD” case. Basically, there are 6 main fields representing the symbol ticker of the touted stock, trading date, bid, ask, price, and volume. This table presents the trading transactions of “CWTD” from 09/03/02 to 09/08/05. A second table was also downloaded to extract the number of tout messages which occurred in this stock. This study begins to prepare the data for the analysis by combining the two tables and adding the field “touted” for the sample period.

There are five main indicators that were used in this paper to analyze the dataset. These indicators are based on the symptoms that are described previously and could be classified into three categories: jump in prices, jumps in volume and bid and ask behaviour.

Category I Price Indicators

The following are indicators that build on the jump of prices symptoms.

- **Indicator PInd_1 (Average Price Indicator):** it is the ratio of the change in prices for a day to the moving average (MA) of the change of prices of the last 20 days. The formula is based on the principle that daily changes in prices should lay in the vicinity of the 20 days moving average in order to consider them as “normal behaviour”.

  *Note:* Twenty days are used for the calculation of the moving average as a reference to the number of trading days in a month.

  \[
PInd\_1\_t = \frac{\Delta price_t}{MA(20)},
\]

  where,

  \[
  \Delta price_t = \frac{P_t - P_{t-1}}{P_{t-1}}
  \]

  \[
  MA(20) = \frac{\sum_{i=1}^{t-18} \Delta price_i}{20}
  \]

- **Indicator PInd_2 (Std. Deviation Price Indicator):** it is the ratio of the change in prices for a day to the moving standard deviation changes of price of the last 20 days. The formula is based on the principle that daily changes in prices should lie in the vicinity of the 20 days moving standard deviation in order to measure the relative price. It shows how many standard deviations above average (type of z-score).

  \[
PInd\_2\_t = \frac{\Delta price_t}{\sigma(20)\_t},
\]

  \[
  \sigma(20)\_t = \sqrt{\frac{\sum_{i=1}^{t-18} (\Delta price_i - MA(20)\_i)^2}{20}}
  \]
Category II Volume Indicators

The following are indicators that are built on the jump of volume symptoms.

- **Indicator VInd_1 (Average Volume Indicator):** it is the ratio of the volumes for a day to the moving average (MA) of the volumes for the last 20 days. The formula for calculation is based on the principle that daily changes in volumes should lay in the vicinity of the 20 days moving average in order to consider them as “normal behaviour”.

\[
VInd_{-1} = \frac{\Delta volume_t}{MA(20)_t}
\]

where,

\[
\Delta volume_t = \frac{V_t - V_{t-1}}{V_{t-1}}
\]

\[
MA(20)_t = \frac{\sum_{t=1}^{18} \Delta volume_t}{20}
\]

With \(V_t\) the total volume of day \(t\), and with \(t = 1, 2..., n=425\) in the range 02-01-2004 to 08-09-2005.

- **Indicator VInd_2 (Std. Deviation Volume Indicator):** it is the ratio of the volumes for a day to the moving standard deviation volumes of the last 20 days. The formula for calculation is based on the principle that daily changes in volumes should lay in the vicinity of the 20 days moving standard deviation in order to measure the relative volumes.

\[
VInd_{-2} = \frac{\Delta volume_t}{\sigma(20)_t}
\]

\[
\sigma(20)_t = \sqrt{\frac{\sum_{t=1}^{18} (\Delta volume_t - MA(20)_t)^2}{20}}
\]

Category III Bids and Asks behaviour Indicators

The following is the indicator that builds on imbalance between bids/asks symptoms.

- **Indicator BAInd_1 (Bids/Asks Indicator):** It is the average of buying and selling pressures implicit in the bids and asks prices. To build the indicator first, it is necessary to signal a day’s close as a “buy” or “sell” day in terms of how close was the closing price to the bid or the ask closing offers. It is argued that if the closing price of the day is close to the ask price, the buying pressure was predominant that day, because buyers choose to pay this ask price according to their expectations of prices going up in the future. In contrast, if the closing price is close to the bid price, then the selling pressure was predominant that day, because sellers choose to sell at this price according to their expectations of lower prices in the future. On average it could be expected that buying and selling pressures compensates each other, making the price fluctuate around his fundamental value. If the buy or sell pressure is persistent without any fundamental reason or news, this could indicate the presence of a manipulation in the market.

First to signal a day as a “buy” or “sell” day, a variable is constructed as following:

\[
BA\_var1_t = P_t - A(bids_t, asks_t)
\]

where,

\[
A(bids_t, asks_t) = \frac{bids_t - asks_t}{2}
\]
correspond to the columns that correspond to the built indicators that were previously defined. For example, Price\(^*\) column is just a convenient variable that takes the same values as the closing prices. There are other indicators, the column describes the standard deviation price indicator \(\text{Indstdprc}\), describes the imbalance between bids and asks \(\text{Indvol}\), \(\text{Indvolsd}\). Furthermore, we integrated the \(\text{BA_var1}\) indicator. Regarding the price category indicators, we added the \(\text{Indprc}\) field, which is related to the average price indicator category \(\text{PInd_1}\). Additionally, the \(\text{Indstdprc}\) column describes the standard deviation price indicator \(\text{PInd_2}\). For the volume indicators, we added the \(\text{indvol}\) field to describe the volume average \(\text{VInd_1}\) indicator and \(\text{indvolsd}\) field for the second volume indicator \(\text{VInd_2}\), which calculates the standard deviation of the volumes in this stock.

\[
BA\_\text{var}2_t = \begin{cases} 
1 & \text{if } BA\_\text{var}1_t \geq 0 \\
0 & \text{if } BA\_\text{var}1_t < 0 \\
\text{null} & \text{if } BA\_\text{var}1_t = 0
\end{cases}
\]  

Finally, the Bids/Asks Indicator is calculated as:

\[
BA\_\text{Ind1}_t = \frac{\sum_{i=1}^{t-18} BA\_\text{var}2_i}{20}
\]

4 Analysis

We performed and ran the analysis using SPSS Clementine software. The analysis is based on the built financial indicators explained in the previous section; it mainly has three phases: No lags of touting email messages (Phase I), up to 5 days lag of the touting email messages (Phase II), and aggregate summary of the 5 days lag of the touting email messages (Phase III).

This work adopts the CRISP-DM model to define the problem of the spam scenario, selecting and understanding the dataset, preparing the dataset, modeling, and evaluating. Before performing any analysis the dataset requires data cleaning, integration and preprocessing. This kind of data preprocessing helps to produce quality data and a tuned dataset from the raw data before employing the model. As the most touting email messages occur from the period 02-01-2004 to 08-09-2005, we filtered the dataset to match this criterion and discard any records before the year 2004. This produces a table that contains the “CWTD” stock trading activities from 02-01-2004 to 08-09-2005. In addition, the built financial indicators have been added to the dataset.

Figure 3 represents the final filtered dataset which contains the date column, followed by the bid price, the ask price, closing price and volume, as they were originally taken from the cited web site. Furthermore, we integrated the touted column into the dataset to show the number of touted messages received in a particular day. Price\(^*\) column is just a convenient variable that takes the same values as the closing prices. There are other columns that correspond to the built indicators that were previously defined. For example, \(\text{Prc-Avg}\) corresponds to the \(\text{BA}\_\text{var1}\) indicator. \(\text{Signal}\) field corresponds to the \(\text{BA}\_\text{var2}\) indicator. Moreover, \(\text{20davg}\) describes the imbalance between bids and asks \(\text{BAInd_1}\) indicator. Regarding the price category indicators, we built the \(\text{Indvol}\) field, which is related to the average price indicator category \(\text{PInd_1}\). Additionally, the \(\text{Indstdprc}\) column describes the standard deviation price indicator \(\text{PInd_2}\). For the volume indicators, we added the \(\text{indvol}\) field to describe the volume average \(\text{VInd_1}\) indicator and \(\text{indvolsd}\) field for the second volume indicator \(\text{VInd_2}\), which calculates the standard deviation of the volumes in this stock.
The data mining techniques that have been utilized in this study required the existence of a categorical field to functionally work. In this case, we set a new field to the dataset called Toutlabel based on a conditional format. This condition classifies the touted records into four categories high (touted>=30), medium (15<=touted<30), low (0<touted<15), and notout (touted=0). We set a conditional format based on a histogram for the touting email messages as shown in figure 4. The cutting point was 50 email messages.

![Figure 4. Toutlabel Histogram](image)

Logically, the touting should not occur very often; thus, this massive amount of notout records should be balanced. As shown in the previous histogram the green bar describes the notout category which has the highest proportion in the dataset. Unfortunately, this may have a negative impact and misdirect the techniques to predict the highest possibilities of notout transactions. For that reason, we adjust the current balancing directives of the dataset and decrease the proportion of notout records in the training dataset. It is minimized to 25% of the records and set the proportion of High and Medium to use the whole sample (100%) which is the study objective.

In our study, we used the lag analysis to examine the effect of the tout spam on the “CWTD” stock after a few days of dissemination. The objective of this analysis is based on our assumption that tout spam could heavily affect the trading behavior of the stock in the days following the email distribution. Therefore, we started our analysis by examining the impact of tout spam on the cited stock at the same day of the email distribution. Additionally, we evaluated the affect of stock spam on “CWTD” through five consecutive trading days. This helps to forecast the moving average of email touting which pinpoints any changes in the stock behavior. Lastly, this study employed the aggregation of the 5 days lag as another analysis to evaluate the abnormal behaviours of the spam data in the whole 5 days at the same time.

The experiment can be described as follows: Given the fact that “CWTD” stock has been touted form 02-01-2004 to 08-09-2005, we employ data mining techniques which should be able to predict the highest possibilities of High and Medium categories in the “toutlabel” field of this stock. For the validity and reliability of study, we train 50% of the dataset as a random sample to evaluate the whole analysis and eventually generate a model then test it. This is a real world scenario where the data preparation has been revisited more than once for the amendments needed to fit the modeling approach. Furthermore, it will help in training the model on a sample period then predicting the subsequent time period.

The algorithms were evaluated: C5.0 for building a decision trees, neural network, and logistic regression. All models were built using SPSS Clementine as mentioned before. C5.0 model was run with the default parameters. The simple training mode with accuracy favour option was selected. Furthermore, C5.0 parameters were set automatically using pruning severity of 75% and minimum of 2 records per child branch to produce the most accurate possible fraudulent tree. The “toutlabel” field decision tree is the result of the generated model. Secondly, neural network were used with the “prune” method. This method begins with a large network and removes (prunes) the weakest units in the hidden and input layers as training proceeds. In addition, the three hidden layer were utilized in neural network prediction. This study used 100% of the specified proportion as a sample of the data to train the neural network. However, this method is often slow, but it usually gets better results than other methods. Lastly, logistic regression was used with the “multinomial” procedure because “toutlabel” field is a set field that has more than two values. Furthermore, the method that has been used in this model is the “backwards” method that initially contains all of the terms as predictors, and terms can only be removed from the model. Basically, it initiates a model contains all features and iteratively removes feature that do not significantly add value to the model. Again, the default Clementine parameters were employed.
5 Discussion and Results

This study evaluates and discusses the ability of the generated model to engender accurate predictions based on the aforementioned analysis. Figure 5 demonstrates the results for the output field “toutlabel”. The top level section contains a table showing the number and percentage of correct and incorrect rate predictions and the total number of records that have been processed. Furthermore, within the “toutlabel” field section is a subsection for each prediction field associated with that output field. Give the fact that “toutlabel” field is a categorical output, the predictions rates were based on scoring records and comparing the responses predicted by the model to the actual results of the “toutlabel” field. This contributes to evaluate the generated model as will be explained in the following sections.

![Figure 5. Toutlabel Field Result Comparisons](image)

We began our analysis by examining the abovementioned data mining techniques through three phases (phase I, phase II, phase III) before partitioning the dataset. The results are summarized in table 1. Phase I experiment shows that C5.0 technique identified 14 (3 high, 11 medium) abnormal behaviours over the full range of data. In this analysis C5.0 scored 58% correct prediction rate. Neural network techniques were able to detect 19 cases with a correct rate of 66%. Logistic regression was slightly worse, detecting only 2 high touting labels with 46% correct rate.

Phase II experiment shows that C5.0 successfully detected fraudulent cases from the whole sample with a correct rate ranged from 66% to 71%. For example, at the first day of touting (1st day lag) after the email distribution C5.0 could recognized 27 high and medium touted labels. Second day (2nd day lag) shows a slight decrease to 25 touted stocks. The number of touting decreased to 15 in the third day and rose up again to 26 touted in the fourth day. On the fifth day it decreased again to detect only 18 cases. In contrast, the neural network did not come up with any touted stock for the whole 5 days with a correct rate ranged from 44% to 52%. At the same time the logistic regression was just able to detect 2 touted stocks at the first day and only one touting in the second day with correct rate ranged from 41% to 48%.

Phase III shows that C5.0 generated impressive results in this stage by detecting 47 touting email messages with a correct rate of 71%, which is the best detection result in the whole analysis. Neural networks were able to detect 13 cases with a correct rate of 50%. Unfortunately, logistic regression only picked one case with a correct rate of 45%.

Table 1. The Summary of Primary Results

<table>
<thead>
<tr>
<th>Analysis Stages</th>
<th>C5.0 Correct rate</th>
<th>Neural Network Correct rate</th>
<th>Logistic Regression Correct rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st day lag</td>
<td>58%</td>
<td>66%</td>
<td>40%</td>
</tr>
<tr>
<td>2nd day lag</td>
<td>69%</td>
<td>51%</td>
<td>47%</td>
</tr>
<tr>
<td>3rd day lag</td>
<td>63%</td>
<td>52%</td>
<td>55%</td>
</tr>
<tr>
<td>4th day lag</td>
<td>72%</td>
<td>52%</td>
<td>48%</td>
</tr>
<tr>
<td>5th day lag</td>
<td>66%</td>
<td>44%</td>
<td>41%</td>
</tr>
<tr>
<td>Aggregate of 5 days</td>
<td>71%</td>
<td>50%</td>
<td>45%</td>
</tr>
</tbody>
</table>

Table 2 shows the summary of the training analysis results (50% of the sample) using the three data mining techniques. Overall, the three techniques in phase I have not performed well in this analysis. C5.0
technique detected only 2 high touted labels as abnormal behaviours of the “CWTD” stock with 58% correct rate. Neural network were not able to pick any cases of high or medium touting with a correct rate that reached 43%. Logistic regression detected just one high touting label with a total of 48% correct rate.

However, in phase II C5.0 successfully detected abnormal behaviours in this stage with a correct rate ranged from 64% to 69%. For the first day of touting the C5.0 came up with 11 touted labels. The second day shows an increase to 25 touted stocks. The number of touting decreased reaching 15 in the third day and no cases were detected in the fourth day. For the fifth day the number of touting reached 13 cases. Unfortunately, the neural networks were able to detect 4 cases only in the first day and nothing in the other days, with a correct rate range from 44% to 52%. At the same time, the logistic regression was able to detect 3 touted stocks in the first day and 2 touting in the second day, failing to detect any other cases for the rest of the days with a correct rate ranged from 45% to 57%.

In phase III, C5.0 shows a significant success in this stage by detecting 24 touting email messages with a correct rate of 71%. At the same time, neither the neural networks nor logistic regression were able to detect any abnormal case.

Table 2. The Summary of Training Analysis Results

<table>
<thead>
<tr>
<th>Analysis Stages</th>
<th>C5.0</th>
<th>Neural Network</th>
<th>Logistic Regression</th>
<th>C5.0 Correct Rate</th>
<th>Neural Nw Correct Rate</th>
<th>Logistic Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td>No days Lag</td>
<td>2 (2 High)</td>
<td>0</td>
<td>1 (1 High)</td>
<td>58%</td>
<td>43%</td>
<td>46%</td>
</tr>
<tr>
<td>1st day lag</td>
<td>11 (3 High, 3 Medium)</td>
<td>4 (4 High)</td>
<td>3 (3 High, 3 Medium)</td>
<td>64%</td>
<td>35%</td>
<td>57%</td>
</tr>
<tr>
<td>2nd day Lag</td>
<td>12 (2 High, 10 Medium)</td>
<td>0</td>
<td>1 (1 High)</td>
<td>66%</td>
<td>51%</td>
<td>48%</td>
</tr>
<tr>
<td>3rd day Lag</td>
<td>10 (4 High, 6 Medium)</td>
<td>0</td>
<td>0</td>
<td>69%</td>
<td>46%</td>
<td>49%</td>
</tr>
<tr>
<td>4th day lag</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>55%</td>
<td>42%</td>
<td>40%</td>
</tr>
<tr>
<td>5th day Lag</td>
<td>13 (3 High, 10 Medium)</td>
<td>0</td>
<td>0</td>
<td>69%</td>
<td>46%</td>
<td>40%</td>
</tr>
<tr>
<td>Aggregate of 5 days</td>
<td>24 (6 High, 18 Medium)</td>
<td>0</td>
<td>0</td>
<td>71%</td>
<td>46%</td>
<td>54%</td>
</tr>
</tbody>
</table>

Table 3 demonstrates the results of the testing analysis (the other 50% of the sample). In phase I, C5.0 technique detected 8 touted labels as abnormal behaviours of the “CWTD” stock with 60% correct rate. At the same time, neural networks were not able to get any cases with a correct rate of 47%. However, Logistic regression was able to detect three touting label with 52% correct rate.

It is clearly seen in phase II that C5.0 was successful in detecting abnormal behaviours for this sample, with a correct rate ranged from 64% to 68%. For the first day touting, C5.0 picked nine cases. The second day shows a dramatic decrease reached to two cases. The number of touting rose up to reach eleven cases in the third day and continue increasing up to fifteen cases in the fourth day. The fifth day reached eleven touting stocks. Unfortunately, the neural networks were unable to detect any cases in this stage. At the same time, the logistic regression was able to detect two touted stocks on the fourth day and three in the fifth day with correct rate ranged from 43% to 52%.

Again in phase III, C5.0 performed well by detecting twenty eight touting email messages with a correct rate reaching 83%. Neural networks did not detect any abnormal case in this stage. Additionally, logistic regression picked five cases only with 46% correction rate.

Table 3. The Summary of Testing Analysis Results

<table>
<thead>
<tr>
<th>Analysis Stages</th>
<th>C5.0</th>
<th>Neural Network</th>
<th>Logistic Regression</th>
<th>C5.0 Correct Rate</th>
<th>Neural Nw Correct Rate</th>
<th>Logistic Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td>No days Lag</td>
<td>8 (2 High, 6 Medium)</td>
<td>0</td>
<td>3 (1 High, 2 Medium)</td>
<td>60%</td>
<td>47%</td>
<td>52%</td>
</tr>
<tr>
<td>1st day lag</td>
<td>9 (6 High, 3 Medium)</td>
<td>0</td>
<td>0</td>
<td>68%</td>
<td>46%</td>
<td>46%</td>
</tr>
<tr>
<td>2nd day Lag</td>
<td>2 (2 High)</td>
<td>0</td>
<td>0</td>
<td>60%</td>
<td>42%</td>
<td>45%</td>
</tr>
<tr>
<td>3rd day Lag</td>
<td>15 (3 High, 8 Medium)</td>
<td>0</td>
<td>2 (1 High, 1 Medium)</td>
<td>65%</td>
<td>46%</td>
<td>56%</td>
</tr>
<tr>
<td>4th day lag</td>
<td>11 (3 High, 7 Medium)</td>
<td>0</td>
<td>3 (2 High, 1 Medium)</td>
<td>64%</td>
<td>46%</td>
<td>52%</td>
</tr>
<tr>
<td>5th day Lag</td>
<td>18 (1 High, 7 Medium)</td>
<td>0</td>
<td>3 (1 High, 2 Medium)</td>
<td>83%</td>
<td>48%</td>
<td>46%</td>
</tr>
<tr>
<td>Aggregate of 5 days</td>
<td>28 (6 High, 18 Medium)</td>
<td>0</td>
<td>0</td>
<td>71%</td>
<td>46%</td>
<td>54%</td>
</tr>
</tbody>
</table>

As it was expected, some techniques performed better than others for this particular problem. Specifically, the C5.0 technique outperformed the alternatives techniques in the phases of the analysis. Neural networks and linear regressions were slightly worse, but they still may be suitable for other types of problems. These results are in line with previous results, such as the research of [2], in terms of the order and the power of classification the techniques. As shown in [2], the C4.5 algorithm also outperformed the neural networks, and the logistic regression, for a problem of classification for early detection of insider trading in the option market. At this stage of the study, it is premature to confirm which one of the five proposed financial indicators was the best
for the modelling phase. All indicators performed well in the analysis and helped the techniques for the classification task.

The results presented in this section suggest that some investors actually do respond to the spam stock touts, in concordance with previous spam research [3] and with the accumulated effect captured in the 5 days lag analysis. The experiment shows that aggregate summary of the 5 days lag analysis has the highest correct rates range from 71% to 83% of the C5.0 technique, as well as a successful analysis to detect the highest possibilities of abnormal behaviour in all phases of the spam case as highlighted in the tables 1, 2 and 3. Regarding the reliability and the validity of the analysis, especially the testing phase results prove that the data mining models and techniques can work and detect fraud patterns of this kind of problem.

6 Conclusions

Spammers are utilizing the Internet as a framework to target large number of investors. Spam emails have been used as one of the information-based manipulation tool. Spammers (manipulator) are working to drive the price up or down and change the value of the announced stock and gain illegal profit. Therefore, this study uses the China World Trade Corporation “CWTD” stock as a case study to detect the highest possibilities of fraudulent activities. This paper focus on utilizing data mining techniques to detect any fraudulent activities.

The experiment shows that the proposed analysis is able to identify the highest possibilities of High and Medium categories in the “toutlabel” fields of the “CWTD” stock. The results strongly suggest that the data mining models and techniques can work and detect fraud patterns for these kinds of problem. Particularly, the analysis shows that C5.0 technique outperformed the alternatives techniques in all the phases. Neural networks and linear regressions were slightly worse, but they still may be suitable for other types of problems. The built financial indicators performed well and helped the techniques to work properly and produced a reasonable results.

The analysis demonstrates that aggregate summary of the 5 days lag analysis has the top correct rates range from 71% to 83% of the C5.0 technique, as well as a successful analysis to detect the highest possibilities of abnormal behaviour in the two phases of the spam case. That could be an indicator of some investors acting upon the spam stock touts, in the same line with previous spam research of [3] and with the accumulated effect captured in the 5 days lag analysis.

Following the work reported in this paper, there are a number of issues that need to be addressed through further investigation. In particular, the research did not have access to neither the intra-day price transactions nor the number of bid or ask offers of “CWTD” stock that can provide details about any suspicious price movements occurring during the day. Additionally, the date range did not allow the research to look for re-occurring instances of trading anomalies. The paper did not consider the control of the news effects. The research focused only on one market (Pink Sheets market) which is characterized as an unstructured market. Furthermore, the techniques and the algorithms used in this research can be expanded to include other techniques such as neural networks, kappa statics and others.

Finally, further research could address additional financial indicators based on the intra-day price transactions. Future work could also consider using text mining techniques to analyze the common characteristics of the spam emails. The combination of different sophisticated techniques and algorithms is expected to increase the accuracy of detecting different manipulation patterns.

References

MODELING PATH VIOLATION IN A DECISION SUPPORT SYSTEM FOR AIRCRAFT APPROACH AND DEPARTURE

Laura Saviciene

Vilnius University, Department of Mathematics and Informatics, Naugarduko 24, LT-03225
Vilnius, Lithuania, laura.saviciene@mif.vu.lt

Abstract. The control of approaching and departing aircraft is an important function of the air traffic control. In the EU FP6 SKY-Scanner project, it has been proposed to use the lidar (laser radar, Light Detection And Ranging) for the aircraft surveillance. It will improve the situational awareness of the human decision-makers – the controllers. As part of the project a decision support system (DSS) is being developed. It is based on the radar and lidar data fusion. The DSS estimates possible risks for aircraft and proposes corrective actions to the controllers. This paper presents design decisions and advances in the DSS development. Areas of interest include representing knowledge about normative requirements for aircraft trajectories, calculating risk of path violation and visualization of detected deviations in approach and departure phases of flight. Approach data model is defined in terms of fly-over points. Eight path violation types are discerned. A method to model path violation risk is proposed. It is based on how much the observed trajectory deviates from normative requirements. An interface paradigm based on 2D and 3D view integration is defined. Finally, a two-area decision support model consisting of strict and non-strict constraint checking areas is proposed.

Keywords: knowledge representation, air traffic control, decision support model, information visualization, instrument approach procedures.

1 Introduction

Air traffic control (ATC) is a service provided by the ground-based controllers for the purpose of preventing collisions and maintaining an orderly flow of traffic [13]. One of the ATC functions is the control of approaching and departing aircraft. Surveillance equipment (primary radar, secondary radar, etc.) is used to establish the controllers’ situational awareness of their assigned airspace. In the EU FP6 SKY-Scanner project, it has been proposed to use the lidar for the aircraft surveillance to improve the controllers’ awareness and decision-making capacity. Lidar is installed on ground and, unlike other surveillance systems (secondary surveillance radar or automatic dependent surveillance broadcast), does not require additional equipment to be installed on the aircraft. Lidar is more precise than the primary radar, but it may function worse in certain weather conditions (rain, fog). So, the lidar could be used in conjunction with the primary radar [15].

Figure 1. The functions of the decision support system (DSS)

New air traffic management systems have a long requirements building cycle. Research and development projects generate ideas that are examined and elaborated in subsequent projects. Before the implementation, a solution is verified from various perspectives in different projects. Presented work is performed within the project that aims at developing a novel laser tracking technology – the SKY-Scanner system – capable to detect and track aircraft up to at least 6 nautical miles from aerodrome traffic zone (ATZ) barycenter. The primary target is demonstrating the use of lidar for aircraft detection and tracking. A decision
support system (DSS) for aircraft approach and departure is part of the SKY-Scanner system. It performs radar and lidar data fusion, estimates possible risks and proposes corrective actions to the controller (Figure 1).

This paper concentrates on the path violation risk estimation issues. The task of the system is to evaluate if the observed aircraft trajectory (the “is” trajectory) is “correct”. The “correct” trajectory (the “ought-to-be” trajectory) is defined in terms of requirements, which are listed in normative documents that regulate flights (e.g., flight rules and terminal procedures).

Currently, the project is in the development phase. The DSS research prototype is being developed incrementally in Matlab/Simulink. Simulations are used to verify if the model is working as expected. Existing flight data is used to fine-tune the algorithms.

In the following sections, we present current issues and design decisions in modeling path violation in the DSS. Based on the study of the approach/departure regulations, the following results have been reached:

1. Conceptual approach/departure model, including flight phase models.
2. Approach/departure constraint data model.
3. Eight path violation types. They are mapped to relevant flight phases.
5. Interface paradigm with requirements for trajectories being represented on 2D projections.
6. Decision support scenario consisting of a soft control area and a strict control area.

2 DSS as a Knowledge-Driven System

We regard the DSS as a knowledge-driven system. The system receives information about an “is” trajectory and uses the knowledge about the “ought-to-be” trajectory to make a decision whether the observed trajectory is “correct” and what actions to take, if it is not. The “ought-to-be” trajectory can be extracted from normative documents.

Treating the DSS as a knowledge-driven system has some advantages:

- The system is more adaptive – only the “knowledge” part of the system has to be changed if some rules change;
- Decisions are more transparent, they are easier to understand and rely on for the domain experts that will be using the system.

There are several knowledge sources, that are taken into account in the decision making process. Besides explicit knowledge, which is found in the documents, tacit knowledge also has to be considered. For example, “the pilot knows that half-a-dot deviation on the horizontal situation indicator is still acceptable”.

![Figure 2. From “rules in law” to “rules in software” (DSS)](image)

Thus the main problem in the development of the DSS is representing the rules that define the “ought-to-be” trajectory (Figure 2). The rules, which are presented in a human-convenient form (e.g., charts) in the normative documents (“rules in law”), should be represented in the DSS data structures (knowledge base) and algorithms (“rules in software”).

Most reports on aviation-related systems [6, 9, 11] show the traditional “problem-solving” approach and do not look deep into knowledge representation.
3 Instrument Approach Procedures

We use instrument flight rules (IFR) as a source for the “ought-to-be” trajectory requirements, as they define constraints, which can be used to evaluate the observed trajectories. The instrument approach procedures are described in this section.

Instrument flight rules are regulations and procedures for flying aircraft by referring only to the aircraft instrument panel for navigation. Even if nothing can be seen outside the cockpit windows, an IFR-rated pilot can fly, while looking only at the instrument panel. Most scheduled airline flights operate under IFR.

Instrument approach procedure is a series of predetermined maneuvers from the initial approach fix to a point from which a landing can be completed [13]. Approaches are classified as either precision or nonprecision, depending on the accuracy and capabilities of the navigational aids used. Precision approaches utilize both lateral (localizer) and vertical (glide path) information. Nonprecision approaches provide lateral course information only.

Instrument approach procedures are depicted in the Instrument Approach Charts. All aerodromes, where instrument approach procedures are established, should issue Instrument Approach Charts [1]. The charts are prepared and designed in accordance with International Civil Aviation Organization (ICAO) requirements and recommendations. These documents graphically depict the specific procedure to be followed by the pilot for a particular type of approach to a given runway. There are different procedures for different navigational aid types – very high frequency omni-directional radio range (VOR), non-directional beacon (NDB), instrument landing system (ILS) and others. The number of controlled parameters is also different. ILS procedures provide most information about the approach. Procedures depict prescribed altitudes and headings to be flown, as well as obstacles, terrain, and potentially conflicting airspace. In addition, they also list missed approach procedures and commonly used radio frequencies (Figure 3).

Figure 3. An example of the approach procedure [7]

An instrument approach may be divided into four approach segments: initial, intermediate, final, and missed approach. Additionally, some routes provide a transition from the en route structure to the IAF [8]:

- Arrival: where the pilot navigates to the Initial Approach Fix (IAF), and where holding (keeping an aircraft within a specified airspace while awaiting further clearance) can take place.
- Initial Approach: the phase of flight after the IAF, where the pilot commences the navigation of the aircraft to the Final Approach Fix (FAF), a position aligned with the runway, from where a safe controlled descent back towards the airport can be initiated.
- Intermediate Approach: an additional phase in more complex approaches that may be required to navigate to the FAF. Intermediate Approach begins at the Intermediate Fix (IF).
• Final Approach: between 4 and 12 nautical miles (NM) of straight flight descending at a set rate (usually an angle of between 2.5 and 6 degrees).
• Missed Approach: an optional phase; should the required visual reference for landing not have been obtained at the end of the final approach, this allows the pilot to climb the aircraft to a safe altitude and navigate to a position to hold for weather improvement or from where another approach can be commenced.

4 Representing Approach Procedures

Approach phases terminate on fly-over points (Figure 4) that have associated constraints. Approach procedures determine the number of the fly-over points.

Two types of constraints define the approach procedure:
1. Global: defined for the whole procedure;
2. Local: defined for the particular fly-over point.

Global attributes are the following:
1. Name of the procedure;
2. Glide Path (GP in degrees or GP INOP in %);
3. Reference Datum Height (RDH);
4. Obstacle Clearance Altitude or Height (OCA/H) attribute for each airplane type;
5. Sink Rate (SR, in feet per minute) for a given Ground Speed (GS);
6. Time needed to fly between defined points for a given GS;
7. Runway orientation (in degrees).

Summarizing the fly-over point constraints the approach procedures were concluded to have the following local attributes:
1. Name of Distance Measuring Equipment (DME) device (required);
2. Name of a fly-over point (optional);
3. Lateral distance to DME (required);
4. Lateral distance to Touchdown Point (TP) (required);
5. Altitude (required);
6. Course or track (required).

Table 1 provides an example of local constrains for Napoli/Capodichino airport approach procedures through IAF “Bento” fly-over point.

<table>
<thead>
<tr>
<th>No.</th>
<th>Reaching point type and name if available</th>
<th>ILS-P procedure for Runway 24 DME INP</th>
<th>Track</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DME, nautical miles</td>
<td>Alt, feet</td>
<td>°</td>
</tr>
<tr>
<td>1.</td>
<td>IAF “Bento”</td>
<td>19</td>
<td>7000</td>
</tr>
<tr>
<td>2.</td>
<td>IF</td>
<td>16</td>
<td>5900</td>
</tr>
<tr>
<td>3.</td>
<td></td>
<td>13</td>
<td>4830</td>
</tr>
<tr>
<td>4.</td>
<td>FAF</td>
<td>10</td>
<td>3770</td>
</tr>
<tr>
<td>5.</td>
<td></td>
<td>7</td>
<td>2730</td>
</tr>
<tr>
<td>6.</td>
<td></td>
<td>5</td>
<td>2000</td>
</tr>
<tr>
<td>7.</td>
<td></td>
<td>4</td>
<td>1646</td>
</tr>
<tr>
<td>8.</td>
<td></td>
<td>3</td>
<td>1293</td>
</tr>
<tr>
<td>9.</td>
<td></td>
<td>2</td>
<td>939</td>
</tr>
<tr>
<td>10.</td>
<td>0.8</td>
<td>504</td>
<td>236</td>
</tr>
</tbody>
</table>
The table is interpreted the following way: if the aircraft is flying according to the ILS-P procedure, the course should be 236°, and, for example, at the distance of 13 nautical miles to the airport, the aircraft’s altitude should be 4830 feet (see line 3).

5 Modeling the Risk of Path Violation

Based on the approach/departure procedure constraints, eight path violation types are defined and mapped to relevant flight phases:

1. Vertical position violation
   1.1. Altitude violation (Initial Climb, En-Route, Holding, Initial Approach, Final Approach and Missed Approach);
   1.2. Glide path violation (Final Approach);
   1.3. Obstacle clearance violation (Initial Climb);

2. Speed violation
   2.1. Climb gradient violation (Initial Climb and Missed Approach);
   2.2. Indicated airspeed violation (Initial Climb, Holding and Missed Approach);

3. Position violation
   3.1. Course violation (Initial Climb, En-Route, Holding, Initial Approach, Final Approach and Missed Approach);
   3.2. Maneuver area violation (Holding);
   3.3. Circling sector violation (Holding).

A DSS function is to estimate the risk of path violation. This section is devoted to risk modeling. We propose modeling with linear representation. It is based on how much the “is” value deviates from the “ought-to-be” value. Risk can be defined as a function that maps the deviation into the risk level from the interval [0, 1]. Zero means no risk, 1 means the maximum risk. Path violation risk is currently defined for vertical (altitude and glide path) and horizontal (course) violations. The “ought-to-be” values from the procedures are given some allowable deviation, like a funnel centered on the “ought-to-be” value. When the deviation of the “is” value from the “ought-to-be” value is small, the risk of path violation is zero, or small. As the “is” deviation approaches the limits of the funnel, the risk increases. When the deviation is outside the defined limits, path violation risk is denoted by 1. The result is similar to a fuzzy set [17] membership function (Figure 5).

![Figure 5. Calculation of path violation risk](image)

A narrow funnel is defined for the glide path constraint (Figure 5a). The risk function is interpreted in the following way. Path violation risk is zero only when “is” value is equal to the “ought-to-be” value, for example, 3° (i.e., the deviation is zero). The risk increases as the deviation gets close to –0.25° or 0.25°. The risk of glide path violation is 1, when the deviation exceeds these limits.

Similarly, a wider symmetrical funnel is defined for course (horizontal) violation (Figure 5b). Demonstration model allows ±5° course violation. A deviation from –2° to 2° gives zero course violation risk. Course violation risk increases for deviation values that are more than 2° (less than –2°) and approach 5° (–5°). The risk of course violation is 1, when the deviation exceeds the limits.

Funnel for altitude violation is asymmetric (Figure 5c). The approach charts depict the minimal allowed altitude value, i.e. tolerance for the negative deviation (“is” values lesser than “ought-to-be”) should be smaller
than for the positive deviation. When the “is” altitude is less than the “ought-to-be” altitude minus 0.5% it gives an altitude violation risk of 1. A deviation from 0% to 2% gives zero violation risk. The maximal allowed deviation is 5%. Approaching the limits (–0.5% or 5%) increases the altitude violation risk.

The allowed deviation values are chosen for demonstration purposes only and are subject to fix by experts. For example, ILS instrument precision requirements may be used, as the pilot controls the aircraft according to instruments, and the DSS must not require greater precision than the equipment provides.

6 DSS Interface Paradigm

The design of DSS interface defines how controllers will perceive and interact with information provided by the DSS: aircraft positions, predicted trajectories and detected risks. People, not computers, fly the aircraft, direct the traffic, and have the final word in all decisions. Therefore, how computer systems present information to the human controllers, play a crucial role in the effectiveness of ATC systems [3].

User-need investigation showed that there are two goals to achieve: improve situational awareness and reduce cognitive workload. The system should help the controller to understand if aircraft trajectories correspond to approach and departure procedure requirements.

Recent developments in the field of visualization for ATC show that traditional 2D visualization is no longer sufficient [5, 16], and pure 3D visualization has some significant drawbacks [2, 16]. Some authors propose to use 3D in combination with the more traditional 2D display to see both contextual and required altitude information at the same time [2].

One recent project [12] analyzed innovative visualizations in ATC, Command and Control, Medical, Geographic and other fields. Based on this analysis, authors draw a conclusion that not all possibilities to integrate 2D and 3D in ATC systems have been tried. They propose several new strategies [12]. These ideas were reviewed to see how they could be used in the DSS interface:

- The user selects a portion of the main 2D view and that portion is represented in 3D. It was noted that users prefer such solutions that preserve continuity (i.e., make it easy to identify aircrafts after switching on the 3D picture) and do not use distortion.
- To show 2D walls in the main 3D view and mark projection of the aircraft position on the wall. For example, a vertically oriented gradation (“altitude ruler”) helps monitor and control the holding stack.
- To use augmented reality. In this case, a 3D view of the whole airspace is provided, and virtual tools are used to closer examine the situations of interest.

The chosen solution has to take into account the technical limitations of the system model. There will be two 21” LCD color displays: one for graphic and another for hypertext visualizations [14]. Hence, complex augmented reality or virtual reality solutions are not suitable.

Figure 6. DSS user interface example

After reviewing solutions proposed in other projects, an interface model based on 3D was selected. The system will present a pure 3D display with 2D projections (“curtains”) in relevant places, showing aircraft positions and requirements for trajectories, or rulers helping estimate distances more accurately. The prototype shown in Figure 6 meets defined user needs in the following way:

- 3D display improves situational awareness;
- “Curtains” reduce the cognitive workload.

- 85 -
Selection of the objects (the number of objects and the level of detail) to be visualized in the interface has to be optimal. The DSS works with a lot of data every moment – aircraft coordinates, past and foreseen trajectories, risk alerts, etc. Some information is presented in a separate display (flight data) [15]. However, switching from one display to a separate information source could be time consuming and taking attention away from the traffic situation [10]. So, some authors suggest that all required information should be shown in one display, for example, attaching information about an aircraft to its representation. On the other hand, one must consider that minimizing clutter and distractions is vital to controllers [4].

Objects that are important to represent in the DSS interface are aircraft, past and estimated future trajectories, terrain and the approach/Departure requirements for the trajectories. Alternatives for object representation were considered: sphere, cone or 3D model for the aircraft, continuous or dotted lines, or a “ghost” plane for past and estimated trajectories, coloured map, generalized or photo-realistic models for terrain.

In the approach procedures, main requirements for the aircraft trajectory are shown in profile view (Figure 7). These requirements can be represented in a 2D projection – the wall (Figure 6). The profile view of the procedure could be quite long (in some procedures – more than 10 NM) and, showing it in real proportions, the scene becomes too fine to be useful. So, it was decided to try out these alternatives: represent requirements for trajectories in separate segments (i.e., with little or no possibility for the overview) or to use distortion of the projection. In other projects distortion was not accepted positively [12].

![Figure 7. An example of the profile view of the approach procedure [7]](image)

7 Decision Support Usage Model

Current DSS prototype proposes the following usage scenario. The overall situation is presented in 3D view window with generalized airport landscape and tracks detected by SKY-Scanner system. A 2D control panel holds user interface buttons and a message board for the text messages. Violations are visualized in 3D view using colors and explained in the message board.

The whole observed area is divided into two parts:

- Soft control area where only altitude control is performed;
- Strict control area when certain landing procedure is assigned to the current track, and procedure constraints (altitude, speed, and course) can be followed.

Strict control area is defined between IAF, FAF and TP. When track receives a clearance for landing, the landing procedure is assigned for the track. Operator turns on the DSS support and assigned procedure is presented on the projection walls. Aircraft position validity can be easily detected visually and confirmed with colours. The DSS control panel presents the current information about tracks and possible risks.

In soft control area the global parameter OCA is traced.

The button “Show prediction” visualizes a predicted situation in a determined time interval.

When airplane receives a clearance for certain landing procedure, the DSS support scenario is available:

- Turn on the projection curtains.
- Select the landing or take off procedure, assigned to current track. The projections of assigned procedure appear on the curtain.
- View the situation: airplane position should be on the lines of the visualized procedure.

Path violation can be detected on the projection walls: airplane projection signs have to be right on the procedure lines. Path violation risk is visualized using colors:
• **Green** means that risk is equal to zero;
• **Yellow** means that current positions approaches the safe limit and path violation risk is in the interval \((0, 1)\);
• **Red** indicates a path violation, which means that position is out of the safe funnel.

8 Conclusions

The use of lidar for aircraft tracking will enable controllers to observe phases of flight that are near to the ground. The aircraft approach and departure decision support system will allow for some control over the aircraft in these phases. This is not possible using only the primary radar.

DSS is designed as a knowledge-based system. It tracks the aircraft in accordance with the requirements for trajectories provided in the approach and departure procedures.

The procedures provide a complex definition of the requirements composed of both textual and graphical descriptions. The main constraints form a series of rectangular “gates” defined at certain distances from the airport. Moreover, there are several such procedures for each airport. This makes representing the requirements in the system a challenge.

The current research has undertaken the first steps in representing the approach/departure procedure requirements and using them for path violation risk estimation and visualization. Although the model is promising, it is still far from being used in the operational environment.

Future work includes, but is not limited to, estimating future aircraft positions and using them to foresee future path violations and testing the models with existing flight data.

References


INVESTIGATION OF SELF-ORGANISATION
PHENOMENA AND PROCESSES IN DECISION SUPPORT
SYSTEMS ACCORDING TO RELATIONSHIPS OF
PARTICIPANTS’ GOALS

Igor Kirikov, Alexander Kolesnikov, Sergey Listopad
Kaliningrad branch of the Institute of Informatics Problems of the Russian Academy of Sciences, 5 Gostinaya street, Kaliningrad, Kaliningrad region, Russia, kfipiran@yandex.ru, avkolesnikov@yandex.ru, ser-list-post@yandex.ru

Abstract: This is an analytical review of known approaches to modeling of self-organisation of groups of people, making decisions. The authors consider one of the approaches to creation of an intelligent computer system of support for self-organising decision-making, based on the analysis of experts’ goals. A method of measurement of intelligent agents’ interaction in computer decision support systems, relevant to assessment of expert teams’ performance and to improvement of the quality of work of such systems, is proposed. A universal structure of a multi-agent system is described, as a decision support system model for carrying out computing experiments for research into dependence of self-organising processes on the relationships of agents’ goals.

Keywords: computer decision support system, similarity measure of agents’ fuzzy goals, method of measuring intelligent agents’ interaction degree

1 Introduction

While making decisions in difficult unique situations, experts/advisers are involved in the search of variants of solution of a problem or its parts, i.e. decision support systems (DSS) are created. There are numerous reasons for that: collective decisions are more rational and less subjective, the decision-making principle, based on teamwork, is more democratic; the team members, making decisions together, share the responsibility for it, and, in addition, the probability of the decision realization increases because, at least, part of the team is interested in its implementation.

A DSS conceptual model [5] is presented in Figure 1. The arrows, connecting experts, show their versatile interaction. Some experts are subordinated to one or several other experts with regard to the job, i.e. there can be an organisational structure in the DSS. Co-operating during discussions, the experts exchange data, knowledge, explanations, and partial solutions of the common problem. There can be teams of experts, not connected by subordination. Among them, there can be explicit or implicit leaders, which even more ‘escalates’ the heterogeneity of collective decision-making.

![Figure 1. DSS conceptual model](image)

Each team member, expert or decision maker (DM), listens to other participants and expresses his/her own opinion. Generally, each DSS participant has his/her own vision of the problem and a way of its solution and tries to convince the team of its correctness. The decision-making process in the DSS consists in searching for a compromise, controlled by the DM. The purpose of the search is to find ‘a state of resonance’ of the course of discussion in the DSS, which leads to occurrence of a synergetic effect, when the collective integrated solution quality appears better and is free of defects of individual experts’ opinions. The occurrence of the synergetic effect, explaining higher quality of collective decisions in comparison with individual experts’ decisions, depends on the DM’s efficiency in establishing one kind and breaking off other kinds of relations among the participants during the debates.
Looking at collective software engineering as a decision-making process, we can consider the example of an effective DM’s administrative decision, which produced a synergetic effect in the process of development of the Windows NT operating system by the Microsoft Company [13]. The $150-million project was implemented by 250 programmers with many megastars among them. This software product contains over six million code lines, and it was implemented within 3.5 years instead of planned 1.5 years. The coordination was actually minimal: it was a crowd of individual hackers and very small groups with leaders, designing software parts. The reporting was utterly loose, many code writers ignoring even their own leaders. The work was on the verge of collapse, in complete chaos, with a constant deficit of proper management. People independently looked for assistance for troubleshooting. How could the system consistency be achieved in this situation without destroying the creative environment? The project chief David Cutler suddenly ordered that everyone should “eat only one’s own dog food”. This happened after a year of difficult time-marking and meant that the developers were to use previously compiled unfinished and imperfect versions as an operating system. In the construction laboratory, the code fragments were hastily ‘electronically joined’ and distributed as an operating system for all next-weeks works. It was like writing a novel using words and phrases from previously written chapters. This rule turned the crowd of intellectuals disliking one another into a team with a common system of values. Firstly, they were amazed at the crudeness of ‘the dog food’ and the number of their own errors, ruining all plans for the week. Secondly, a positive feedback began to work. Each lucky finding facilitated the code-writing and led to skills, knowledge, and resources to solve it. In [7, 11], the MAS is a collection of four objects: in
t[dependence of the CDSS’s effectiveness on the degree of its members’ interaction had not been paid sufficient
directions in the last decade’s scientific life. The self-organisation process research is based on its
specialists all around the world. The self-organisation concept is one of the most outstanding and promising
strongly depends on the DM’s experience and skills.

The need to advance in solving this problem has been repeatedly emphasized by scientists and specialists all around the world. The self-organisation concept is one of the most outstanding and promising directions in the last decade’s scientific life. The self-organisation process research is based on its multidisciplinary nature and systemic approach. The self-organisation ideas go back to the works of Immanuel Kant, Georg Wilhelm Friedrich Hegel, Charles Robert Darwin, and – in the 20th century – Alexander Bogdanov, Erwin Shredinger, Norbert Wiener, Ludwig von Bertalanffy, Nikita Moiseyev, Ilya Prigogine, Izabella Stengers, Hermann Haken, Werner Von Ebeling, etc. [11]

Thus, it is relevant to develop an intelligent computer decision-making support system (CDSS) capable of advising the DM on the necessity of changes in the team composition, redefinition, and matching of their purposes with the DM’s preferences, based of the team participants’ interaction analysis. For such a CDSS, it is important to study the interaction of the team members as confederates who agree with each other, and competitors who dispute the correctness of the decision. Such a CDSS could simulate the DM’s work in analyzing the current situation, calculating the similarity of the team members’ positions and choosing a strategy for further CDSS activity, in solving complex problems. The analysis of the literature has shown that the issue of dependence of the CDSS’s effectiveness on the degree of its members’ interaction had not been paid sufficient attention.

Therefore, in order to determine whether the probability of arising of a synergetic effect is higher in confederates’ or in competitors’ groups, an original method for estimating the degree of interaction of intelligent agents in a CDSS was developed and is presented in this article. The method will be considered based on the example of a multi-agent system (MAS), as a DSS computing model. This would eliminate extraneous factors, such as the participants’ insincerity, which inherently exists in human communities and affects the research results.

Description of the method for estimating the degree of interaction of intelligent agents in a CDSS starts with considering concept of agent’s goal and its representation in the form of fuzzy set in the second section. This formalized description of the agent’s goal allows providing in the third section measure of similarity of goals, alternative classification of relations between agents and classification of MAS architectures depending on similarity of agents’ goals. In the fourth section, a universal structure of MAS is described, as a DSS model for carrying out computing experiments for research into dependence of self-organising processes on the relationships of agents’ goals.

2 Knowledge and goals of an intelligent agents’ team

The basis of the multi-agent approach is a distributed, interactive social intelligence of cooperating intelligent systems (subsystems, agents), in contrast to the classical approach to artificial intelligence, according to which the single intelligent system must have a global vision of the complex problem and have all necessary skills, knowledge, and resources to solve it. In [7, 11], the MAS is a collection of four objects:
where $A$ is a set of agents, which reflect the DSS participants (the experts and the DM) in Figure 1; $E = \{ e \}$ is the environment, in which the MAS is located, i.e. the mapping of the DSS’s environment; $\text{CL}$ is inter-agent relationships, depicted by arrows in Figure 1; $\text{ORG}$ is a set of the MAS basic architectures. In the MAS, an agent has a partial view of the problem and limited resources to solve it, so in complex situations, interaction of the agents, is required, inseparable from the MAS organisation. The way of the agents’ interaction in the MAS is determined by its basic architecture.

Agent is a hardware or (more frequently) a software-based computer system, possessing the properties of autonomy, sociability, activity, reactivity, and ‘mental qualities’ [12].

Autonomy is interpreted as the agent’s ability to operate without human intervention, to exercise self-control over its actions and internal status. Sociability of the agent is an ability to communicate with other agents, and possibly with humans by means of some language. Reactivity is an ability of the agent to perceive the environment and react to its changes. Agent’s activity is interpreted as ability not only to respond to external events, but also to take independent actions to achieve its goals. ‘Mental qualities’ (intentional concepts) are the components (knowledge, goals, desires, intentions, beliefs, commitments, etc.) of the agent that are responsible for formation of the agent’s preferences and its behavior strategies. The key mental qualities, underlying other qualities, are knowledge and goals.

Knowledge is part of information about the agent, the environment, and other agents, which is not changing on-stream [4, 9].

Goal is essentially the state of affairs, which the DM aims to achieve, having a subjective value for the DM [6]. In [10] goal is perfect anticipation of the result of activity, being its regulator, and in [3] it is a situation or an area of situations to be achieved through operation of the system in proper time. Summarizing these definitions, we identify the goal characteristics as follows: it represents the status of the control object, it is a regulator of activity, it is temporal in nature (a function of time), and subjectively it is useful for the DM.

Summarizing the above definitions, the goal of agent, as the regulator, is the state $st$ of the control object, having some sentimental value for the agent and determining the agent’s activity for a certain period of time. Status $st$ of the control object is described by a set of its properties $P = \{ p_1, ..., p_n \}$, where $n$ is the number of properties describing the state of the control object. One of the properties in the set $P$ can be time associated with the operation of the control object. Then the agent’s goal became dynamic (changes over time).

In the process of modeling, the properties in the set $P$ are represented as variables. It is obvious that to describe real control objects, it is necessary to use modeling methods of various classes, such as: analytical, statistical, logical, linguistic, or fuzzy ones. The methods of each of these classes can ‘work’ with a certain type of variables: deterministic, stochastic, logical, exact linguistic, or fuzzy linguistic, respectively. Thus, individual properties of the control object in the set $P$ can be described by variables of different types, and several toolkits will be used in goal-setting.

This causes the difficulty of decision modeling, if it is required to compare sub-goals, described by methods of different classes. Such situation arises, for example, if there are Pareto-optimal solutions, and it is required to select only one of them. Suppose there is a control object with two properties $p_1$ and $p_2$ as well as two states of the control object $st_1$ and $st_2$, $st_1$ being closer to the goal state $st$ than $st_2$ on the first criterion, and $st_2$ is closer to the goal state $st$ than $st_1$ on the second one. If the properties are expressed in different variables (e.g. stochastic and fuzzy linguistic), processed by methods of different classes, it will be difficult to choose one of the decisions, as different types of variables can be operated only in an integrated environment. However, if the properties are represented by variables of the same type, it is possible to set a metric for two-dimensional vector space of admissible states of the control object, and to determine the distance between $st_1$ and $st$, and between $st_2$ and $st$, and thereupon to compare them. To avoid such situations, we choose one method for presentation of all properties, defining the control object’s state and therefore used to describe the DM’s and agents’ goals.

The analysis showed that the most convenient for this purpose is the apparatus of the fuzzy sets theory. It adequately addresses all types of uncertainty in a DSS and consolidates all available heterogeneous information [2], so the agents’ goal will be described using the fuzzy sets theory apparatus. The agents’ fuzzy goal $fg$ is a fuzzy set with the membership function $\mu^{fg}(st)$, $st \in ST$. The membership function possesses values from the set $[0; 1]$, thus, the higher its value, the closer the control-object state $st$ to the agent’s goal $fg$. Let’s consider how agents’ fuzzy goals interact in the MAS.

3 The method of measuring of intelligent agents’ interaction

The method of measuring of intelligent agents’ interaction will be considered in the case where the state $st$ is described by a single property $p$, i.e $\mu^{fg}(st) = \mu^{fg}(p)$. The source information is the objectives of the
intelligent agents (experts). The result is the type of the MAS architecture on the degree of agents’ interaction as a text line.

Firstly, the agents’ goals are represented by fuzzy goals $f_{g}$, and the agents, whose fuzzy goal is a constant ($\mu^{fg}(p) = const$), are excluded from consideration. Then, in accordance with expression (2), the goals similarity measure is computed for each pair of agents $A$ and $B$.

$$s(A, B) = 0.5 \left( \int_{p_{\min}}^{p_{\max}} \mu_{A,a}^{fg}(p) d(p) + \int_{p_{\min}}^{p_{\max}} \mu_{B,a}^{fg}(p) d(p) \right)$$

where $p_{\min}$ and $p_{\max}$ are the minimum and maximum values of the property $p$, respectively, and $\mu_{A,a}^{fg}(p)$ and $\mu_{B,a}^{fg}(p)$ are the membership functions of fuzzy goals of the $A$ and $B$ agents, respectively.

Formula (2) is half of the sum of the quotient of the area of the gray shaded region to the area of the gray region and of the quotient of the area of the gray shaded region to the area of the shaded region in Figure 2. The values of the similarity measure are the real numbers in the interval $[0; 1]$.

$$\text{Figure 2. Measure of similarity of the agents’ fuzzy goals}$$

Then, based on the value of the measure of similarity, the value of the linguistic variable ‘relationship’ is calculated [5], for each pair of the agents, which is represented by the expression:

$$cl = \{\beta, T, U, Pr_{1}, Pr_{2}\},$$

where $\beta = “relationship”$ is the name of linguistic variable; $T = \{“competition”, “neutrality”, “collaboration”\}$ is a set of its values (the term-set), each of which represents a single fuzzy variable name; $U = [0; 1]$ is a definitional domain of fuzzy variables, being part of definition of the linguistic variable; $Pr_{1} = \emptyset$ is the syntactical procedure, describing the process of formation of new terms from the set $T$ members; $Pr_{2} = \left\{ \mu_{\text{competition}}(s) = \frac{1}{1 + (3 \cdot s)^3}, \mu_{\text{neutrality}}(s) = \frac{1}{1 + (6 \cdot (s - 0.5))^3}, \mu_{\text{collaboration}}(s) = \frac{1}{1 + (3 \cdot (s - 1))^3} \right\}$ – semantic procedure, matching each term of the set $T$, as well as each new term formed by the procedure $Pr_{1}$, with the meaningful content through formation of a respective fuzzy set. Figure 3 graphically represents the membership functions of the fuzzy variables, belonging to the linguistic variable $cl$.

As a result, the $CL$ matrix of the variable $cl$ “relationship” values is to be obtained. Then, for each pair of competing (neutral) agents, the agent, which collaborates with both agents in the pair, is identified. Further, the relationships between the agent and each of the competing (neutral) agents in the $CL$ matrix are replaced with “neutrality”. According to the modified $CL$ matrix, one of the types of the MAS architectures is chosen:

1. an MAS with collaborating agents consists only of collaborating and neutral agents and is completely free of competing relationships;
2. an MAS with neutral agents, in which only neutral relationships exist;
3. an MAS with competing agents, in which there is at least one pair of agents between which the competition relation is established (in such an MAS there could also be neutral and collaborating agents; if collaborating agents exist, they could be viewed as “super agents”; then, all agents of such an MAS will be competing or neutral).
The implementation of the method of intelligent agents’ interaction measurement in a CDSS allows creating intelligent self-organising CDMS systems, capable of rebuilding their structure on-stream, to improve their efficiency.

In order to construct such a CDSS, it is required to include in its structure a “decision-making agent” element, simulating the DM’s work for decomposition of the problem, distribution of subtasks among the agents, integration of the solutions obtained from each of the agents, analysis of interactions between the CDSS participants, organisation of their interaction, and so on. The analysis of interaction function $f_{ai}$ is used both for monitoring the agents/experts’ interaction, and for identification of the MAS architecture, based on the analysis of the agents’ goals, using the method considered above. According to the results of execution of this function, “the decision-making agent” can establish the need to replace one type of agents’ relationship with another and to adjust their goals. In fact, this means that at some point the CDSS modifies its functioning algorithm and converts to the MAS basic architecture of another type. Thus, a change of the MAS organisational structures (architectures) from the ORG set in expression (1) and self-organisation of the CDSS takes place.

The considered method affords an opportunity to clarify the concept of “self-organisation of multi-agent system”. Self-organisation in an MAS is the process of changing of the MAS architecture by an agent, simulating the DM and belonging to it, based on analysis of other agents’ interaction, to improve the quality of decisions. As a result, a self-organised MAS (1) can be expressed as (4):

$$MAS_s = \{A^*, E, CL, ORG, ACT, \tau\},$$

where in addition to the objects considered in expression (1), we’ll allocate $A^*$ as the set of MAS agents, including “the decision-making agent” $a_{dm}$, i.e. $A^* = \{a_1, \ldots, a_n, a_{dm}\}$, where $n$ is the number of agent-experts in the MAS; $ACT$ is the set of the system actions; $\tau$ is a mapping of the set of agents $A^*$ to the set of the MAS agents’ actions $ACT$: $ACT \rightarrow A^*$, $ACT_a = \tau(a)$, and the set of actions of “the decision making agent” $a_{dm}$ includes the analysis of interactions $f_{ai}$, executed in accordance with the above-considered method of intelligent agents’ interaction measurement, i.e. $f_{ai} \in ACT_{ai}$.

It is obvious that to make a decision on replacing one MAS architecture with another, the information about the current MAS agents’ interaction alone is insufficient. To make such a decision, the “decision making agent” has to know which MAS architecture types are more efficient in certain conditions. To obtain such knowledge, a series of computing experiments with an MAS with various types of architecture is to be carried out. To determine the type of the MAS architecture in this series of experiments, the proposed method should also be used. In addition, in order to separately investigate the dependence of the MAS solutions effectiveness on the MAS architecture by the degree of interaction between the agents, eliminating the influence of other factors, the universal MAS structure for the MAS construction, considered in the next section, with the same number of agents and their algorithms of functioning will be used. Thus, the multi-agent systems with various architecture types, involved in the experiments, will differ only by the agents’ goals.

4 The structure of a multi-agent system, to study the self-organisation effect

Above, we considered three classes of MAS architectures with regard to the degree of agents’ interaction: with cooperating agents, with neutral agents, and with competing agents. The creation of such an MAS will be based on a universal MAS structure for construction of hybrid intelligent systems (HIS) proposed in [14], which is shown in Figure 4 in a specified form. In this figure, solid and dashed arrows are relationships between the agents, such as information inquiries, assistance in solving sub-problems, transfer of their decision results, etc. The chain lines show the agents’ interaction with the ontology.
Let’s consider in detail the designed goal of each agent in this structure:

1. The interface agent interacts with the user through the input/output subsystem, inquires the information necessary for problem solving and informs the user about the system’s work result.

2. The decision-making agent is responsible for the activation and synchronization of different agents. The agent develops work plans, in accordance with its existing planning algorithm, for solving of the problem received from the interface agent and verifies whether the plans are executed. It selects one of the alternative solutions, reported to him by the problem-solving agents, or aggregates the results, in accordance with its existing aggregation algorithm, to obtain the final decision, which is subsequently transmitted to the interface agent.

3. The problem-solving agent is a specialized agent with its own knowledge base in the domain needed for solving part of the problem assigned to it; in addition, the agents have basic meta-knowledge of when they should apply for intelligent technology agents’ assistance.

4. The mediator agent tracks the names, models and capacities of all registered intelligent technology agents. This information is stored as records in the intelligent technology agents’ database available to the mediator agent. The interaction with this database is provided by the database management subsystem, which adds, deletes, and searches for records about the intelligent technology agents. The MAS agents can refer to the mediator agent to find out which of intelligent technology agents can assist them in solving the part of the problem assigned to them.

5. The intelligent technology agent provides ‘services’ to other agents with the use of some homogeneous or heterogeneous intelligent technology algorithms. The agent receives tasks from the problem-solving agents or other intelligent-technology agents and sends back the work results. The agent can also apply for other intelligent-technology agents’ assistance. Before such agent can starts working in the MAS structure, it should be registered at the mediator agent, i.e. provide information about subtasks which it is capable of solving and about the intelligent technology it is going to use.

6. The ontology is the basis of the agents’ relationships. The agents interpret the meaning of received messages based on this ontology. The ontology also determines when a problem-solving agent or an intelligent technology agent requires other agents’ assistance.

In the MAS with such structure, each problem-solving agent can take advantage of any intelligent technology represented by intelligent-technology agents, presented in the system. Interactions between the agents of such MAS, in contrast to the classical HIS, are easily interpretable, because they are based on the transfer of knowledge and information. The presence of the mediator agent in the universal structure implies an adaptive organisation of the MAC, allowing to move, add, delete or substitute intelligent agent technology (for example, when an agent who can better perform the same task is presented in the system). The presence of the mediator agent also provides additional stability of the MAS. For example, if one of the intelligent technology agents cannot be used, the problem-solving agent can use another one with the same or similar capabilities,
finding him through the mediator agent. Thus, the interactions between the agents are determined directly during the MAS operating time.

Using this MAS structure allows creating an MAS for solving problems, which are complex in modeling, and whose solution requires application of several intelligent technologies. An example of such problem can be a complex traveling salesman problem [8], which is proposed for testing the MAS performance. A detailed description of this problem won’t be given here. We’ll only say that a complex traveling salesman problem is a serious approximation of the traveling salesman problem to the practical problem of delivering goods to customers by several vehicles. It takes into account such stochastic factors as the probability of occurrence of traffic jams, resulting in the probability of a tardy arrival to the customer, the probability of losses due to cargo breakage, etc. This seriously complicates solving of the problem, making the use of known methods for solving classical traveling salesman problem impossible. On the other hand, the search of ways of delivery of goods to customers without taking those factors into account often results in rather ‘fragile’ decisions, which become ineffective, even after a small change in the problem parameters.

5 Conclusion

So, we have considered the self-organisation effect in a DSS. The measure of similarity of the agents’ fuzzy goals and the method of measuring of intelligent agents’ interaction, based on the analysis of their goals, are proposed. In addition, a universal MAS structure, on the basis of which an MAS with various types of architecture should be implemented for carrying out of computing experiments, is proposed. The experiments are necessary for research into the dependence of synergetic effect occurrence on the type of the MAS architecture.

For each MAS architecture type a part of experiments should be found in their total number, in which the MAS-made decision is better than any of the decisions produced by each individual agent. Thus, it is necessary to calculate, within which MAS the probability of synergetic effect occurrence is higher for arriving at a CDSS-based decision, preferable to any of those offered by each individual agent. This is a subject of further studies. The present study results will allow creating an MAS capable of using positive effects of self-organisation, where a collective solution found by the MAS is considerably better than any of the individual solutions found by each separate agent.

References
THE USE OF INDUCTIVE LEARNING IN INFORMATION SYSTEMS

Ilze Birzniece

Riga Technical University, Department of Systems Theory and Design, Meza 1/3, Riga, LV – 1048, Latvia, ilze.birzniece@rtu.lv

Abstract. Machine learning attempts to build computer programs that improve their performance by automating the acquisition of knowledge from experience. Inductive learning, one of machine learning paradigms, draws inductive inference from a teacher or environment-provided facts. Inductive learning enables the program to identify regularities and patterns in the prior knowledge or training data, and then to extract them as generalized rules. In literature there are proposed two ways of machine learning usage in information systems: (1) for building tools for software development and maintenance tasks and (2) for incorporation into software products to make them adaptive and self-configuring. However, considering information systems in more detail, division in three situations of inductive learning use in the context of information systems can be proposed, namely, first, in the information system development project management, second, to collect the information that is to be built in information system, third, to help the information system to adapt to the changing environment. The analysis of inductive learning role in information system development and usage is given. The future directions point to the e-commerce and similar domains on the Web for the role of inductive learning in information systems.

Keywords: inductive learning, machine learning, information systems, knowledge acquisition.

1 Introduction

Nowadays, the significance of information and information systems (IS) is increasing. Researchers and developers strive for sustainability and adaptability of IS [22]. Machine learning could be considered as one of the tools for ensuring the adaptation of IS. However, this topic has not been largely discussed in the 21st century IS conferences. The paper aims at investigating use of machine learning in general and inductive learning in particular in IS development to promote discussion on this issue. This survey could help to identify opportunities of embedding the learning capability in contemporary [1, 4, 23] IS solutions.

Research on Artificial Intelligence (AI) became active in sixties’ of the previous century leading to the emergence of a new type of software systems - knowledge based systems [9]. Most typical systems of this type are expert systems. Expert systems (ES) are software programs that accumulate and describe expert knowledge in some specific field with the aim to spread that knowledge and solve other tasks of this field. After creation of first expert systems knowledge acquisition has become a subject of computer science. Various methods have been used for knowledge acquisition. They can be divided in manual, semiautomatic and automatic methods [9, 14]. There are manual methods that are used by knowledge engineer in process of knowledge takeover, analysis and structuring. They include brainstorming, interviewing, and protocol analysis techniques. In case where a part of this manual process is carried out with the computer program, methods are called semiautomatic, e.g. knowledge acquisition interfaces. Automatic methods may perform their tasks without knowledge engineer assistance. As the knowledge takeover from human experts with manual methods is both time-consuming and expensive, in many domains the machine learning methods are more preferable because they are more effective and efficient [15]. Inductive learning is one of machine learning subfields highly estimated in data mining, rule inducing for autonomous systems and expert systems due to its ability to create transparent classifier [3]. Inductive learning methods have also been used in developing software products [21]. The purpose of this paper is (1) to show how inductive learning can be used in IS development and (2) to build basis for broadening existing application areas of inductive learning. The rest of the paper is organized as follows. Section 2 presents the definition of IS used for the purpose of this paper. Section 3 provides an overview of machine learning and inductive learning approach. In section 4 existing inductive learning applications in IS are surveyed. Section 5 points to the newest application areas of machine/inductive learning in IS development. Finally the conclusions are drawn.

2 What is information system

It is not an easy task to define what an IS is. Many different definitions have been proposed over the years by researchers and textbook authors. One simple and easy to understand definition is given by Davis: “Information system is a system in the organization that delivers information and communication services needed by the organization” [2]. Another proposed definition that is relevant to discuss in the context of this paper comes from Alter [2]. He defines IS as a special case of work system. The motivation for such definition is founded on fact that businesses operate through work systems. “An information system is a special case of work system in which human participants and / or machines perform work (processes and activities) using
information, technology, and other resources to produce informational products and / or services for internal or external customers” [2]. Defining IS as a work system implies that most of the concepts and knowledge that apply to work systems in general also apply to information systems in particular. This approach involves treatment of IS as a system, rather than a tool. System development often refers to developing software tools that meet requirements and satisfy the needs of users, rather than developing or modifying a work system in an organization. The tool view focuses on the form, affordances, and limitations of the interface. The system view includes those factors, but goes further in showing why the limiting factor is sometimes the person rather than the technology. This definition of IS is consistent with learning approach discussed in this paper because it includes both humans and technologies.

Before investigating the use of machine learning and inductive learning in IS, machine learning principles are outlined in next section.

3 Machine learning

Cios and Kurgan [5] define machine learning as the ability of a computer program to improve its own performance, based on the past experience, by generation of a new data structure that is different from an old one, like production rules from input data. The demand of machine learning applications, in particular in the areas of data, image and text mining, has created an urgent need for systems that can efficiently search for regularities or data descriptions in very large information sources [8]. Most typical applications of machine learning can be divided in three groups [21, 22]:

- Data mining in large data bases containing valuable implicit regularities that can be discovered automatically.
- Poorly understood problem domains where little knowledge exists to develop effective algorithm.
- Domains where programs must dynamically adopt to changing environment.

There is a wide range of methods to be used for machine learning [19, 17], e.g., artificial neural networks, Bayes classifier, K-Nearest Neighbours classifier, genetic algorithms, inductive learning, etc. Machine learning can be broadly classified into three categories: supervised learning, unsupervised learning and reinforcement learning [22]. The inductive learning is a special class of the supervised learning techniques. There are several more specific classifications of machine learning methods; classification varies depending on author and time period (e.g., different classifications are given in [22, 10]). Although machine learning algorithms are domain independent, in many domains generated descriptions or patterns need to have not only a high predictive accuracy, but also are required to be easy interpretable and comprehensible for user. Different applications may demand different description forms, i.e. reasoning system should be able to transform its results from one form to another. Inductive learning algorithms are preferable over other machine learning methods in systems where understanding of decision-making steps and further processing of results is needed. For instance, expert systems are systems where the rules induced by learning algorithms can be used [5]. Basic concepts of inductive learning, namely, the notion of inductive learning, overview of inductive learning methods, and general inductive learning system are discussed further in this section.

3.1 Inductive learning

Induction is a process of conversion of particular facts into general regularities. In computer science inductive learning is learning by example where a system tries to induce a general rule from a set of observed instances [12]. This involves classification - assigning the name of a class to every particular input. Classification is important to many problem solving tasks. Inductive learning methods are considered attractive for many real-life applications (e.g., medical diagnostic [3]), most due to their interpretability.

Inductive learning constructs a description of a function from a set of input/output examples. An example is a pair \((x, f(x))\), where \(x\) is the input and \(f(x)\) is the output of the function applied to \(x\). The task of induction is to return the hypothesis \(h\) that approximates \(f\) from a given set of examples [18]. Then, the generated hypothesis is applied to the new examples to predict their class membership [6].

Classification of machine learning methods isn’t strict. There is also no agreement among authors concerning the scope of methods belonging to inductive learning. In this paper with this name are denoted methods which can provide a classification for unseen instance not only on the basis of a given set of examples (as, e.g., K-Nearest Neighbours classifier does), but which also offer generalized model for classifying new instances. Such knowledge compression can be done by decision trees and related approaches (e.g., rule induction).

There are several dimensions along which learning algorithms can be classified. Depending on the way of learning, inductive learning methods can be divided in incremental and nonincremental (or static) ones. Other option to divide inductive methods is to consider the way the classifier is obtained and described – whether it forms decision tree, generates rules or combines both. For instance, the most popular algorithms in each category
are as follows [5, 17]: ID3, C4.5, and CART for decision tree, AQ for rules and CN2 for hybrid methods (quite often added to rules generating methods).

### 3.2 General inductive learning system

Generally classification task with inductive learning is organized as follows. First, the classifier for particular domain is formed; afterwards it is used for automatic or semi-automatic classification of new instance. Classifier formation consists of two parts, classifier training and testing, which is followed by applying it. General schema of inductive learning is proposed in Figure 1.

![Figure 1. Steps of classifier forming and using](image)

In the training phase an inductive learning method is used to infer description (either in form of decision tree or classification rules) from a given set of examples, where the class for every single record is known. Example set can be accumulated form observations, generated by expert or from both. The evaluation of description accuracy for unseen examples from the same domain follows. The class is assigned to every test example in accordance with description gained in training step. As the test example’s true class is known, one can rate the accuracy of predictions and the overall accuracy of the classifier.

### 4 The role of inductive learning in IS context

A software system is a part of IS. Machine learning and inductive learning methods are used especially in software systems. Zhang [21] points to two ways of machine learning usage in software systems. He states that inductive learning and other machine learning algorithms can be used for both (1) building tools for software development and maintenance tasks and (2) incorporation into software products to make them adaptive and self-configuring [21]. Considering IS development in more detailed way, „building tools for software development and maintenance tasks” can be divided in two more specific parts. Learning can refer to the IS development project management and to the development of particular IS that needs the results gained by inductive learning. So the division in three (instead of two) IS issues and situations of inductive learning use in the context of these issues is proposed.

1. **Inductive learning could be used in the IS development project management.**
   With the help of inductive learning models for software development process can be built. For example, software engineering data can be analyzed to predict software costs [22].

2. **Inductive learning could be used in particular IS development.**
   Some parts of IS may not be created without knowledge acquisition of problem domain, e.g. the initial rules base should be inferred to build it in the autonomous IS.

3. **Inductive learning could be applied in IS usage.**
   Inductive learning system could be implemented in the IS in order to let it learn from experience while it is working. An example of such a system is a practical diagnostic supporting workbench, which incrementally incorporates new competences with an existing knowledge base [3].

There are four fields (“learning issues” in the context of this paper) identified which are tightly connected with IS development or IS usage and involve inductive learning. They are knowledge acquisition, software development, expert systems, and complex system modeling. These learning issues are discussed in more detail in sections 4.1. – 4.4. They aren’t isolated as knowledge acquisition is not just an independent subject but also the base for inductive learning use in software development and expert systems. The overlapping between three IS issues and four learning issues is reflected in Figure 2. The references point to the sources where the use of inductive learning in particular learning issue in connection with the IS issue is mentioned.
Knowledge acquisition emerges in all IS issues showing the need for this field in all IS stages. The need of learning in expert systems can show up both in IS development stage (creation of rules to be embedded into the system) and in IS use (generation of new rules from experience).

4.1 Knowledge acquisition by inductive learning

Knowledge acquisition in IS context is most often connected with expert systems. The main reason for research on knowledge acquisition is to develop techniques for making expert systems easier to build and to make them more explainable, robust, and intelligent [14]. Knowledge engineering captures human knowledge and places it into a computer system where it is used to solve complex problems normally requiring a high level of human expertise.

Knowledge acquisition is recognized as one of the major problems in an expert system development. Knowledge elicitation from domain experts and machine learning are two distinct approaches to knowledge acquisition [15]. Knowledge takeover from human experts are both time consuming and expensive. In many domains the machine learning methods should be preferred because they are automated and more effective and efficient [15]. Comparative review of knowledge engineering and inductive learning is done also by Jackson [7]. Both knowledge elicitation from domain experts and machine learning are employed in medical domain – tonsillectomy/adenoidectomy patients are classified into normal and abnormal diagnostic groups with respect to their predispositions to bleeding.

While Oprea [15] and Jackson [7] separate manual knowledge engineering and automated machine learning methods, Langley and Simon [10] argues that machine learning may never entirely replace knowledge engineering as a framework for knowledge based systems. It only increases the levels of automation in the knowledge engineering process. Taki [14] proposes the idea that knowledge acquisition should be done by integrating manual and inductive (automated) methods, where human expert evaluates and corrects automatically obtained knowledge, if necessary.

In 1995 Langley and Simon [10] stated that inductive learning and other learning methods will become increasingly prevalent in progress toward automation in knowledge acquisition. Nowadays, the use of software tools to aid the acquisition process has increased, but none dares to state that knowledge acquisition has become an automated process.

4.2 Inductive learning in software development

Many software engineering tasks could be formulated as learning and classification problems [21]. Machine learning and data mining methods (including inductive learning) can be and are used in developing software products [20, 21]. Classification that can be carried out with the help of inductive learning methods is one of mining tasks in software development [20]. For machine learning applications the following areas of software development and maintenance are identified:

- requirement engineering (knowledge elicitation, prototyping), which overlaps with knowledge acquisition;
- software reuse (application generators);
- testing and validation;

![Figure 2. The use of inductive learning in IS context](image-url)
• maintenance (software understanding);
• project management (cost, effort, or defect prediction or estimation).

The strength of ML methods in software engineering lies in the fact that they have sound mathematical and logical justifications and they can be used to create and compile verifiable knowledge about the design and development of software artifacts [22].

Considering the use of inductive learning in software system development, particular applications from [22] report should be mentioned. In software quality prediction a decision tree based method is used to generate measurement based models of high-risk components. Software engineering data are analyzed to predict software costs. In maintenance task effort prediction, software resource analysis, correction cost estimation, and reusability prediction predictive models are built through decision tree induction. Inductive learning finds its application also in organization of reusable software engineering components by predicting the cost of rework. Since requirement acquisition is related to knowledge acquisition, where inductive learning methods take of great importance, inductive methods are used likewise in requirement elicitation. The observed situation proves that inductive learning approach has a stable position in software engineering process, most frequently dealing with the issue of how to build models or estimate certain property of software development process.

The use of data mining in software development has been described in a comprehensive review [20]. It shows up a practical classification tasks that are carried out for dynamic call graphs to determine the bug location in software and for text classification to assign bug reports to specific developer.

4.3 Inductive learning and expert systems

Expert systems are widely discussed in the literature. They are closely connected with knowledge acquisition as mentioned before. During knowledge-based system development the knowledge acquisition process is crucial and time consuming. Inductive learning methods can help in automating this process. Automation increases the speed and reduces the cost of development by reducing the amount of time needed from experts and knowledge engineers [16]. It could also be argued that, in some cases, automation may uncover knowledge that might otherwise be left unnoticed.

Diagnostic supporting tool (that is an expert system actually) described in [3] shows an example of expert system which use inductive learning facilities not only in development level of this system, but also in usage phase. Incremental decision tree is used to enrich systems knowledge-base with new data merging them into the existing tree model.

4.4 Complex system modeling

Inductive learning algorithms are also used for complex system modeling. The fundamentals of this approach are given by Madala and Ivakhenko in [11]. This monograph surveys new types of learning algorithms for modeling complex scientific systems in science and engineering.

One of complex tasks modelled by inductive learning is academic timetabling [13]. This task is complex due to dependencies on various rules and constraints [13]. Three inductive learning algorithms has been implemented and compared on timetabling for the purpose to see their ability towards solving complex problems and adapting to the new environment by changing rules. In building an optimal timetable system, a lot of rules, dependencies, and constraints are needed. Failure to identify that knowledge may cause the system to have an ineffective solution. The ability of inductive learning to learn and generalize rules is fully used in timetabling so that the system has its own competence to analyze the data and gather the needed information for proper functioning. The conclusions driven from real experiments with inductive learning algorithms ID3, AQ and ILA show that inductive learning algorithms can be successfully implemented into timetabling and complex problem domain overall [13].

4.5 Inductive learning methods used in practice

In Table 1 there are gathered particular inductive learning methods used in learning issues discussed previously in this section. This summary is not meant as complete listing of inductive learning algorithms that are used in these tasks. It just gives an insight to variety of inductive learning methods used in IS context.

<table>
<thead>
<tr>
<th>Knowledge acquisition</th>
<th>Software development</th>
<th>Expert systems</th>
<th>Complex system modeling</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID3 [15], ILA [15], AQ [14], CN2 [14]</td>
<td>Decision trees (not specified)</td>
<td>ID3 [21], C4.5 [21], Assistant [21]</td>
<td>ID3 [13], AQ [13], ILA</td>
</tr>
<tr>
<td>Decision trees</td>
<td>ID3 (Incremental Tree Induction) [3], CLIP [5]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- 99 -
5 The new frontiers

It is obvious that most optimistic papers about automation of IS development with the help of machine learning methods are tree to eighteen years old. In IS conferences in 2009 (AMCIS, PACIS, CAiSE, ISD) such an issue almost can’t be found.

In recent years machine learning applications find their place in e-commerce and similar Web related areas. One of such an area is interactivity between Web content producers and consumers [4, 23]. Customer reviews posted on the Web have grown significantly. Because customers represent the primary stakeholder group of a company, understanding customers’ concerns expressed in reviews could help marketers and business analysts to identify market trends and to provide better products and services [4]. The approach consists of the steps of sentence extraction, aspect identification, sentiment classification, and review summarization. Likewise Zhang [23] describes analysis relative to e-commerce involving mining the product reviews that thrive on the Web. Typical tasks include polarity prediction – distinguishing positive, negative, and neutral reviews – and opinion extraction. This new task in text sentiment analysis improves product review ranking services, helping shoppers and vendors leverage information from multiple sources. Different but closely related field to e-commerce is Question Answering Communities (such as Naver, Baidu Knows, and Yahoo!) [1]. By posting questions for other participants to answer, information seekers can obtain specific answers to their questions. Modeling information-seeker satisfaction uses machine learning approaches.

Only some of new challenges for automated learning tasks in information systems were mentioned above. It seems promising for inductive learning to find its place in dealing with such problems. This possibility has to be studied in detail.

To extend concepts shown in Figure 2, learning processes within e-commerce could be placed in IS usage issue (see Figure 3). This field is initiated with knowledge acquisition because knowledge acquisition is used to accumulate product reviews.

<table>
<thead>
<tr>
<th>IS issue</th>
<th>Learning issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>IS development project management</td>
<td>Knowledge acquisition [21] → Software development [21, 22]</td>
</tr>
</tbody>
</table>

Figure 3. Extended view to inductive learning applications in IS context

Figure 3 shows only areas of inductive learning application. To understand better the potential of inductive learning in these IS issues in the context of contemporary technological and economical environment, it is necessary to reveal and define workable characteristics of these IS issues relevant to inductive learning.

6 Conclusions

This paper summarizes the existing applications of inductive learning in IS development and usage. A new classification of the use of inductive learning in IS context is proposed. Inductive learning applicability in IS is divided in three issues: (1) inductive learning could be used in the IS development project management; (2) inductive learning could be used in development of particular IS; (3) inductive learning could be applied in the usage of IS.

There are four fields identified which are tightly connected with information system development or information system usage and involve inductive learning. They are knowledge acquisition, software development, expert systems, and complex system modeling. These fields aren’t isolated as knowledge acquisition is not just an independent subject but also the base for inductive learning use in software development and expert systems.
Future work includes revealing and specifying characteristics of IS issues defined in this paper to improve IS development and usage in the spheres related to machine learning and inductive learning.

7 Acknowledgement

This work has been supported by the European Social Fund within the project „Support for the implementation of doctoral studies at Riga Technical University”.

References

TAKAGI-SUGENO REASONING PROCEDURE FOR PATTERN RECOGNITION

Raimundas Jasinevicius\textsuperscript{1}, Vytautas Petrauskas\textsuperscript{1}, Radvile Krusinsiene\textsuperscript{2}

\textsuperscript{1}Kaunas University of Technology, Studentu 50-204A, Kaunas, Lithuania, 
raimundas.jasinevicius@ktu.lt, vytautas.petrauskas@ktu.lt

\textsuperscript{2}Lithuanian Academy of Physical Education, Department of Informatics and Languages, 
Sporto 6, Kaunas, Lithuania; r.krusinskiene@lkka.lt

Abstract. The problem of pattern recognition is well-known as well as the reasoning procedure of Takagi-Sugeno (T-S), but up till now both of them have not been used together to describe solutions of the same fuzzy problems. Therefore, the paper analyses the pattern recognition process as Takagi-Sugeno reasoning procedure and defines (T-S) rules weights, solving special linear programming problem, which is constructed according to experts’ fuzzy knowledge. Such an approach for pattern recognition problem has been used for the first time. A practical, comprehensible and usable theory enabling practitioners to build adequate hardware/software tools for pattern recognition is presented in the article.

Keywords: pattern recognition, Takagi-Sugeno reasoning, fuzzy rule, linear programming problem.

1 Introduction

The pattern recognition problem is old enough, very well-known and perfectly described in scientific literature. The work of R. O. Duda and others \cite{1} may be described as one of the highly recommended texts concerning this topic and including an extremely good list of references. In general, a formulation of this problem itself implies a pretty good amount of fuzziness. According to \cite{2}, pattern recognition is defined as a search for structure in data, which is performed in three steps: data acquisition, the extraction of pattern features from data by the reduction of dimensionality and the mapping of extracted features to pattern classes. The main problem, related with the issue, is connected with techniques aiming to cope with the problem of fuzziness of expert knowledge.

There are three main goals of this article.

1. Considering the fuzziness of the description of pattern recognition problem in general, to use for the first time Takagi-Sugeno (T-S) fuzzy reasoning procedure, which is known and successfully used in other applications.

2. To propose to apply solutions of specially formulated linear programming problem (LPP) for calculation of weights in (T-S) reasoning procedure instead of using time consuming and poorly determined neural-type training processes. It is considered that formulation of LPP itself, using expert’s knowledge, is more adequate to the fuzzy description of the pattern recognition problem.

3. To present a practical, comprehensible and usable theory enabling practitioners to build adequate hardware/software tools for pattern recognition.

Successful implementations of the proposed theory in different fields will be summarized and presented in the next article of the authors. The inference of ordinary fuzzy systems is based on: 1) a deriving verbal (linguistic) or parametric consequents by pre-processing lists of fuzzy rules that contain verbal or parametric antecedents linked by certain fuzzy logic operations, and 2) a defuzzification process, applying some compositional rule or formulae \cite{2}, \cite{3}.

The types of rules are:

\begin{align}
\text{IF} \ x \ \text{is} \ A \ \text{AND} \ y \ \text{is} \ B \ \text{THEN} \ z \ \text{is} \ C \ & \text{ (for Mamdani fuzzy models)} \\
\text{IF} \ x \ \text{is} \ A \ \text{AND} \ y \ \text{is} \ B \ \text{THEN} \ z = F(x,y) \ & \text{ (for T–S fuzzy models)}
\end{align}

Defuzzification procedures for the two cases mentioned above can be described as reasoning on the basis of a set of consequents $C$ using the centre of gravity (CoG) or a mean of maximum (MoM) methods for Mamdani type models, and a fuzzy mean (FM) method as reasoning by evaluation of all results $z$ included and processed according to the certain formula $\Phi(z)$ for T-S models.

2 Pattern Recognition Based on T-S Reasoning Procedure

A pattern of a class is considered (as a physical or abstract structure of class’s objects) described by a set of distinctive features as described in \cite{1} and \cite{2}. The pattern recognition problem can be formulated as follows. It is considered in the article that $p = 1, 2, \cdots, S$ classes of objects and each class $p$ has its own pattern. Each object is described by $n = 1, 2, \cdots, i, j, \cdots, N$ features. After the extraction of a feature as well as the procedures of measurement and normalization as described in \cite{4}, the i-th feature of an object that belongs to
the p-th class (i.e. corresponding to the p-th pattern) can be represented by a real number \( \alpha_{pi} \), which expresses a degree of intensity of this particular feature, and the object is represented by a vector-row 
\[ \vec{\alpha}^p = \left( \alpha_{p1}, \alpha_{p2}, \ldots, \alpha_{pi}, \ldots, \alpha_{pN} \right) \]. If there are \( l = 1, 2, \ldots, k, \ldots, L \) objects and their dependence to class \( p \) (i.e. they originate from the p-th pattern) is known in advance, class \( p \) is represented by a set of vectors 
\[ \vec{\alpha}^p_l, \forall p, l \].

The main task of the pattern recognition procedure involves the development of T-S rules and defuzzification instruments. Rules and instruments must be defined applying all available information about patterns, which is stored in sets of \( \vec{\alpha}^p_l, \forall p, l \) and experts’ experience that is presented in a verbal form and was collected when working with features of objects and patterns. Such reasoning allows constructing an instrument of pattern recognition providing a possibility to assign any unknown but properly described object \( \vec{x} \) to any of possible classes (and patterns) \( p \). An accuracy of the assignment depends on the instrument’s decision making efficiency to process fuzzy information.

Frequently, better reasoning results are achieved when features of objects are normalized and centred [4], [5]. Then the object is represented as a vector-row 
\[ \vec{\alpha}^{ol}_p = \left( \alpha_{olp1}, \alpha_{olp2}, \ldots, \alpha_{olpi}, \ldots, \alpha_{olpN} \right) \] the components of which are calculated as follows:
\[ \alpha_{olp} = \alpha_{pi} - \frac{1}{N} \sum_{j=1}^{N} \alpha_{pj} \] (2)

Therefore, T-S reasoning procedure can be expressed using Eq (1). It means that a set of positive rules (6) for T-S pattern recognition procedure consists of a list of statements:

**IF** < degree of certainty that feature \( i \) with intensity \( x_i \) belongs to the pattern \( p \) is \( K_{pi}> \) **THEN** < \( z_i^+ = K_{pi} x_i \) **RECOMMENDED**

It means:
\[ IF < \mu^+(x_i^p) = K_{pi} > THEN z_i^+ = K_{pi} x_i^p, \forall p, i \] (3)

Similarly, a set of negative rules (6) for T-S pattern recognition procedure consists of a list of statements:

**IF** < degree of certainty that feature \( i \) with intensity \( x_i \) belongs to any other pattern except \( p \) is \( 1-K_{pi}> \) **THEN** < \( z_i^- = -K_{pi} x_i \) **NOT RECOMMENDED**

This means:
\[ IF < \mu^-(x_i^p) = (1 - K_{pi}) > THEN z_i^- = -K_{pi} x_i^p, \forall p, i \] (4)

According to the concept of hyper-inference (6)
\[ \mu(x_i^p) = \max \{ \mu^+(x_i^p), \left(1 - \mu^-(x_i^p)\right) \} \] (5)

and
\[ z_i = z_i^+ - z_i^- = K_{pi} x_i^o, \forall p, i . \] (6)

When the unknown object \( \vec{x}^o \) is considered, its degree of belonging to class \( p \) can be evaluated by a dependence function:
\[ \Phi_p (\vec{x}^o) = \sum_{i=1}^{N} x_i^o K_{pi}, \forall p \] (7)

This defuzzification method for T-S procedure in a vector notation is expressed as:
\[ \Phi_p (\vec{x}^o) = \vec{x}^o K_p^T, \forall p , \] (8)

where \( T \) stands for a transposition of a certainty vector
\[ \vec{K}_p = \left( K_{p1}, K_{p2}, \ldots, K_{pi}, \ldots, K_{pN} \right) \] (9)

A block diagram, representing a final decision making act of fuzzy pattern recognition based on T-S reasoning procedure, is shown in Figure 1.
In practice, there is no possibility to formulate verbally understandable lists of rules for sets presented by Eq. (3) and (4). Usually neural-type training procedures based on gradient methods are used [1]. An intention to avoid comparatively unpredictable and very clumsy neural-type training procedures for the determination of certainty vectors $\bar{K}_p, \forall p$ in T-S reasoning procedure leads to a formulation of an objective function maximization problem, subjected to a set of constrains and constructed according to fuzzy information. In the next section of this paper a special linear programming problem (LPP) is proposed to find optimal certainty vectors $\bar{K}_p, \forall p$ for (3)-(6). LPP problem is formulated according to experts’ fuzzy information which enables to define rule weights in T-S reasoning based pattern recognition procedure.

3 Linear Programming Problem for Determination of T-S Procedure Rule Weights

A structure of Eq. (7) implies a simple (linear) form of an objective function to be maximized as well as linearity of constraints. According to the description of pattern recognition problem presented in section 2 of this paper, information about the patterns is stored in the sets of $\alpha^p_{\alpha v}$. In spite of its fuzziness, the formulation of a problem for the determination of certainty vectors $K_{pi}, \forall i$ for T-S reasoning procedure can be constructed as follows. One object $\alpha^ok_{p}$, which represents class $p$ and is central, is randomly selected and coordinates of certainty vector $K_{pi}, \forall i$ are found in order the dependence function $\Phi_p(\alpha^ok_{p})$ would be maximized:

$$\Phi_p(\alpha^ok_{p}) = \sum_{i=1}^{N} \alpha^ok_{pi} K_{pi} \rightarrow \text{max}$$

under the following constrains:

$$\sum_{i=1}^{N} \alpha^pi_{\alpha v} K_{pi} \geq \gamma \sum_{i=1}^{N} \alpha^ok_{pi} K_{pi}, \forall l$$

$$\sum_{i=1}^{N} \alpha^ri_{\alpha v} K_{pi} \leq \kappa \sum_{i=1}^{N} \alpha^ok_{pi} K_{pi}, \forall r \neq p, \forall l.$$  (12)

Eq. (11) is tightly connected with the concept of “positive similarities” within the class and extracts them using the set of positive rules, as it was described in section 2. Eq. (12) reflects dissimilarities between certain classes (in this particular case class $p$) and all other classes (or “negative similarities”) and extracts those dissimilarities by the set of negative rules. Optimal values of $\gamma$ and $\kappa$ are recommended from the interval [0-1], and $\gamma > \kappa$ [4]. Particular values depend on the prior knowledge (or guess) concerning the structure (internal connections and dispersion of patterns’ features) of classes (or patterns). The difference $\gamma - \kappa$ determines experts’ fuzzy assumption concerning a possible structure of classes under investigation and must be chosen on the basis of some experience [4]. Coefficients $\gamma$ and $\kappa$ allow controlling fuzzy level of those similarities and dissimilarities.
As it can be easily noticed the problem belongs to the class of LPP where inequalities (11) and (12) need additional constrains:

\[ 0 \leq K_{pi} \leq A, \forall i, \]

(13)

where \( A \) is any practically convenient real number.

A solution of Eq. (10)-(13) for the class \( p \) consists of the obtained value of

\[ \max \Phi_p (\alpha_p^k) = \Phi_p \max, \]

\[ \bar{K}_p = (K_{p1}, K_{p2}, \ldots, K_{pi}, \ldots, K_{pN}) \]

(14)

The procedure must be repeated for all classes \( p \). In this way a set of \( S \) solutions will be obtained and the recognition procedure must be performed according to the Figure 3 taking into account the need of fulfilling a condition of proportionality:

\[ c_1 \Phi_1 \max = \ldots = c_p \Phi_p \max = \ldots = c_s \Phi_s \max = B, \]

(15)

where \( c_i \) are real numbers.

Eq. (15) plays a role of normalization and it means that the fuzzy term very similar must be evaluated by the same number \( B \) whatever pattern is considered.

As it was delivered before, T-S reasoning procedure presents degrees of certainty, indicating that the description of an unknown object \( \bar{x}^o \) belongs to pattern \( p \) (see Figure 1). If there is a need to use this information as a recommendation for a decision maker, the value \( u_p = \Phi_p \left( \frac{\bar{x}^o}{\max(\bar{x}^o)} \right) \) must be determined.

Practically an interval for \( u_p \) is [0-1] with some zone of accuracy as it is shown in the Figure 2, where the degree of certainty \( \mu_p(\bar{x}^o) \) is presented.

![Figure 2. Recommended certainty that \( \bar{x}^o \) belongs to \( p \)](image)

When \( u_p \) is in the zone, special additional investigations of the object’s properties are strongly recommended before the final decision is taken.

The entire classification process is shown in Figure 3 and represents the process of pattern recognition combined with decision making recommendations.
4 Final remarks

In the analytical part of this paper a new approach to the problem of fuzzy pattern recognition was presented. First of all the rule-based fuzzy inference, concerning the measure of patterns’ similarity, was introduced, then a description of the process of pattern recognition was based on T-S reasoning procedure, enhanced by specific decision making recommendations. The weights of rules in T-S procedure were defined, solving special LPP, constructed according to the fuzzy information presented or supposed in advance.

A practical, comprehensible and usable theory, enabling practitioners to build adequate hardware/software tools for pattern recognition is presented in the article. Successful implementations of the proposed theory in different fields will be summarized and presented in the next article of the authors of this paper.

5 Acknowledgements

The authors of the article want to express their gratitude for the stimulating ideas to carry out this investigation by prof. Christian Borgelt from the European Centre for Soft Computing (in Mieres, Asturia, Spain).

This research was in part supported by COST programme and financial funding received from Lithuanian Agency for International Science and Technology Programmes.

References


FAST FINGERPRINT IMAGE SYNTHESIS

Andrej Kisel

Vilnius University, Faculty of Mathematics and Informatics, Department of Computer Science
II, Naugarduko 24 Vilnius LT-03225 Lithuania, kisel.andrej@gmail.com

Abstract. This paper presents a fingerprint synthesis method that can generate a fingerprint with predefined minutiae points. Fingerprint type is chosen randomly and singular points positions and quantities are chosen randomly according to the fingerprint type. Orientation map is generated using fingerprint orientation model. Frequency map is generated. Initial image with drawn minutiae points that are oriented by orientation map is constructed. Iterative filtering of the initial image with Gabor filters that are oriented using orientation map and constructed using frequency map produces fingerprint image with minutiae points located at the predefined positions. An optimization of the iterative filtering is described. Synthetic fingerprint images are used to evaluate extraction algorithm's stability to noise. A measure of extraction algorithm’s robustness to noised fingerprint images is proposed.

Keywords: fingerprint synthesis, fingerprints, performance evaluation, quality measure

1 Introduction

Much attention is being paid to biometric technologies such as person identification by face, fingerprints, eye iris or voice. Fingerprint identification is one of the most popular ways to identify a person and research in this area of biometrics is done in universities and companies.

Identification process consists of fingerprint image acquisition, feature extraction and feature matching. Different methods to evaluate algorithm performance are proposed [1], [2], [3], [4] and most of them use fingerprint databases such as NIST SD4 [5] or NIST SD14 [6] to calculate accuracy. Fingerprint features are extracted in enrollment process [7, 8] and then features are matched against each other in matching phase [9], [10], [11]. Receiver Operating Characteristic (ROC), Detection Error Trade-off (DET) or other statistics [12] may be calculated to analyse extractor or matcher performance. Fingerprint verification competitions [13], [14] have been arranged to analyze and benchmark commercial and academic algorithms. The biggest and most thorough of them was NIST arranged competition MINEX [15]: vendors could send their extraction or matching algorithms and best extractors and matchers were selected. Majority of vendors send both algorithms and it is interesting, that in many cases matching algorithms performed better with extractors from the same vendors (it can be seen from scenario 1 in MINEX report [15]). It can be easily explained, since many vendors develop both matcher and extractor and they know about typical problems of their extractors and can compensate for them in matching phase. It is hard to develop an accurate extraction algorithm because it is hard to evaluate it without a good matcher and since most vendors develop both algorithms, they are not sure that even if performance of their matcher or extractor is good enough, it will be good when used with other extractor (or matcher).

The estimation of biometric algorithms performance expressed in ROC or DET curves have several disadvantages: the result depends on database of fingerprint images, quality of extraction and quality of matching algorithm. To have the possibility to estimate extractor and database quality separately from matching routine, we propose to utilize synthesized fingerprints images. They can serve as a reference or as an ideal database that allows introducing some quantitative quality measures for estimation of extractor’s performance for a particular database. If the same extractor is applied on several fingerprint image databases, the database quality can be associated with the proposed quality measure. The situation is similar to situation when the quality of several extraction algorithms can be compared if same database for each extractor is used to calculate statistics.

The ROC and DET characteristics of extracting algorithm can be replaced (or completed) by a quality measure that accounts information about exact positions, types and orientations of minutiae. To have fingerprints images with predefined minutiae points, we extend SFINGE fingerprint image synthesis method [16].

2 Fingerprint synthesis

Available synthesis methods were analyzed [16], [17], [18] and SFINGE [16] was chosen as a base method because it is well described and its ability to generate finger-like images was tested in fingerprint verification competitions (FVC2000 [1] FVC2002 [13] FVC2004 [14] and FVC2006), in which one of four databases was generated synthetically. Fingerprint images generated with SFINGE look like real ones, and identification algorithms performance is similar to performance obtained on real fingerprints. What is more important, SFINGE ability to generate high-quality fingerprints was validated in [24].

SFINGE (Synthetic Fingerprint Generation) consists of several steps: Fingerprint form generation, fingerprint type and orientation map generation, density map generation, ridge generation.
2.1 Fingerprint form

Fingerprint generation is starting by fingerprint form determination. Fingerprint form can be described using following methods: it can be described by an ellipse, polygon, or by a square with rounded corners. SFINGE method uses five coefficients[16], which can be generated randomly, chosen manually or derived from the real fingerprint (Fig.1).

![Fingerprint form](image1)

Figure 1. Five coefficients that describe fingerprint form (left) and fingerprints with corresponding forms (right)

2.2 Orientation map and fingerprint type

These properties are strongly related, because fingerprint type depends on positions of singular points (loops and deltas), and positions of singular points depend on orientation map. Different methods of orientation map generation are described in literature [19], [20] and any of them can be used to generate fingerprint orientation map. Sherlock-Monroe [19] method was chosen in this article because of its simplicity. Fingerprint orientation map is calculated in following steps:

- Fingerprint type is selected (manually or randomly);
- Quantities of singular points are selected depending on fingerprint type. Positions of all singular points (loops and deltas) are selected (randomly or manually) with restrictions that depend on chosen type.

Having loops and deltas quantities and positions, orientation map is generated by the following formula:

\[
\Theta = \frac{1}{2} \sum_{i=1}^{N_d} \text{arg}(z - d_{s_i}) - \sum_{j=1}^{N_c} \text{arg}(z - l_{s_j})
\]  

(1)

Where \(z\) – complex number made from (x, y) coordinates of point, in which orientation is calculated;

\(N_d\) – number of delta type singular points; \(N_c\) – number of core type singular points; \(d_{s_i}\) – complex number made from \(i\)-delta coordinates; \(l_{s_j}\) – complex number made from \(j\)-loop coordinates; \(\text{arg}\) – complex number argument. Orientation is calculated in each pixel of fingerprint image (Fig.2).

![Orientation map](image2)

Figure 2. Example of orientation maps for different type fingerprints

2.3 Ridge density map generation

Ridge density map is generated using following information about fingerprint characteristics: Default distance between ridges is 9 pixels (here and below we assume that scanner’s resolution is 500 pixels-per-inch (dpi)), ridge frequency is lower on the top of the image, and lower on the bottom [21].

2.4 Ridge generation

Ridges are generated by iterative filtering of the blank image (initial image) with random dots, by Gabor filter [22] that is created using orientation and density from orientation and density maps:

- Black image with white dots in random positions is generated
Image is filtered several times with spatial (Gabor) filter (Fig. 3), which has orientation and frequency properties. Filter is generated by the following formula:

\[
h(x, y, \phi, f) = \exp\left(-\frac{1}{2} \left( \frac{x^2}{\delta^2_x} + \frac{y^2}{\delta^2_y} \right) \right) \cos(2 \pi f x),
\]

\[
x_\phi = x \cos \phi + y \sin \phi,
\]

\[
y_\phi = -x \sin \phi + y \cos \phi
\]

Where \(\theta\) – orientation, \(f\) – frequency (density), \(\delta_x = \delta_y\) – chosen empirically.

Orientation and frequency values are taken from orientation and frequency maps. Filter is applied to entire initial image, and after several iterations random dots begin to grow into the lines and lines begin to form fingerprint ridges. Ridges fill the image and minutiae points (ends and bifurcations) appear (Fig. 4).

2.5 Analysis

An algorithm was implemented, and after experiments and research it was modified to generate minutiae points not in random positions, but in predefined ones. The following section describes a method that is fast and can generate images with minutiae in given positions.

3 Modified SFINGE Method

Since method is based on SFINGE method, steps like orientations map calculation and filtration are not described here once more. This section is focused on the difference between original and modified method.

Main steps of fingerprint generation are: fingerprint type is chosen (randomly or manually); orientations are generated; gabor filters of orientations that present in orientations image are generated before filtering to speed up generation.

Initial image is constructed in the following order: zero valued image of given size is constructed; coordinates (positions) and types of minutiae points are generated randomly or selected manually; small images of minutiae are drawn on the initial image in the selected positions so that minutiae orientations are aligned with orientation map. There are two types of minutiae points – line ends and bifurcations. Since these types are invertible (line end is a bifurcation on the inverted image), a bifurcation is drawn using positive (+1) value pixels, and line end is drawn using negative (-1) value pixels.
It is already possible to generate a fingerprint image by filtering initial image with Gabor filters that are oriented by orientation image (Fig.5), but since responses of Gabor filters are calculated in every pixel, it is a slow operation. An improvement was implemented to perform iterative filtering only in those pixels that are required in current iteration to generate a fingerprint. The main idea of the improvement is to start filtering from positions of drawn minutiae and near it, and to extend filtering area until entire image is generated. For example, if Gabor filter is 10 pixels wide, then in the first iteration pixels that are from 0 to 10 pixels away from the drawn minutiae points are filtered, in the second iteration – pixels that are from 1 to 11 pixels away from drawn minutiae points are filtered, in third iteration – pixels that are from 2 to 12 pixels away from drawn minutiae points are filtered, in fourth iteration – from 3 to 13 pixels away, and so on, until the fingerprint is generated (no pixels are left to filter). To perform fast calculation of distances to minutiae points, Euclidean Distance Map (EDM) [23] is calculated from initial image. The EDM map is an image, and value of each pixel indicates the distance to the nearest object (drawn minutiae point). The order of filtering can be easily calculated from EDM. In the first filtering iteration only pixels that have values from 0 to 10 in EDM are filtered, in second filtering iteration only pixels with values from 1 to 11 in Euclidean Distance Map are filtered in initial image, in third iteration only pixels with values from 2 to 12 are filtered, in fourth iteration – from 3 to 13 and so on. Fig.6 shows an example of five iterations.

Filtering is performed by applying Gabor filter (orientation $\theta$ is taken from orientation map, frequency $f$ is taken from the frequency map) to each pixel of the initial image in the order that is calculated from EDM.

Generated image can be noised or transformed after generation to provide it more natural look.

Described optimization speeds up generation process more than 5 times.

Resulting generation speed is more than 2 fingerprints per second on Intel Core 2 Duo 6600 processor.

It can be noticed that though some additional minutiae points appear in result image, all initial minutiae points are present and location, direction and type of them are the same as in the initial image.

Similar synthesis approach was independently used in [24] with the aim to investigate the possibility of fingerprint reconstruction from standard template. The main differences in generation process are summarized in the following table (Table 1):

<table>
<thead>
<tr>
<th>Proposed method</th>
<th>Method described in [24]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fingerprint form is described by 5 coefficients that are generated randomly</td>
<td>Fingerprint form is approximated from minutiae positions</td>
</tr>
<tr>
<td>Fingerprint type is selected randomly, singular points are selected randomly with restrictions of chosen type, orientations are generated using fingerprint orientations model</td>
<td>Orientations are derived from template</td>
</tr>
<tr>
<td>Pixels of initial image can have values [-1, 0, 1]. In the beginning the images consists of zeroes. In the places where bifurcations should be generated, thin lines that represent bifurcations are drawn using positive (1) values. In the places where bifurcations are generated, thin lines that represent bifurcations are drawn using positive (1) values.</td>
<td>Initial image is binary and constructed using minutiae prototypes that look like small raster images of minutiae points cropped from real fingerprint image</td>
</tr>
</tbody>
</table>
**Proposed method**

where line ends should appear, thin lines that represent bifurcations are drawn using negative (-1) values.

**Method described in [24]**

Fingerprint image generation is done by filtering of the initial image with Gabor filters in the order specified by Euclidean Distance map.

Fingerprint image generation is done by iterative filtering of the initial image with Gabor filters, until fingerprint is generated.

### 4 Extraction algorithm performance evaluation

The aim of the extraction algorithm (extractor) is to extract unique fingerprint characteristics and properties such as minutiae point’s locations, orientations and types, fingerprint type estimation and so on. Most extraction algorithms (extractors) consist of several image processing algorithms such as image normalization, texture orientation estimation, binarization, skeletonization and ridge frequency analysis. Developers of algorithms have to overcome many problems: source image can be noisy, fingerprint may be with scars and other imperfections, errors made in one extraction step can strongly affect other step, so it is necessary to have some way to evaluate extractor performance and find what errors can be fixed and which steps need more attention. It would be very useful to have some reference fingerprints, with known characteristics (Fig.7). The output of the extractor could be compared to known characteristics and errors could be calculated. Such task could be done by manual analysis of several real fingerprints with marking all the necessary properties such as minutiae points, fingerprint type, singular points locations and so on, but it is a very time consuming operation and the resulting data will be non-objective (results will depend on person). For example – one fingerprint can have more than hundred minutiae points, and each minutiae point have such characteristics as: location, direction, and type. To evaluate extractor’s performance to a good degree of accuracy one may need to repeat this procedure hundreds or thousands times so it becomes obvious that such work cannot be done manually.

Fingerprint image synthesis can be used to evaluate extractor’s accuracy. For example, many extractors perform orientation map estimation and since orientation map in synthetic fingerprint is generated by a known mathematical model, exact orientation in every pixel is known. It can be compared to an estimated orientation and error can be calculated. Fingerprint type, ridge frequency, minutiae point’s locations, types and directions are all predefined and can be used to evaluate extractor’s accuracy. It is possible to generate thousands of fingerprints and evaluate extractor’s performance with unprecedented accuracy. Extractor’s stability to noise, deformations or scars can also be evaluated by adding artificial noise to generated images and then initial data can be compared to the output of the extractor.

![Figure 7. Orients map (left), predefined minutiae (center), generated fingerprint (right)](image)

### 5 Experiments

To illustrate, how synthetic images can be used to evaluate extractor’s performance, an experiment was done. The method was used to analyze how artificial noise affects extractor’s results. Extractor from NIST certified VeriFinger SDK [25] was chosen.

Three databases were prepared: DB1 - 1000 synthetic fingerprint images with 50 initial minutiae points per fingerprint; DB2 is a publicly available database „DB1A“ from FVC2004 [14] competition (high quality fingerprint images from FBI certified scanner); DB3 is a publicly available database „NIST SD29“ [26] - WSQ compressed fingerprint images from scanned fingerprint cards (only plain fingerprints were used)

The resolution of images in DB2 and DB3 is 500 dpi; resolution of synthetic fingerprint images is about 500 dpi.

Images were noised with shot noise (‘salt and pepper’ noise) - some percent of all image pixels were set white or black. Positions of noised pixels were chosen randomly (Fig.8).
Minutiae points from noised images were extracted and following statistics for each noise density $d$ (percentage of the noised pixels) from 0 to 100 were calculated:

a) Percent of initial minutiae points that was found on noised image (Minutiae point was considered as found if it is the nearest minutiae point to the initial minutiae point and the distance to it is less than 10 pixels

b) How found minutiae positions were affected by noise (the average distance in pixels between initial minutiae and found minutiae)

Since accurate minutiae positions on real databases are not known, minutiae points extracted from not noised images were considered as initial minutiae points. The results are presented in the following figures:

It can be noticed from the Fig.9 that on DB1 more than 90% of minutiae points are detected up to noise densities around 60% (when more than half of image pixels are corrupted) and the average distance to initial minutiae point is about 0.5 pixels (Fig.10).

On DB2 more than 90% of minutiae are detected if noise density is less than 20% (Fig. 9), but the average distance to original minutiae point is still less than one pixel (Fig. 10).

On DB3 more than 90% of minutiae are detected only if noise density is less than 5% (Fig.9). At noise density 20% only 60% of minutiae are found and the average distance to original minutiae point is more than one pixel (Fig.10).

Real fingerprint images have lower signal-to-noise ratio because some noise is added during fingerprint acquisition process (dust on scanner, marks from previously scanned fingerprints, digitization errors, image compression artifacts)

We propose to use synthesized image database as references in estimation of quality of the extractor. For example curve of Fig 9 that corresponds to synthetic DB1 is the most right. We postulate that this curve
corresponds for ideal scanner or ideal database and has 100% quality. The measure of quality can be defined as 100 * S1 / S1, where S1 is an area below the graph in Fig 9 that corresponds to DB1. Quality of DB2 and DB3 databases or their corresponding scanners is 100 * S2 / S1 and 100 * S2 / S1. Here S2 and S3 are areas below the corresponding graphs in Fig.9 After calculation of the areas, quantitative extractor quality on DB2 and DB3 was calculated. The numbers are 64.7% for DB2 and 35.3% for DB3. Graphs in Fig. 9 represent proportion of reliable detected minutiae in respect of added “salt and pepper” noise. Thus the calculated qualities represent extractor quality for the two databases in the sense of quantity of reliably detected minutiae.

Graphs in Fig. 10 can be used in the similar manner to estimate precision of extracted minutiae positions. Average error on DB2 is 7.8 times higher than on DB1, on DB3 – 11.1 times higher than on DB1. Note that in both cases extraction qualities for DB3 were lower and this well correlates with published ROC and DET characteristics for these databases [14], [27].

6 Conclusion

A method of fingerprint image generation is described in detail. The method is fast, and can generate images with minutiae points that have predefined properties such as location, direction and type. It can be used not only to generate large databases of fingerprints, but to precisely evaluate extraction algorithm performance and detect its weak sides (stability to noise, scratches, deformations and other imperfections). Experiments show that quality of generated images is much better than quality of real fingerprints and since initial minutiae positions are known, the accuracy of evaluation is higher.

It is common to evaluate database, extractor or matcher accuracy by calculating statistics like ROC on some database, but the result of such evaluation depends on all three components – quality of database, quality of extractor and quality of matcher. When synthetic database is used, evaluation results depend on only two components – quality of extractor and quality of database. Influence of the matcher is eliminated by using synthetic database as a reference. Two measures for extractor’s performance estimation were introduced. The resulting quality is in the range of 0 and 100%, where 100% corresponds to the case of synthetic images. The obtained numeric values of the qualities correspond to known ROC characteristics of test databases.

7 Future Work

The method will be used to design an extraction algorithm that will be more tolerant to scanner noise, fingerprint scratches and deformations. Since additional minutiae points appear in result image during generation process, additional work is needed to overcome this issue.

References


- 113 -


