Editor **DANA SIMIAN**

MODELLING AND DEVELOPMENT OF INTELLIGENT SYSTEMS

Proceedings of the Third International Conference on
MODELLING AND DEVELOPMENT OF INTELLIGENT SYSTEMS

October 10-12, 2013, Sibiu, ROMANIA

Lucian Blaga University Press, 2014

Editor DANA SIMIAN

MODELLING AND DEVELOPMENT OF INTELLIGENT SYSTEMS

Proceeding of the Third International Conference
"Modelling and Development of Intelligent Systems"
October 10-12, 2013, Sibiu, ROMANIA

Lucian Blaga University Press, Sibiu

2014

Lucian Blaga University Press, Sibiu, 2014

Editor Dana Simian

All papers in this volume were peer review by two independent reviewers

ISSN 2067-3965

Associate editor Laura Florentina Stoica
Cover design Ralf Fabian

Proceedings of the Third International Conference
Modelling and Development of Intelligent Systems
October 10 - 12, 2013, Sibiu, ROMANIA

Preface

This volume contains refereed papers which were presented at the Third International Conference Modelling and Development of Intelligent Systems. The conference was held between October 10 - October 12, 2013, at the Faculty of Sciences, "Lucian Blaga" University of Sibiu, Romania.

MDIS conference provides an opportunity for sharing ideas and establishing scientific cooperation in the field of intelligent systems. It aims to bring together scientists, researchers, students, interested and working in fields which can be connected with modeling and development of intelligent systems. Specific topics of the conference includes but are not restricted to: evolutionary algorithms, evolutionary computing, genetic algorithms and their applications, grid computing and clustering, data mining, ontology engineering, intelligent systems for decision support, knowledge based systems, pattern recognition and model checking, motion recognition, e-learning, hybrid computation for artificial vision, knowledge reasoning for artificial vision, geometric modelling and spatial reasoning, modelling and optimization of dynamic systems, large scale optimization techniques, adaptive systems, multiagent systems, swarm intelligence, metaheuristics and applications, machine learning, self learning algorithms, mathematical models for development of intelligent systems. The talks were delivered by universities' members, researchers and students from 10 countries (Bulgaria, Germany, Greece, Republic of Moldova, Romania, Serbia, Syria, Switzerland, Ukraine and USA). During the conference a wide range of theoretical and practical problems related to the conference topics were discussed. The plenary speakers addressed some of the most actual issues in the conference interest field:

- Prof. PhD. George Elefterakis - *The Challenge of Emergent Phenomena in Emergent Applications Software* - College of the University of Sheffield, Computer Science Department, City Liberal Studies, Thessaloniki, Greece
- Prof. PhD. Milan Tuba - *Ant colony optimization pheromone correction strategies* – Megatrend, University of Belgrade
- Prof. PhD. Ioana Moisil - *Scheduling Algorithms. A review* - "Hermann Oberth" Faculty of Engineering, "Lucian Blaga" University of Sibiu.

We thank all the participants for their interesting talks and discussions. We also thank the members of the scientific committee for their help in reviewing the submitted papers and for their contributions to the scientific success of the conference and to the quality of this proceedings volume.

December 2013

Dana Simian
Conference chairman

Scientific Committee

Prof. PhD **Kiril Alexiev**, Bulgarian Academy of Sciences, Bulgaria

Prof. PhD **Octavian Agratini**, Babes-Bolyai University of Cluj Napoca, Romania

Assoc. Prof. PhD **Alina Barbulescu**, Ovidius University of Constanta, Romania

Prof. PhD **Florian Boian**, Babes-Bolyai University of Cluj Napoca, Romania

Prof. PhD **Virgil Chicernea**, Romano-Germana University, Bucuresti

Assoc. Prof. PhD **Ioana Chiorean**, Babes-Bolyai University of Cluj Napoca, Romania

Researcher **Camelia Chira**, Babes-Bolyai University of Cluj Napoca, Romania

Prof. PhD **Domenico Consoli**, Urbino University, Italy

Assoc. Prof. PhD **Daniela Danciulescu**, University of Craiova, Romania

Prof. PhD. **Oleksandr Dorokhov** - Kharkiv National University of Economics, Ukraine

Lecturer PhD **George Eleftherakis**, International Faculty of the University of Sheffield, Greece

Lecturer **Ralf Fabian**, "Lucian Blaga" University of Sibiu, Romania

Assoc. Prof. PhD **Stefka Fidanova**, Institute for Parallel Processing, Bulgarian Academy of Sciences, Bulgaria

Prof. PhD **Vasile Georgescu**, University of Craiova, Romania

Prof. PhD **Dejan Gjorgjevik**, Faculty of Electrical Engineering and Information Technologies in Skopje, Macedonia

Prof. PhD **Heiner Gonska**, Duissburg – Essen University, Germany

Prof. PhD **Gheorghe Grigoras**, "Al. I. Cuza" University, Iasi, Romania

Lecturer PhD **Daniel Hunyadi**, "Lucian Blaga" University of Sibiu, Romania

Prof. PhD **Paul Corneliu Iacob**, "Transilvania" University, Brasov, Romania

Prof. PhD **Julian Ławrynowicz**, University of Lodz, Polish Academy of Sciences, Poland

Prof. PhD **Suzana Loskovska**, Faculty of Electrical Engineering and Information Technologies in Skopje, Macedonia

Prof. PhD **Daniela Marinescu**, Transilvania University of Brasov, Romania

Prof. PhD **Nikos Mastorakis**, Hellenic Naval Academy, Greece

Prof. PhD **Ioana Moisil**, Lucian Blaga University of Sibiu, Romania

Assoc. Prof. PhD **Antoanela Naaji**, "Vasile Goldis" Western University of Arad, Romania

Prof. PhD ing. **Cornelia Novac**, "Dunarea de Jos" University, Galati, Romania

Prof. PhD **Ioana Cristina Plajer**, "Transilvania" University, Brasov, Romania

Prof. PhD **Anca Ralescu**, University of Cincinnati, United States of America

Assoc. Prof. PhD **Livia Sangeorzan**, Transilvania University, Brasov, Romania

Lecturer PhD **Lucian Sasu**, "Transilvania" University of Brasov, Romania

Prof. PhD **Klaus Bruno Schebesch**, "Vasile Goldis" University, Arad, Romania

Prof. PhD **Dana Simian**, "Lucian Blaga" University of Sibiu, Romania

Lecturer PhD **Florin Stoica**, "Lucian Blaga" University of Sibiu, Romania

Table of Contents

Third International Conference
Modelling and Development of Intelligent Systems
October 10 - 12, 2013
"Lucian Blaga" University
Sibiu - Romania

Plenary Lecturer I

The Challenge of Emergent Phenomena in Emergent Applications Software

George Eleftherakis

Computer Science Department
CITY College, International Faculty of the University of Sheffield
Thessaloniki, GREECE
E-mail: eleftherakis@city.academic.gr

In the last few decades the application of distributed solutions to computerized systems has expanded to a variety of new domains, facing a growing number of users that demand more advanced services leading to emergent applications software. Present systems are not trustworthy, and this together with the predicted rise of complexity is going to lead us to an enormous increase of the cost aiming to build trustworthy systems. Combined with the vision of the Internet of things, and visions from leading companies like IBM's smarter planet, as well as the introduction of service oriented computing, cloud computing and similar technologies, has imposed a considerable strain on the design and operational performance of distributed systems. Consequently, the architectures upon which distributed systems are built have moved from the initial centralized structured approaches, to more decentralized solutions that avoid the single point failure problem and offer better utilization of network resources. Moreover, unstructured approaches, where the overlay network follows a random graph distribution, have been introduced in order to cope with churn, heterogeneity, as well as to avoid the topology constraints which create significant problems in open dynamic environments that utilize structured architectures. Latest research efforts have concentrated on developing hybrid solutions which combine different paradigms in terms of decentralization and structure.

In this context, many novel approaches have used biological systems as inspiration in the design of artificial distributed systems aiming for solutions to various problems and challenges encountered. The rationale for looking in nature for inspiration is based on the notion that the structure, the behaviours of individuals and the laws that govern their interactions in decentralized biological systems existing in nature seems to solve seamlessly and effortlessly problems common in open distributed ICT systems. Large scale biological collectives like ant colonies and termite hives have shown a remarkable ability to produce a variety of useful behaviours including availability, scalability, self- organization and adaptation in a fully decentralized manner.

Emergence is known to be the enabler for a variety of beneficial properties in natural systems. Adaptability, scalability and robustness, as well as a multitude of self-* properties, have been shown to emerge out of simple interactions at the microscopic level of a system. Distributed systems are particularly well suited to hosting emergent phenomena especially in cases where the individual nodes possess a high degree of autonomy and the overall control tends to be decentralized. Being able to engineer macroscopic behaviours in distributed systems by introducing behaviours and interactions of individual nodes inspired by systems found in nature could greatly assist with managing the complexity inherent into artificial distributed systems.

This talk is discussing the challenge of emergent phenomena, either positive or negative towards the behaviour of the system, in emergent applications software, and identifies some possible research direction towards harnessing emergent phenomena in engineered systems.

Brief Biography of the Speaker: George Eleftherakis is a Senior Lecturer and Research Coordinator of the CS department at CITY College Thessaloniki, which is an International Faculty of the University of Sheffield. He is also leading the Information and Communication Technologies Research Track of the South Eastern European Research Centre (SEERC). He holds a BSc in Physics from the University of Ioannina, Greece and, an MSc with distinction and a PhD in Computer Science from the University of Sheffield, UK. He is the head of the Intelligence, Modelling & Computation research group at CITY. His main research work is in the area of Formal Methods, Biologically Inspired Computing, Complex Systems, Emergence, Multi-Agent Systems, Education, Serious Games, and Information Security. He gave more than 15 invited talks to Universities, conferences and companies around the world and published more than 60 papers in International Conferences and Journals. He edited 8 books, more of them in the area of formal methods. He also organized, chaired and joined scientific committees of several international conferences in the areas of his research, and also a number of International Student Conferences (the Student Spring Symposium series and the South East European Doctoral Student Conference). He founded and is leading Thessaloniki's Java User Group. He is a Senior Member of the Association of Computing Machinery (ACM) chairing ACM's Council of European Chapter Leaders. He is also a member of the Greek Computer Society, serving the last years as a member of the administration board of the Macedonia-Thrace annex.

Third International Conference
Modelling and Development of Intelligent Systems
October 10 - 12, 2013
"Lucian Blaga" University
Sibiu - Romania

Plenary Lecturer II

Swarm Intelligence Optimization Algorithms in Image Processing

Milan Tuba
University Megatrend Belgrade
Faculty of Computer Science
Belgrade, SERBIA
E-mail: tubamilan@ptt.rs

Image processing is one of the most applicable scientific areas; it has been widely used in medicine, astronomy, quality control, security etc. Image processing is a large collection of very different techniques. The only common element is the digital image itself, while low level signal processing, medium level morphological processing and segmentation for feature detection and high level artificial intelligence algorithms for object recognition, information extraction, representation and understanding, belong to completely different areas. On these different stages of image processing some hard optimization problems occur. For example, multilevel image thresholding is a step in segmentation, but even though this problem at first sight seems to be simple, to determine optimal n numbers in the range 0-255 is NP-hard combinatorial problem. Such hard optimization problems have been recently successfully solved using nature inspired metaheuristics. Swarm intelligence is an important branch of this class of nondeterministic optimization methods. Here we present successful application of the latest swarm intelligence algorithms: Firefly algorithm, Cuckoo search and Bat algorithm to multilevel image thresholding.

Brief Biography of the Speaker: Milan Tuba is Professor of Computer Science and Provost for mathematical, natural and technical sciences at Megatrend University of Belgrade. He received B. S. in Mathematics, M. S. in Mathematics, M. S. in Computer Science, M. Ph. in Computer Science, Ph. D. in Computer Science from University of Belgrade and New York University. From 1983 to 1994 he was in the U.S.A. first as a graduate student and teaching and research assistant at Vanderbilt University in Nashville and Courant Institute of Mathematical Sciences, New York University and later as Assistant Professor of Electrical Engineering at Cooper Union Graduate School of Engineering, New York. During that time he was the founder and director of Microprocessor Lab and VLSI Lab, leader of scientific projects and supervisor of many theses. From 1994 he was Assistant Professor of Computer Science and Director of Computer Center at University of Belgrade, from 2001 Associate Professor, Faculty of Mathematics, and from 2004 also a Professor of Computer Science and Dean of the College of Computer Science, Megatrend University Belgrade. He was teaching more than 20 graduate and undergraduate courses, from VLSI Design and Computer Architecture to Computer Networks, Operating Systems, Image Processing, Calculus and Queuing Theory. His research interest includes mathematical, queuing theory and heuristic optimizations applied to computer networks, image processing and combinatorial problems. He is the author or coauthor of more than 150 scientific papers and coeditor or member of the editorial board or scientific committee of number of scientific journals and conferences. Member of the ACM since 1983, IEEE 1984, New York Academy of Sciences 1987, AMS 1995, WSEAS, SIAM, IFNA.

Third International Conference
Modelling and Development of Intelligent Systems
October 10 - 12, 2013
"Lucian Blaga" University
Sibiu - Romania

Plenary Lecturer III

Scheduling Algorithms. A review

Ioana Moisil
"Lucian Blaga" University of Sibiu
"Hermann Oberth" Faculty of Engineering
Sibiu, ROMANIA
E-mail: im25sibiu@gmail.com

Scheduling is one of the most complex topics that appear in almost all fields of activity, from scientific research to industrial applications. As most of the complex problems, scheduling is still a challenge for researchers, especially for those involved in optimization studies. In this contribution, after introducing the classification of scheduling problems, I will briefly present classical scheduling algorithms for solving single machine problems, parallel machine problems, and shop scheduling problems. Aspects concerning polynomial algorithms, procedures based on dynamic programming, combinatorial optimization and computational complexity will also be discussed. In the last part of the paper I will present some metaheuristic algorithms that are widely used in solving scheduling problems. The results of a computational study using metaheuristics i.e. an adaptive Ant Colony Optimization algorithm for the Resource-Constrained Project Scheduling will also be presented.

Brief Biography of the Speaker: Ioana Moisil received the M.Sc. in Mathematics at the University of Bucharest, in 1971, the scientific grade in Statistical, Epidemiological and Operation Research Methods Applied in Public Health and Medicine at the Universite Libre de Bruxelles, in Belgium, in 1991 and the Ph.D. in Mathematics at the Romanian Academy in 1997. Work places: the National Institute for Research & Development in Informatics - I.C.I (1971-1986), Carol Davila Faculty of Medicine Bucharest – department of Biophysics, CCSSDM Center of the Ministry of Health. At present she is a full-time Professor and a Senior Researcher at the Department of Computer Science and Automatic Control – Faculty of Engineering at the "Lucian Blaga" University of Sibiu. She is the author/co-author of fourteen books and over 150 scientific papers. Her scientific interests include intelligent systems, healthcare telematics, web technologies, data-mining, e-learning, modelling and simulation, uncertainty management, human-computer interaction. Professor Moisil participated in several EU funded projects as project manager for the national partner (Telenurse ID ENTITY, MGT, PROPRACTITION, PRO-ACCESS), in Tempus projects and in national funded projects as research manager and software development coordinator (INFOSOC – eUNIV, AMTRANS – eCASTOR, INFOSOC - e-Scribe, INFOSOC – DANTE, e-EDU-Quality, eTransMobility, CNCSIS 2007-code 33, Studies on multivariate interpolation, polinomial classifiers and applications, CNCSIS 2007 – cod 1502, Aspects concerning the psycho-cognitive abilities of artificial intelligent agents and applications in ITC based education). Ioana Moisil is a member of EARLI (European Association for Research in Learning and Instruction), she is Romanian representative in the IMIA SIG and EFMI WG5 Nursing Informatics, honorary member of the Bohemian Medical Association J.E.Purkyne of Bio-engineering and Medical Informatics, member of the ISCB – International Society for Clinical Biostatistics, a member of the National society of Medical Engineering and biological Technology, of the Romanian General Association of Engineers, member of the IITM- International Institute of Tele-Medicine and of the Romanian Society of Mathematics Sciences. She is vice-president of the Romanian Medical Informatics Society; vice-president of the HIT Foundation for Health Informatics and Telematics and a member of RoCHI-ACM. Professor Moisil is taking part in several international peer-review committees and conferences scientific boards.

Third International Conference
Modelling and Development of Intelligent Systems
October 10 - 12, 2013
"Lucian Blaga" University
Sibiu - Romania

Approximation of bivariate functions by truncated classes of operators

Octavian Agratini, Saddika Tarabie, Radu Trîmbiţaş

Abstract

Starting from a general class of positive approximation processes of discrete type expressed by series, we indicate a way to modify the operators into finite sums. The new operators are suitable to be generated by software. Examples are delivered.

AMS 2000 Subject Classification: 41A36.
Keywords and phrases: linear positive operator, error of approximation.

1 Introduction

It is known that Approximation Theory, an old field of mathematical research, has a great potential for applications to a wide variety of problems. The study of the linear methods of approximation, which are given by sequence of linear and positive operators, became a firmly entrenched part of Approximation Theory. Usually, two types of positive approximation processes are used – the discrete respectively continuous form. In the first case, multiple classes of linear positive operators are expressed by series. We recall two classical examples of such operators used to approximate functions defined on unbounded intervals. We refer to Mirakjan-Szász operators S_n, $n \in \mathbb{N}$, and Baskakov operators V_n, $n \in \mathbb{N}$, respectively. They are defined as follows

$$\begin{aligned}(S_nf)(x) &= \sum_{k=0}^{\infty} s_{n,k}(x) f\left(\frac{k}{n}\right), \quad s_{n,k}(x) = \frac{(nx)^k}{k!} e^{-nx}, \quad x \geq 0,\\ (V_nf)(x) &= \sum_{k=0}^{\infty} v_{n,k}(x) f\left(\frac{k}{n}\right), \quad v_{n,k}(x) = \binom{n+k-1}{k} x^k (1+x)^{-n-k}, \quad x \geq 0,\end{aligned} \tag{1}$$

where f belongs to the space $C_2(\mathbb{R}_+)$, $\mathbb{R}_+ := [0, \infty)$,

$$C_2(\mathbb{R}_+) = \{f \in C(\mathbb{R}_+) : \lim_{x\to\infty} (1+x^2)^{-1} f(x) \text{ is finite}\},$$

endowed with the norm $\|\cdot\|$, $\|f\| = \sup_{x\geq 0} (1+x^2)^{-1} |f(x)|$.

As can be seen, the construction of such operators requires an estimation of infinite sums and this fact restricts the operators usefulness from the computational point of view. A question arises: how can we modify the operators to became usable for generating software programmes for approximation of functions. In this respect it is useful to consider partial sums which have only finite terms depending upon n and x. For the above mentioned operators this approach has already been made. For example,

J. Grof [5] examined the operator $(S_{n,N}f)(x) = \sum_{k=0}^{N(n)} s_{n,k}(x)f(k/n)$ establishing that if $(N(n))_{n\geq 1}$ is a sequence of positive integers such that $\lim_{n\to\infty}(N(n)/n) = \infty$, then $(S_{n,N}f)$ converges pointwise to f. Also the following modified operators of Mirakjan-Szász respectively Baskakov-type were investigated

$$(S_{n,\delta}f)(x) = \sum_{k=0}^{[n(x+\delta)]} s_{n,k}(x)f\left(\frac{k}{n}\right), \quad (V_{n,\delta}f)(x) = \sum_{k=0}^{[n(x+\delta)]} v_{n,k}(x)f\left(\frac{k}{n}\right), \quad x \geq 0. \tag{2}$$

Here $[\alpha]$ indicates the largest integer not exceeding α. The first class was studied by Heinz-Gerd Lehnhoff [6] and the second has approached by J. Wang and S. Zhou [9]. In (2) the number of terms considered in sum depends on the function argument. Roughly speaking, the initial operators are truncated losing their "tails". Following this route, a one dimensional general case is investigated in [1].

The aim of this note is to present similar constructions for bivariate classes of discrete operators. Instead of a double series we consider a finite sum. This way the use of computers in approximating functions is possible with lesser effort. The focus of the paper is on presenting different examples comparing the approximations generated by the series and by corresponding "amputated" series.

2 The operators and their truncated variants

Following [2], we investigate operators useful to approximate functions defined on $\mathbb{R}_+ \times \mathbb{R}_+$. Therefore, on this domain we define for every $(m,n) \in \mathbb{N}\times\mathbb{N}$ a net of form $\Delta_{m,n} = \Delta_{1,m} \times \Delta_{2,n}$, where

$$\Delta_{1,m}(0 = x_{m,0} < x_{m,1} < \ldots) \quad \text{and} \quad \Delta_{2,n}(0 = y_{n,0} < y_{n,1} < \ldots).$$

Set $\mathbb{N}_0 = \{0\}\cup\mathbb{N}$. Products of parametric extensions of two univariate operators are appropriate tools to approximate functions of two variables. For this reason, the starting point is given by the following one-dimensional operators

$$(A_m f)(x) = \sum_{i=0}^{\infty} a_{m,i}(x)f(x_{m,i}), \quad (B_n f)(y) = \sum_{j=0}^{\infty} b_{n,j}(y)f(y_{n,j}), \tag{3}$$

where $a_{m,i}$, $b_{n,j}$ are non-negative functions belonging to $C(\mathbb{R}_+)$, $(i,j) \in \mathbb{N}_0\times\mathbb{N}_0$, such that the following identities

$$\sum_{i=0}^{\infty} a_{m,i}(t) = \sum_{j=0}^{\infty} b_{n,j}(t) = 1, \quad t \geq 0, \tag{4}$$

take place.

In the above $f \in \mathcal{F}_1(\mathbb{R}_+)$ where $\mathcal{F}_1(\mathbb{R}_+)$ stands for the domain of L_n containing the set of all continuous functions on $\mathbb{R}_+$ for which the series in (3) is convergent.

Starting from (3), for each $(m,n) \in \mathbb{N}\times\mathbb{N}$ we introduce a linear positive operator as follows

$$(L_{m,n}f)(x,y) = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} a_{m,i}(x)b_{n,j}(y)f(x_{m,i},y_{n,j}), \quad (x,y) \in \mathbb{R}_+^2, \tag{5}$$

where $f \in \mathcal{F}_2(\mathbb{R}_+\times\mathbb{R}_+)$, the space of all continuous functions on $\mathbb{R}_+\times\mathbb{R}_+$ for which the double series in (5) is convergent. We notice if the function f can be decomposed in the following manner $f(x,y) = f_1(x)f_2(y)$, $(x,y) \in \mathbb{R}_+^2$, then one has

$$(L_{m,n}f)(x,y) = (A_m f_1)(x)(B_n f_2)(y). \tag{6}$$

Actually, the method of using the product of parametric extensions of univariate operators is a classic one. It was first used in the context of multivariate polynomial interpolation. For example in [4] can be found many historical information on this topic.

Further on we indicate a truncated variant of operators defined at (5). Let $u=(u_s)_{s\geq 1}$, $v=(v_s)_{s\geq 1}$ be sequences of positive numbers such that

$$\lim_{s\to\infty}\sqrt{s}u_s=\lim_{s\to\infty}\sqrt{s}v_s=\infty. \tag{7}$$

Taking in view the net $\Delta_{1,m}$, we divide the set $\mathbb{N}_0$ into two parts

$$I(x,u_m)=\{i\in\mathbb{N}_0:\ x_{m,i}\leq x+u_m\}\quad\text{and}\quad \overline{I}(x,u_m)=\mathbb{N}_0\setminus I(x,u_m).$$

Similarly, via the network $\Delta_{2,n}$, we introduce $J(y,v_n)$ and $\overline{J}(y,v_n)$.

For each $(m,n)\in\mathbb{N}\times\mathbb{N}$ and any $f\in\mathcal{F}_2(\mathbb{R}_+\times\mathbb{R}_+)$ in [2], we defined the linear positive operators

$$\begin{aligned}(L^*_{m,n}f)(x,y;u,v)&\equiv(L^*_{m,n}f)(x,y;u_n,v_n)\\&=\sum_{i\in I(x,u_m)}\sum_{j\in J(y,v_n)}a_{m,i}(x)b_{n,j}(y)f(x_{m,i},y_{n,j}),\quad (x,y)\in\mathbb{R}^2_+.\end{aligned} \tag{8}$$

The approach made above represents the general framework. In particular we can consider $a_{n,i}=b_{n,i}$, $n\in\mathbb{N}$ and $i\in\mathbb{N}_0$. Moreover, the network applied to the set $\mathbb{R}_+\times\mathbb{R}_+$ is usually of the form $(i/m,j/n)$, $(i,j)\in\mathbb{N}_0\times\mathbb{N}_0$. In this case the operators defined by (5) turn into the following operators

$$(L_{m,n}f)(x,y)=\sum_{i=0}^{\infty}\sum_{j=0}^{\infty}a_{m,i}(x)a_{n,j}(y)f\left(\frac{i}{m},\frac{j}{n}\right),\quad (x,y)\in\mathbb{R}^2_+. \tag{9}$$

Their truncated version given at (8) becomes

$$(L^*_{m,n}f)(x,y;u,v)=\sum_{i=0}^{[m(x+u_m)]}\sum_{j=0}^{[n(y+v_n)]}a_{m,i}(x)a_{n,j}(y)f\left(\frac{i}{m},\frac{j}{n}\right),\quad (x,y)\in\mathbb{R}^2_+. \tag{10}$$

We mention that in the particular case $a_{n,k}=v_{n,k}$, $k\in\mathbb{N}_0$, see (1), the above sequence turns into the truncated version of bidimensional Baskakov operators studied by Walczak [8].

We discuss how the sequences defined by (10) are becoming approximation processes. Let $(\Lambda_n)_{n\geq 1}$ be a sequence of positive linear operators defined on the Banach space $C(K)$, $K\subset\mathbb{R}$, a compact interval.

The classical theorem of Bohman-Korovkin states: if $(\Lambda_n e_k)_{k\geq 1}$ converges to e_k uniformly on K, $k\in\{0,1,2\}$, for the test functions $e_0(x)=1$, $e_1(x)=x$, $e_2(x)=x^2$, then $(\Lambda_n f)_{n\geq 1}$ converges to f uniformly on K for each $f\in C(K)$. The requirement (4) ensures the identity $A_ne_0=e_0$. If we assume

$$\lim_n A_ne_j=e_j,\quad j\in\{1,2\}, \tag{11}$$

then $(A_nf)_{n\geq 1}$ converges to f uniformly on any compact $K\subset\mathbb{R}_+$.

Setting $e_{i,j}(x,y)=x^iy^j$, $i\in\mathbb{N}_0$, $j\in\mathbb{N}_0$, $i+j\leq 2$, according to a result of Volkov [7] the test functions corresponding to the bidimensional case are the following four: $e_{0,0}$, $e_{1,0}$, $e_{0,1}$, $e_{2,0}+e_{0,2}$.

Since $L_{m,n}e_{i,j}=(A_me_i)(A_ne_j)$, see (6), relation (4) and our hypotheses (11) guarantee that the sequence $(L_{m,n})$ is an approximation process,. Taking in view (7) and following a similar route as in [1, Theorem 2] we can assert that $(L^*_{m,n}f)$ is also an approximation process. The advantage of using $L^*_{m,n}$ is that we work with finite sums, software enabling fast construction of operators.

3 Examples and graphs

We illustrate the effectiveness of construction given in (10) by choosing $a_{n,k}=s_{n,k}, k\in\mathbb{N}_0$, see (1). Consider the following functions

$$\begin{aligned}f_i&:[0,\infty)\times[0,\infty)\to\mathbb{R},\qquad i=1,2,3\\ f_1(x,y)&=e^{-x-y},\\ f_2(x,y)&=e^{x+y},\\ f_3(x,y)&=\sin x\sin y.\end{aligned}$$

If we apply the operator $L_{m,n}$ to our functions we obtain

$$\begin{aligned}(L_{m,n}f_1)(x,y) &= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\frac{\exp(-mx-ny)(mx)^i(nx)^j\exp\left(-\frac{i}{m}-\frac{j}{n}\right)}{i!j!}\\ &= \frac{1}{\exp\left(mx+ny-\frac{ny}{\exp(-1/n)}-\frac{mx}{\exp(-1/m)}\right)};\end{aligned}$$

$$\begin{aligned}(L_{m,n}f_2)(x,y) &= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\frac{\exp(mx+ny)(mx)^i(nx)^j\exp\left(\frac{i}{m}+\frac{j}{n}\right)}{i!j!}\\ &= \exp(-mx-ny+ny\exp(1/n)+mx\exp(1/m)).\end{aligned}$$

$$\begin{aligned}(L_{m,n}f_3)(x,y) &= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\frac{\exp(-mx-ny)(mx)^i(nx)^j\sin\frac{i}{m}\sin\frac{j}{n}}{i!j!}\\ &= -\frac{1}{2}\left(\cos\left(mx\sin\frac{1}{m}+ny\sin\frac{1}{n}\right)-\cos\left(mx\sin\frac{1}{m}-ny\sin\frac{1}{n}\right)\right)\cdot\\ &\exp\left(mx\cos\frac{1}{m}+ny\cos\frac{1}{n}-mx-ny\right)\end{aligned}$$

For $L^*_{m,n}$ we consider successively the following sequences

(i) $u^{(1)}, v^{(1)}$, where $u^{(1)}_m = m$, $v^{(1)}_n = n$;

(ii) $u^{(2)}, v^{(2)}$, where $u^{(2)}_m = \sqrt[3]{m}$, $v^{(2)}_(n) = \sqrt[3]{n}$.

In the sequel, for each function f_i, $i = 1, 2, 3$, we give the following graphs

- $L_{10,10}f_i$
- $(L^*_{10,10}f_i)(.,.,u^{(j)},v^{(j)})$, $j = 1, 2$;
- $\left|L^*_{10,10}f_i)(.,.,u^{(j)},v^{(j)}) - L_{10,10}f_i\right|$, $j = 1, 2$.

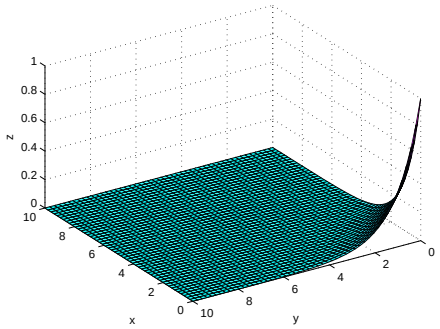
See Figures 1, 2, and 3.

Finally, we consider an example for which $L_{m,n}$ cannot be computed exactly. Let f_4 be given as follows

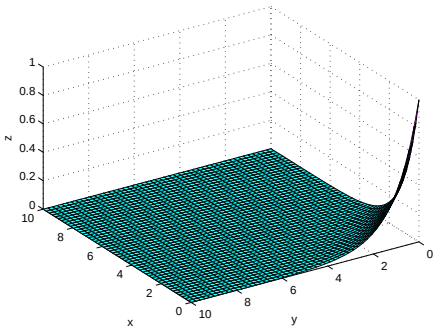
$$f_4 : [0,\infty)\times[0,\infty)\to\mathbb{R}, \qquad f_4(x,y) = \sin\sqrt{x^2+y^2}.$$

Figure 4 gives the graphs as above, excepting $L_{10,10}f_4$.

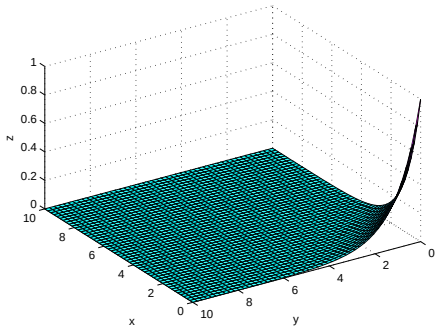
For graphical treatment of other types of bivariate operators see [3].



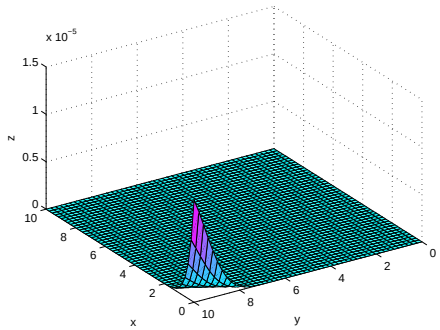
(a) $L_{10,10}f_1$



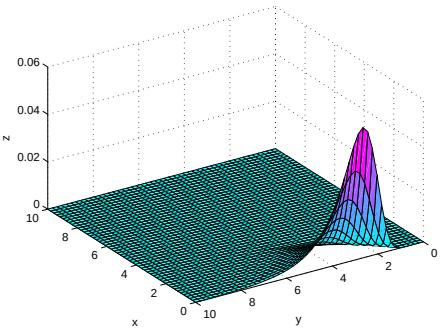
(b) $\left(L^*_{10,10}f_1\right)(\cdot,\cdot,u^{(1)},v^{(1)})$



(c) $\left(L^*_{10,10}f_1\right)(\cdot,\cdot,u^{(2)},v^{(2)})$

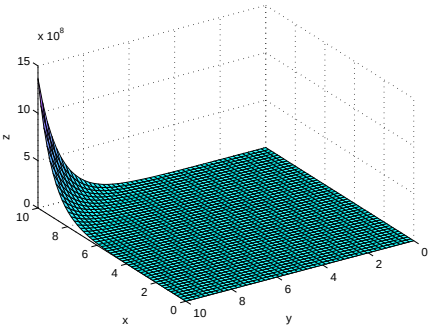


(d) $\left|\left(L^*_{10,10}f_1\right)(\cdot,\cdot,u^{(1)},v^{(1)})-(L_{10,10}f_1)\right|$

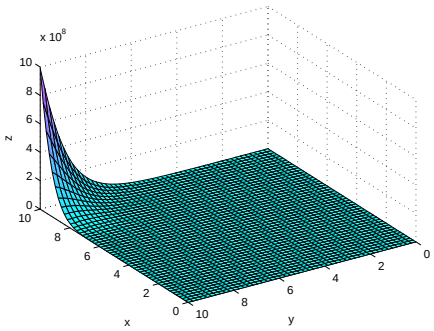


(e) $\left|\left(L^*_{10,10}f_1\right)(\cdot,\cdot,u^{(2)},v^{(2)})-(L_{10,10}f_1)\right|$

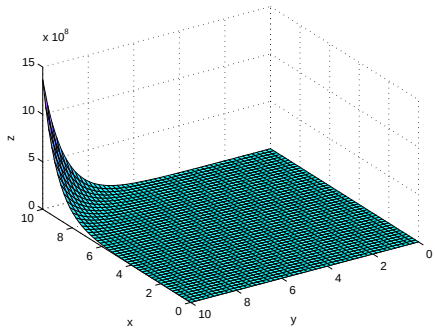
Figure 1: The graphs corresponding to f_1



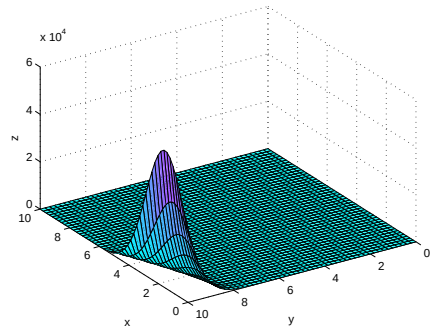
(a) $L_{10,10}f_2$



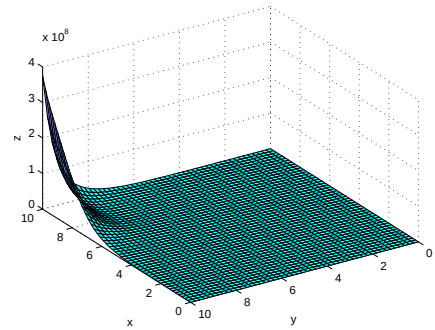
(b) $\left(L^*_{10,10}f_2\right)(\cdot,\cdot,u^{(1)},v^{(1)})$



(c) $\left(L^*_{10,10}f_2\right)(\cdot,\cdot,u^{(2)},v^{(2)})$

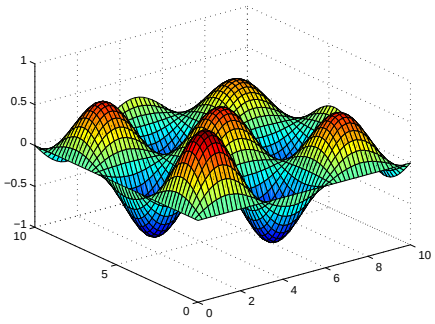


(d) $\left|\left(L^*_{10,10}f_2\right)(\cdot,\cdot,u^{(1)},v^{(1)}) - (L_{10,10}f_2)\right|$

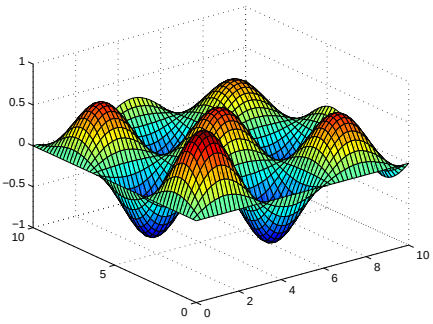


(e) $\left|\left(L^*_{10,10}f_2\right)(\cdot,\cdot,u^{(2)},v^{(2)}) - (L_{10,10}f_2)\right|$

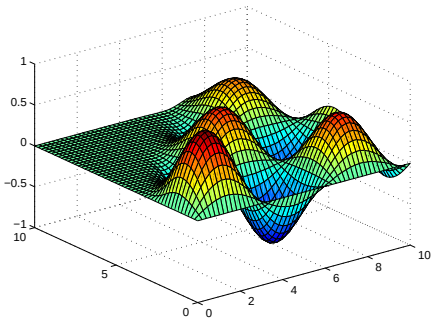
Figure 2: The graphs corresponding to f_2



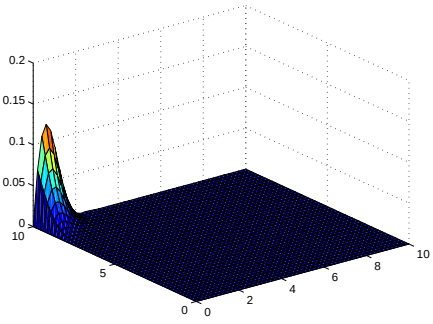
(a) $L_{10,10}f_3$



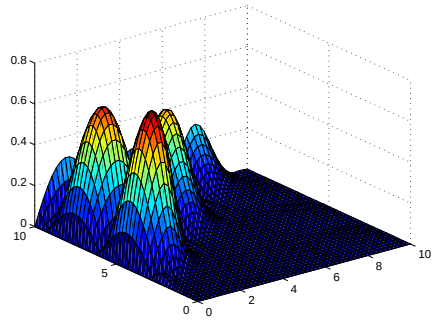
(b) $\left(L^*_{10,10}f_3\right)(\cdot,\cdot,u^{(1)},v^{(1)})$



(c) $\left(L^*_{10,10}f_3\right)(\cdot,\cdot,u^{(2)},v^{(2)})$

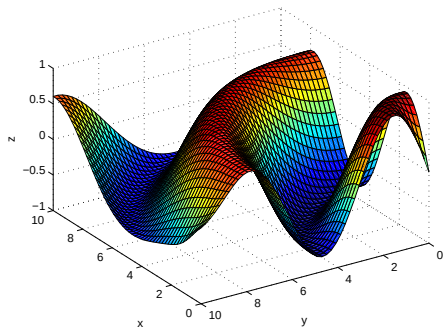


(d) $\left|\left(L^*_{10,10}f_3\right)(\cdot,\cdot,u^{(1)},v^{(1)})-(L_{10,10}f_3)\right|$

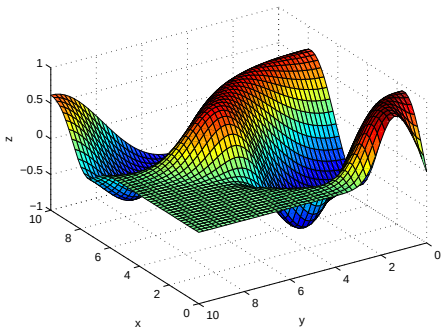


(e) $\left|\left(L^*_{10,10}f_3\right)(\cdot,\cdot,u^{(2)},v^{(2)})-(L_{10,10}f_3)\right|$

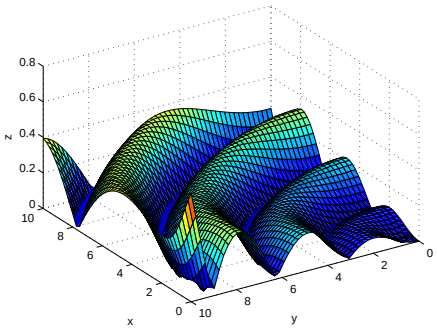
Figure 3: The graphs corresponding to f_3



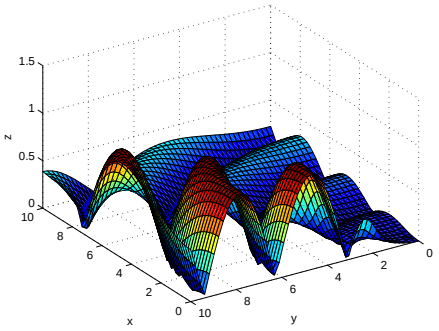
(a) $\left(L^*_{10,10}f_4\right)(\cdot,\cdot,u^{(1)},v^{(1)})$



(b) $\left(L^*_{10,10}f_4\right)(\cdot,\cdot,u^{(2)},v^{(2)})$



(c) $\left|\left(L^*_{10,10}f_4\right)(\cdot,\cdot,u^{(1)},v^{(1)})-f_4\right|$



(d) $\left|\left(L^*_{10,10}f_3\right)(\cdot,\cdot,u^{(2)},v^{(2)})-f_4\right|$

Figure 4: The graphs corresponding to f_4

References

[1] O. Agratini, On the convergence of a truncated class of operators, *Bull. Inst. Math. Academia Sinica*, 31: 213-223, 2003.

[2] O. Agratini, Bivariate positive operators in polynomial weighted spaces, *Abstract and Applied Analysis*, Volume 2013, Article ID 850760, 8 pages.

[3] Gh. Coman, Radu T. Trîmbiţaş, Multivariate Shepard interpolation, *Proceedings of SYNASC 2001, Timişoara*, RISC-Linz Report Series No. 01-20, Research Institute for Symbolic Computation, Johannes Kepler Univesity, Linz, Austria, pp. 6-14, 2001.

[4] M. Gasca, Th. Sauer, On the history of multivariate polynomial interpolation, *J. Comput. Appl. Math.*, 122: 23-35, 2000.

[5] J. Grof, Approximation durch Polynome mit Belegfunktionen, *Acta Math. Acad. Sci. Hungar.*, 35: 109-116, 1980.

[6] H.-G. Lehnhoff, On a modified Szász-Mirakjan operator, *J. Approx. Theory*, 42: 278-282, 1984.

[7] V.I. Volkov, On the convergence of sequences of linear positive operators in the space of continuous functions of two variables, *Dokl. Akad. Nauk SSSR* (N.S.), 115: 17-19, 1957 (in Russian).

[8] Z. Walczak, Baskakov type operators, *Rocky Mountain Journal of Mathematics*, 39:981–993, 2009.

[9] J. Wang, S. Zhou, On the convergence of modified Baskakov operators, *Bull. Inst. Math. Academia Sinica*, 28: 117-123, 2000.

Octavian Agratini
Babeş-Bolyai University
Faculty of Mathematics and Computer Science
Cluj-Napoca, ROMANIA
E-mail: *agratini@math.ubbcluj.ro*

Saddika Tarabie
Tishrin University
Faculty of Sciences
Latakia, SYRIA
E-mail: *sadikatorbey@yahoo.com*

Radu Trîmbiţaş
Babeş-Bolyai University
Faculty of Mathematics and Computer Science
Cluj-Napoca, ROMANIA
E-mail: *tradu@math.ubbcluj.ro*

Third International Conference
Modelling and Development of Intelligent Systems
October 10 - 12, 2013
"Lucian Blaga" University
Sibiu - Romania

Inertial Measurement Unit Simulator

Kiril Alexiev

Abstract

During the last few years microminiaturized inertial sensors were introduced in many applications. Their small size, low power consumption, rugged construction open doors to many areas of implementation. The main drawback of these sensors is the influence of different type of errors, leading to an unavoidable wrong position and orientation estimation. In the paper a simulator of Inertial Measurement Unit is proposed. The simulator is a tool for assistance of trajectory set up and on the base of input data it generates IMU output according given error/noise parameters. It allows us to simulate different types of IMUs based on prior knowledge of the IMU error's properties. One of the main goals in developing of the simulator is to validate new methods involving inertial technology. Something more, the simulator is an excellent tool for tuning complex filtering procedures and enhancing navigation accuracy. The simulation of different scenarios gives more information to receive better understanding of the weight of different sensor noises and errors on the final results.

1 Introduction

Inertial Measurement Unit (IMU) consists from one or more sensors, measuring the change of kinematic energy of a moving body. The sensors are divided in two groups: gyro sensors and accelerometers. Gyro sensor measures rotation rate of the body. Accelerometer provides information about linear acceleration of the body. Usually description of 3D motion of a body is given by 3 orthogonally placed accelerometers giving transition dynamic of the body and 3 orthogonally placed gyro sensors determining the orientation of the body. The axes of the both types of sensors normally coincide – e.g. in a 3D orthogonal coordinate system there are sensors to measure linear accelerations on each of the axes and rotation rate of the same axes. Thus the calculation process is also simplified. Two type of IMU were realized in the years. The first one is built on the scheme of the classical gyroscope and it preserves one and the same (initial) position, remaining independent of body rotation. In this case the body orientation is measured as a difference between gyroscopes axes orientation and the present orientation of the body - its roll, pitch and yaw. The second one, called also strapdown gyro sensor, is fixed tightly on the body and provides measurement of rate of rotation of the body. For this class of sensors, the body orientation is received through the integration of gyro measurements in respect to a priori known body orientation. Usually the strapdown sensors are produced as a MEM device with extremely high robustness and low power consumption. In this paper such a type of devices will be considered. The Inertial Navigation System (INS) is a system that relies entirely on inertial measurements for determination of dynamical body position and orientation. Today a wide range of

strapdown INS is available on the market. The simulator, presented in this paper, emulates the behaviour of standard MEM realization of an INS with three linear accelerometers and three angular rate sensors. It generates inertial sensor measurements in accordance with the precision and accuracy specifications of particular sensor sample. The improvement in computers' capability allows the simulation to become instrumental in technology development [5]. The tool, presented here, will be used to:

- Enhance our understanding of inertial technology;
- Simulate different types of IMUs based on prior knowledge of their specifications;
- Simulate a wide range of scenarios, even unrealistic ones;
- Test and validate new navigation algorithms;
- Study of different error propagation and estimation of the error influence over system precision and accuracy;
- Estimate the required hardware/sensor characteristics for a given application;
- Laboratory test of installed systems to assure that they are working properly before real test and to verify system performance in critical/rare situations.

A modular architecture is used in design of the proposed simulator that allows you to modify, improve and replace the individual modules without changing the overall architecture. Simulator gives also flexability in designing and research work and dramatically reduce time and money consuming field experiments. The system under test can be examined on different motion and vibration probations through the computer generation.

The paper is organized as follows. In the next chapter the mathematical background for inertial sensor modeling and simulation in navigation is revealed. Third chapter is devoted on error propagation for accelerometers and gyro sensors and a short overview of different error types is given. The fourth chapter describes the structure of IMU simulator. Some results are described in the next chapter. The concluding remarks are given the last chapter.

2 IMU based navigation (mechanization equations)

The body motion in an inertial frame of reference can be described as a result of simultaneous action of two forces - gravitational F_g and specific F_{sp}:

$$a_i = \frac{F_{sp}}{m_b} - \frac{F_g}{m_b} = a_{sp} - g\,, \tag{1}$$

where g is acceleration,caused by gravitational force and a_{sp} is the acceleration caused by specific force. Gravitational force is a function, depending on the distance between body and the Earth:

$$F_g = G\frac{M_e m_b}{r^2}\,, \tag{2}$$

where G is the gravitational constant $G = 6.6742 * 10^{-11}$, r is the distance between the interacting bodies, M_e is the mass of the Earth and $K = GM_e = 398600.44 * 10^9$.

To explain the specific force we introduce three frames of reference - one assosiated with the moving body, denoted by subscript $_b$, the second one is a geocentric frame, rotating with the rate of rotation of the Earth - it is associated with the subscribt $_e$ and the last one is also geocetric, but it is inertial and it is marked by subscribt $_i$. Let now denote the rate of the Earth rotation by ω. The last introduction note concerns the differential of a vector in absolute reference frame if it is presented in rotating system:

$$\frac{de_a}{dt} = \frac{de_r}{dt} + \omega \times e_a\,, \tag{3}$$

Let now express the velocity in inertial reference frame, applying expression from (3):

$$v_i = \frac{dr_e}{dt} + \omega \times r_e = v_e + \omega \times r_e , \tag{4}$$

The next step is to express acceleration, applying twice (3):

$$\frac{dv_i}{dt} = \frac{d(v_e)}{dt} + \frac{d(\omega \times r_e)}{dt} = \underbrace{\frac{dv_e}{dt} + \omega \times v_e}_{\text{first term}} + \underbrace{\omega \times \frac{dr_e}{dt} + \omega \times \omega \times r_e}_{\text{second term}} , \tag{5}$$

$$\frac{dv_i}{dt} = a_e + 2\omega \times v_e + \omega \times \omega \times r_e , \tag{6}$$

Regarding the received result as equal to specific acceleration and substituting in (2) we receive:

$$a_i = a_e + 2\omega \times v_e + \omega \times \omega \times r_e - g , \tag{7}$$

The acceleration $2\omega \times v_e$ is result of Coriolis force, and the term $\omega \times \omega \times r_e$ corresponds to centrifugal acceleration. Usually the last two terms of (7) are grouped together and replaced by so called local gravitational acceleration or simply gravity:

$$a_i = a_e + 2\omega \times v_e - g_l(h) , \tag{8}$$

where h is the height of the body above the Earth surface. The equation (8) is regarded as fundamental navigational equation.

It is worth to estimate the significance of all terms. Let consider a motion with velocity of 36 km/h on the Earth surface near to Equator. The applied force creates acceleration equal to 1 m/s^2. For this example $a_e = 0.1g$, $2\omega \times v_e = 1.46 * 10^{-3}$, $\omega \times \omega \times r_e = 3.4 * 10^{-2}$.

For calculation of g_l the Gelmert formula is applied:

$$g_l(h) = 9.7803\,(1 + 0.005302\ \sin^2\varphi - 0.000007\ \sin^2 2\varphi) - 0.00014\ - 2\omega_0^2 h , \tag{9}$$

where φ is northern latitude and $\omega_0 = 1.2383 * 10^{-3}$ is the Schuler frequency.

The goal of navigation is to find coordinates of a body and its orientation. In the case of IMU sensor the task is solved based on IMU measurements and integral equation (10) and (11):

$$r_i = r_0 + \int_0^t \int_0^t (g_l + a_i(t))dtdt \tag{10}$$

where r_0 is the initial body attitude in inertial coordinate system at time t=0.

The body space orientation can be described accordingly:

$$\alpha = \alpha_0 + \int_0^t \omega_i dt \tag{11}$$

where α_0 denotes the initial body orientation in inertial coordinate system at time t, ω_i is the vector 3D rate turn, adjusted to inertial coordinate system.

A simple algorithm for coordinate determination is presented below. The calculation scheme is based on Euler angles. Let us denote the rotation matrix, transforming a vector from the moving body to inertial coordinate system by $C_b^i(t)$. Then an acceleration vector $a_b(t)$ in the body coordinate system will be transformed to inertial coordinate system by (3):

$$a_i(t) = C_b^i(t) a_b(t) \tag{12}$$

Now the rotation matrix $C_b^i(t)$ will be represented by Euler angles [2]:

$$C_b^i(t) = C_z'(t) C_y'(t) C_x'(t), \tag{13}$$

where

$$C_x(t)=\begin{pmatrix}1 & 0 & 0\\ 0 & \cos(\varphi(t)) & \sin(\varphi(t))\\ 0 & -\sin(\varphi(t)) & \cos(\varphi(t))\end{pmatrix}, C_y(t)=\begin{pmatrix}\cos(\theta(t)) & 0 & -\sin(\theta(t))\\ 0 & 1 & 0\\ \sin(\theta(t)) & 0 & \cos(\theta(t))\end{pmatrix},$$

$$C_z(t)=\begin{pmatrix}\cos(\psi(t)) & \sin(\psi(t)) & 0\\ -\sin(\psi(t)) & \cos(\psi(t)) & 0\\ 0 & 0 & 1\end{pmatrix}$$

are the rotation matrixes that rotate vectors on angles $\varphi(t)$, $\theta(t)$, $\psi(t)$ on axes x, y and z. It is important to mention that the order of rotation is important. If the angles of rotation are sufficiently small:

$$\delta t \to \begin{cases}\delta\varphi \to 0\\ \delta\theta \to 0\\ \delta\psi \to 0\end{cases},$$

or the measurement sampling rate is sufficiently high (in other words satisfies Nyquist sampling rate, which guarantees that you capture a signal properly because you sample it at least twice per cycle of the highest frequency component it contains) the following substitutions for an angle α may be applied: $\cos\delta\alpha \approx 1$ and $\sin\delta\alpha \approx \delta\alpha$. The product of small angles can be also approximated by zero: $\delta\alpha * \delta\alpha \approx 0$. The final expression for the change in rotation matrix will be:

$$C_b^i(\delta t)\Big|_{\delta t\to 0}=\begin{pmatrix}1 & -\delta\psi & \delta\theta\\ \delta\psi & 1 & -\delta\varphi\\ -\delta\theta & \delta\varphi & 1\end{pmatrix} == I+\Delta_\upsilon \tag{14}$$

Finally, the rotation matrix is presented as a product of the rotation matrix at t and calculated above rotation matrix $C_b^i(\delta t)$, corresponding to small additional rotations, committed in time interval δt:

$$C_b^i(t+\delta t)=C_b^i(t)C_b^i(\delta t)=C_b^i(t)(I+\Delta_\upsilon) \tag{15}$$

Let now express the derivative of rotation matrix:

$$\dot{C}_b^i(t)=\lim_{\delta t\to 0}\frac{C_b^i(t+\delta t)-C_b^i(t)}{\delta t}=\lim_{\delta t\to 0}\frac{C_b^i(t)(I+\Delta_\upsilon)-C_b^i(t)}{\delta t}=C_b^i(t)\lim_{\delta t\to 0}\frac{\Delta_\upsilon}{\delta t}=C_b^i(t)\dot{\Delta}_\upsilon, \tag{16}$$

where $\dot{\Delta}_\upsilon=\begin{pmatrix}0 & -\omega_z & \omega_y\\ \omega_z & 0 & -\omega_x\\ -\omega_y & \omega_x & 0\end{pmatrix}$ and $\omega_x, \omega_y, \omega_z$ are the lastly received measurements from gyro sensors on corresponding axis.

The solution of (16) is $C_b^i(t)=C_b^i(0)\ell^{\dot{\Delta}_\upsilon}$.

The matrix exponent in solution can be presented as an infinite sum:

$$\ell^{\dot{\Delta}_\upsilon}=\sum_{k=0}\frac{\dot{\Delta}_\upsilon}{k!}=I+\frac{\dot{\Delta}_\upsilon}{1!}+\frac{\dot{\Delta}_\upsilon^2}{2!}+\cdots+\frac{\dot{\Delta}_\upsilon^k}{k!}+\cdots$$

Taking into account only the first two terms (linear approximation) we receive an approximate formula for recurrent calculation of rotation matrix:

$$C_b^i(t+\delta t)=C_b^i(t)(I+\Delta_\upsilon) \tag{17}$$

Let now calculate the exact expressions for angle derivatives. In the differential equation (16) we substitute the rotation matrix taking expression in explicit form from (13). The matrix equation will be resolved for matrix element (3,1) (3-rd row, 1-st column). The corresponding equation looks like:

$$\frac{d(-\sin\theta)}{dt} = \begin{pmatrix} -\sin\theta & \sin\varphi\cos\theta & \cos\varphi\cos\theta \end{pmatrix} \begin{pmatrix} 0 \\ \omega_z \\ -\omega_y \end{pmatrix} \tag{18}$$

Therefore:

$$\dot{\theta} = \cos\varphi\,\omega_y - \sin\varphi\,\omega_z \tag{19}$$

For matrix element (3,2) in a similar way we receive:

$$\dot{\varphi} = \omega_x + tg\,\theta\left(\sin\varphi\,\omega_y + \cos\varphi\,\omega_z\right) \tag{20}$$

To find the expression for $\dot{\psi}$ the equations that contain ψ have to be used. For example, if the element (1,1) is used:

$$\dot{\psi} = \frac{1}{\cos\theta}\left(\sin\varphi\,\omega_y + \cos\varphi\,\omega_z\right) \tag{21}$$

The equation (19), (20) and (21) are most often used for calculation of rotation angles between two successive gyro measurements with a linear approximation only.
The explained above mathematical model is implemented in the simulator.

3 IMU errors

The body attitude is calculated using simultaneously the measurements of 6 sensors - 3 gyros and 3 accelerometers. Body orientation is given by integration of gyro sensors measurements. Transition of the body is calculated by double integration of accelerometers readings, according current body orientation. The integration process quickly accumulates errors. Due to existence of almost constant gravitational acceleration even small errors in the estimates of orientation of the body cause big deviation in the decomposition of gravitational acceleration on the axes, leading to large scale of attitude errors. Due to the quality of sensors IMU are divided in four groups of class of accuracy [1]:

Table 1: Accumulated Error due to Accelerometer Bias Error

Grade	Accel. Bias Error [mg]	Horizontal Position Error [m]			
		1 s	10 s	60 s	1 hr
Navigation	0.025	0.00013	0.012	0.44	1600
Tactical	0.3	0.0015	0.15	5.3	19000
Industrial	3	0.015	1.5	53	190000
Automotive	125	0.62	60	2200	7900000

Table 2: Accumulated Error due to Accelerometer Misalignment

Accelerometer Misalignment [deg]	Horizontal Position Error [m]			
	1 s	10 s	60 s	1 hr
0.050	0.0043	0.43	15	57000
0.10	0.0086	0.86	31	110000
0.50	0.043	4.3	150	570000
10	0.086	8.6	310	1100000

Table 3: Accumulated Error due to Gyro Angle Random Walk

Grade	Gyro Angle Random Walk [deg/√hr]	Horizontal Position Error [m]			
		1 s	10 s	60 s	1 hr
Navigation	0.002	0.00001	0.0001	0.0013	620
Tactical	0.07	0.0001	0.0032	0.046	22000
Industrial	3	0.01	0.23	3.3	1500000
Automotive	5	0.02	0.45	6.6	3100000

As it can be seen from Table 1, Table 2 and Table 3, even small errors in gyro angle estimation may discredit navigation.
The sensors are subject to different types of errors due to sensor imperfectness, model inaccuracy or computational errors.
The main errors influencing on the attitude estimation accuracy may be grouped into three categories [3, 4]:
A. *Sensors do not provide perfect and complete data.*

- Bias errors produce constant or almost constant shift of sensor values from the true ones.
- The scale factor errors cause lack of correspondence between real turn velocities and real straight linear accelerations and output sensors readings (gyro and accelerometer correspondingly).
- Errors due to manufacturing imperfections in IMU. Usually they are caused by non-orthogonally placed accelerometer or gyro sensors on the chip or by lack of coincidence between axes of corresponding accelerometer and gyro sensors. The last error more often is initiated by the first one, but sometimes can exist alone.
- The sensors readings are also contaminated by additive Gaussian noise.
- Temperature dependent errors. Temperature deviation affects output readings.
- There is time synchronization problem. Sensors readings do not belong to one and the same moment of time.
- Dynamic error (lag of sensor reaction/response to force implementation).

B. *Imperfectness of the used models and computational arithmetic*

- Usually the model inaccuracy is caused by inexact sensor approximation, incorrect gravitational acceleration estimate.
- The computational errors are caused by limitations of computer arithmetic, iterative procedures for optimization, calculations of trigonometric functions, loss of orthonormality of matrices, etc.

C. External sources of disturbances (uncontrolled, unpredictable even unknown sources of different type disturbances)

- Platform vibration. The vibration counteracts to sensor accuracy. It depends of different random factors, platform dynamics, mass distribution, switching on/off of different devices, and etc.
- Others

The Fig. 1 below displays the influence of different types of errors on quality of attitude estimation.
Let consider now errors in sensor measurements.
The error propagation for acceleration sensors only looks like:

$$r_i = r_0 + \frac{g_l t^2}{2} + \int_0^t \int_0^t (a_i(t) + \varepsilon_a) dt dt = r_0 + \frac{g_l t^2}{2} + \frac{\varepsilon_a t^2}{2} + \int_0^t \int_0^t a_i(t) dt dt \qquad (22)$$

Here ε_a denotes the error vector of acceleration sensors.

The error propagation for gyro sensors only looks like:

$$\alpha = \alpha_0 + \int_0^t (\omega_i(t) + \vec{\varepsilon}_\omega)dt = \alpha_0 + \varepsilon_\omega t + \int_0^t \omega_i(t)dt \tag{23}$$

Here ε_ω denotes the error vector of gyro sensors.

The equations (22) and (23) give error propagation in the simplest case of independent errors. In practice there are many types of errors, influencing one to others. The influence of rotation rate error measurements on angle determination is obvious from (19), (20) and (21). As a consequence the error propagation in (22), for example, generates/induces nonlinear errors in estimation of accelerations, leading to quickly growing errors in estimated system position. That is why (22) and (23) are used only to approximate the order of generated errors and these equations are not of practical use.

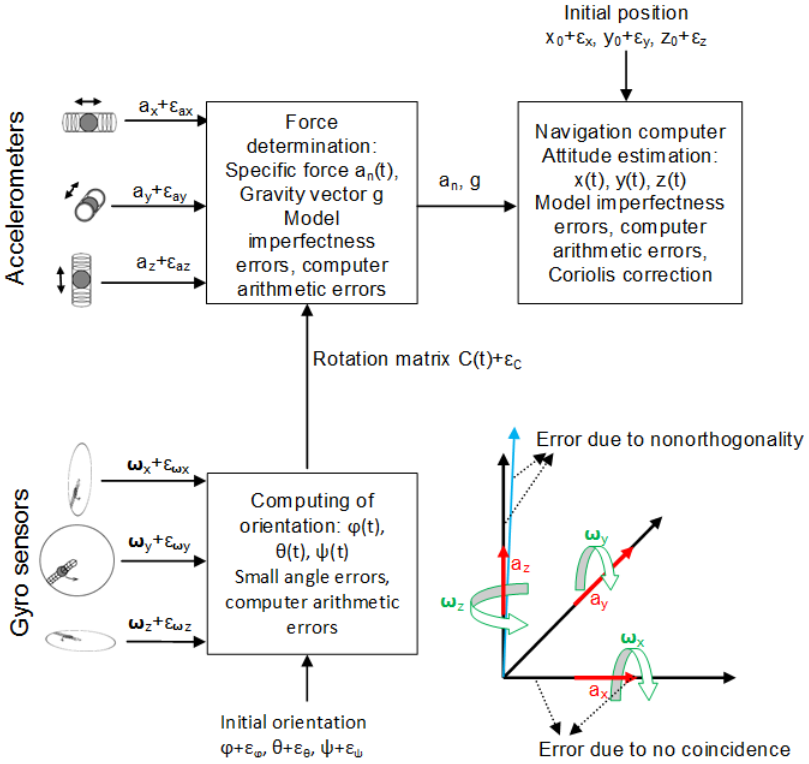


Fig. 1 Errors in an IMU

In order to minimize different type of errors we have to estimate their influence on the position estimate.

There are many well established methods for self-consistency check and normalization. One of them concerns the rows/columns of the rotation matrix. The rotation matrix is direction cosine matrix, which row/columns are projections of unity vector onto orthogonal axes. That means, that the sum of squares of values in each row/column have to be equal to 1 and due to their orthogonality, their scalar products have to be zero. In the cases of using quaternions the normalization means that the sum of squares of quarternion elements has to be equal to 1. This normalization usually doesn't correct errors. Even if optimization procedure is started, the best received result does not guarantee the error compensation. Moreover, the normalization algorithm propagates the error over correct terms. That is why the precise error expression is not of practical use.

4 The structure of IMU simulator

The simulator has modular structure, presented on fig. 2.
Input Data Interface is an interactive module with functionality to insert, edit, save and search user data. The problem of choice what kind of editor to be used for trajectory parameterization (graphical editor or text editor) was resolved in favor of the text editor, which, although being unfriendly and more cumbersome, allows exact parameterization of the trajectories.
Trajectory Generator uses kinematic equations to generate body trajectory. The module has direct output to graphical interface to check generated trajectories and correct them in the case of wrong input data.
Noise Generator adds different type of noises and inaccuracies. This module underwent several adjustments due to change of authors understanding of the influence of different errors on final result.
Inertial sensor model simulates inaccuracy and imperfectness of the sensors like sensor axes non-orthogonality, bias instability and scale errors, lag in sensor data, and others.
Navigation model consists of suit of tested algorithms. There are several classical realizations of navigational algorithms and their modifications for implementation in mobile devices.
Graphical output gives 3D presentation of generated trajectories, noised data and results of navigation algorithms' data processing. A special form of presentation of 3D body orientation is introduced.

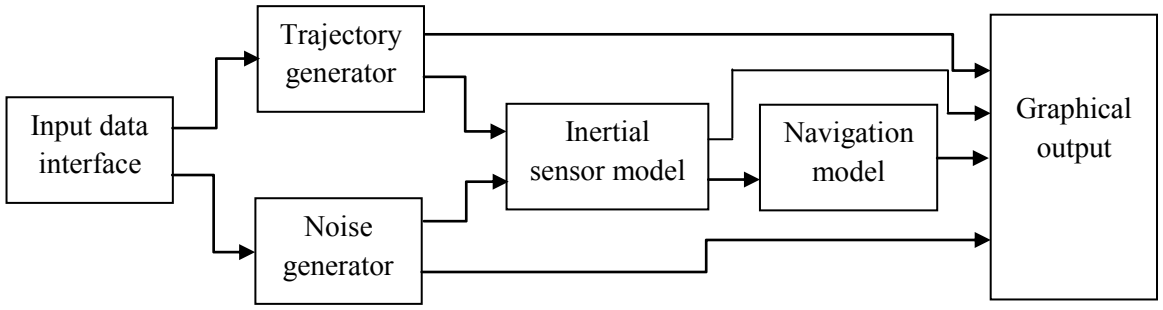


Fig. 2 The simulator structure

5 Results

The simulator was tested in an example for both: simulated data and real hardware generated data (a platform with MPU-6050 strapdown inertial sensors). The experiment on the fig 3 includes a simple body move following contour of a quadrate in horizontal plane. The data flow from simulator and sensors (3 gyros and 3 accelerometers) were saved and different types of navigation algorithms were applied. The hardware gyro and accelerometer signals are shown on Fig. 4. The calculated platform trajectory received by data processing from a navigation algorithm is shown on Fig. 5.

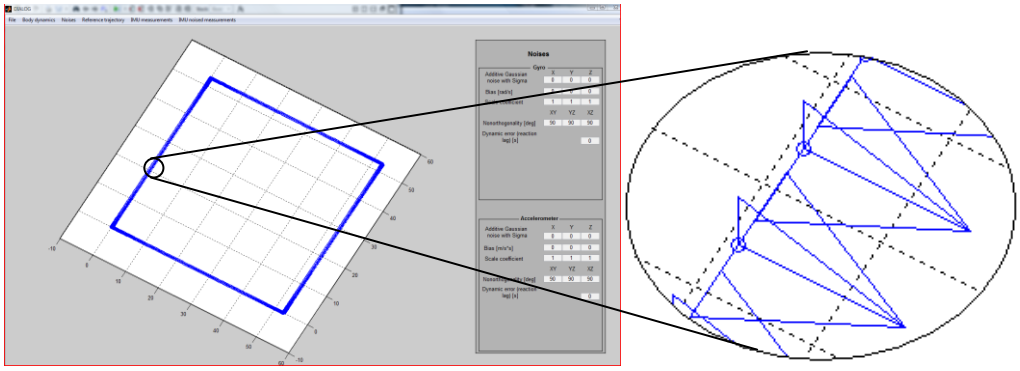


Fig. 3 Simulator with graphical output of the reference trajectory. In the circle on the right side the orientation of the moving body is presented.

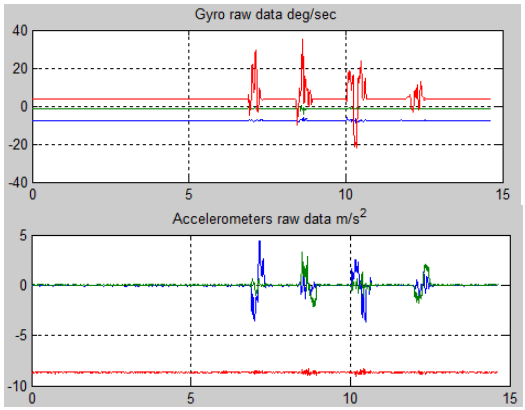


Fig. 4 Gyro and accelerometer sensors raw signals (from hardware platform)

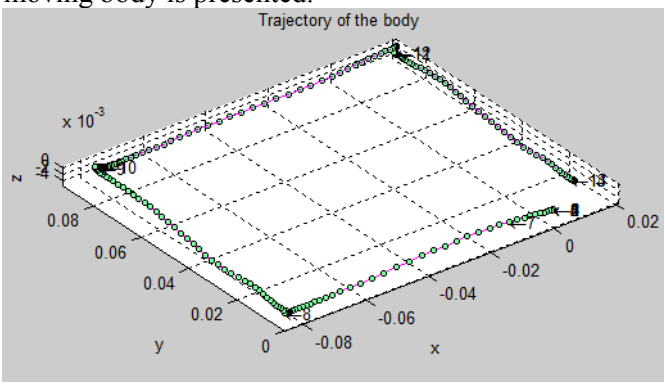


Fig. 5 The output results of navigation algorithm

6 Conclusion

The contemporary strapdown inertial MEMs are far behind in accuracy from the precise, very heavy and costly navigation platforms. In spite of this a lot of applications are waiting for more precise inertial sensors. The proposed in this article simulator of IMU shortened the road from idea generation to design of real application. It improves design by executing comprehensive and exhaustive simulations in the lab, minimising field testing. Something more, the simulator allows optimization of the choice of inertial sensors for a particular application based on published sensors datasheets only, materializing “software in loop” simulation approach. The modular structure of simulator allows further enhancement and enrichment of suit of algorithms. One of the most interesting directions for further development of the simulation tool is realization of “hardware in loop” simulation [6] through appropriate hardware interface and software drivers.

Acknowledgement: The research work reported in the paper is partly supported by the project AComIn "Advanced Computing for Innovation", grant 316087, funded by the FP7 Capacity Programme (Research Potential of Convergence Regions) and by the project No DFNI – I01/8 funded by the Bulgarian Science Fund. All data, laboratory equipment were supplied by “MM Solutions” in the framework of the project “Industrial research for development of technology for image enhancement and video stabilization using inertial sensors”, Contract BG161PO003-1.1.06-0037-C0001, Operational Program "Development of the Competitiveness of the Bulgarian Economy".

References

[1] http://www.vectornav.com/index.php?&id=76
[2] David H. Titterton, John L. Weston Navigation Technology - 2nd Edition, The Institution of Electrical Engineers, 2004, ISBN 0 86341 358 7
[3] Grewal, M.S., Weill L.R., Andrews A.P., Global Positioning Systems, Inertial Navigation, and Integration, John Wiley & Sons, 2001, ISBN 0-471-20071-9.
[4] Oliver J. Woodman, An introduction to inertial navigation, Technical Report UCAM-CL-TR-696, ISSN 1476-2986, 2007.
[5] http://www.sbir.gov/sbirsearch/detail/87111
[6] http://www.americangnc.com/products/rtgis.htm

KIRIL ALEXIEV
Institute of Information and Communication Technologies
Mathematical Methods for Sensor Information Processing
Sofia, 25A Acad. G. Bonchev Str.
BULGARIA
E-mail: alexiev@bas.bg

Third International Conference
Modelling and Development of Intelligent Systems
October 10 - 12, 2013
"Lucian Blaga" University
Sibiu - Romania

Detecting influenza epidemics based on real-time semantic analysis of Twitter data

Radu Balaj, Adrian Groza

Abstract

The paper presents a method for detecting influenza rates in a geographical region, using the knowledge extracted from Twitter messages. The novelty consists of using fuzzy description logic for linking natural language in Twitter streams with formal reasoning in description logic. In order to analyze the data in a medical context, the content of a Twitter post is transformed into the RDF stream format, and queried using the C-SPARQL query language. The conducted research lies in the larger context of semantic stream reasoning.

1 Introduction

The large-scale popularity of social platforms such as Facebook or Twitter makes them suitable for applications which analyze heterogeneous data from multiple users in real-time. One of the areas which can benefit from the nature of the data generated by these platforms is the public health domain.

1.1 Motivation

Public health is concerned with the prevention of diseases that might significantly impact certain communities of individuals. Traditional methods of disease surveillance, such as analyzing data provided by medical laboratories, are time-consuming and can result in a late detection of a disease epidemic by as much as two weeks [1]. Different alternative methods have been proposed in the literature, which take advantage of the real-time, huge amount of data made available on the Internet.

A question that this research addresses is *"Are the rates of influenza indicative of an epidemics outbreak in region X¿ '*. The choice of influenza is due to its large distribution and to its very familiar symptoms. Thus, it is much more likely that Twitter users will generate relevant data for the detection of influenza, more than for any other disease.

1.2 Technical challenge

An application which aims to monitor a great amount of real-time data has its own challenges. One of the recent areas of research which deals with such systems is stream reasoning [2]. Its main objective is the generation of new tools and methods which integrate data streams, the Semantic Web and reasoning systems. Therefore, these new approaches could tackle new and complex problems [8], which can not be solved using just one or two of those areas.

Stream reasoning is most suitable for applications which use data that is:

- *rapidly changing*: most reasoning system use a static knowledge base, which needs an update process in order to integrate new data.
- *of great dimensions*: the data is too big and/or too expensive to store.
- *heterogeneous*: the data itself can be inconsistent, as in the case of data provided by sensors (transmission errors) or extracted from multiple ontologies (difficulties integrating concepts).

Thus, considering the nature of the problem at hand and the choice of Twitter as the data source, the analysis of tweets for influenza detection is suitable to the domain of stream reasoning. The information provided by Twitter is rapidly changing (there are thousands of tweets generated every minute), of great dimensions (it is impractical to store messages) and it is heterogeneous (because the content of a message is susceptible to interpretation errors).

The remainder of this paper is structured as follows. The following section briefly describes the technical solution, before detailing the technical and theoretical background. Section 3 presents important aspects from the implementation of the system. Section 4 describes relevant experimental results, along with a discussion regarding these. The last two sections discuss similar approaches proposed in the literature and present the final conclusions of this research.

2 Technical and theoretical background

This section describes the relevant technical and theoretical aspects of the methods and tools used in the actual implementation of the solution. First of all, it is suitable to offer a brief overview of how these were used so that the reader can put them in the right context.

2.1 Short overview

The posts generated on the Twitter platform are collected by the application using the Twitter Streaming API. Thus, around 1% of the total number of messages generated at one time is available in the application. In order to process just the relevant messages, the content of a tweet is matched against a collection of relevant words (that either describe a symptom or mention flu/influenza directly). The Stanford Dependencies parser is used in order to provide the grammatical relationships of the words in a message, which helps at further discarding irrelevant tweets (by eliminating negative tweets). The result after the parsing phase is a set of symptoms and the corresponding attributes, if any. The symptoms which are checked come from a knowledge base in the form of an ontology. This ontology, called Symptom Ontology, is further refined in order to entail fuzzy notions of symptoms manifestation, using the fuzzyDL (fuzzy Description Logic) syntax. The protocol for detecting influenza in a Twitter message is a set of rules which are interpreted by the fuzzyDL reasoning engine. After applying the protocol, a message is transformed in the RDF stream format. A C-SPARQL query over the generated RDF stream gives the number of influenza cases detected in a certain period of time, for a certain geographical region. Figure 1 outlines the generic architecture, with its main components.

2.2 Technical background

The tools and methods mentioned in the overview are briefly described in this section.

Twitter Streaming API. This API[1] allows an application to connect to a Twitter streaming endpoint, which feeds the messages to the application, without the need to constantly make explicit requests. A

[1] https://dev.twitter.com/docs/streaming-apis

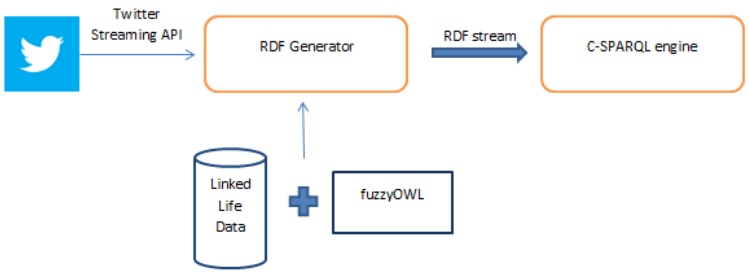


Figure 1: The architecture of the proposed solution.

random feed of messages is available through the $GETstatuses/sample$ endpoint. From the total number of messages, a percentage of 1% is available at any time.

Stanford Dependencies parser. Parsing a message in natural language is a difficult task. The SD parser [3] was built in order to provide a description of the grammatical relationships in a sentence, in a way that is understood by non-specialists. The main design principles of the parser is the representation of grammatical relationships through pairs of words and the choice of using traditional notations for those relationships. A parsed sentence has a hierarchical tree form, with the most generic relationship as the root of the tree.

Linked Life Data ontologies. Ontologies are a formal method of representing the knowledge from a domain. Since the functionality of the proposed solution is part of the medical domain, it is suitable to make use of existing ontologies. The Linked Life Data repository [4] is a platform for the integration of medical and biological data. It is comprised of more than 20 ontologies, one of which is of great use for this solution. Symptom Ontology (SO) is a hierarchical ontology which entails symptoms according to the body region in which they arise. These symptoms can be linked with concepts from other knowledge bases such as the Disease Ontology. We make use of SO for describing the influenza symptoms.

FuzzyDL. The fuzzyDL reasoner[2] is a Description Logic reasoner which supports fuzzy logic reasoning, by extending the classic description logic SHIF to the fuzzy case. In order to create or modify ontologies for fuzzyDL, a plug-in for the ontology editor Protege can be used, called fuzzyOWL2[3]. This plugin was used to add fuzzy notions to the already existing Symptom Ontology.

C-SPARQL. In order to interpret the knowledge from an ontology, the SPARQL query language is used. However, SPARQL can only query on static knowledge. Hence, the C-SPARQL [5] extension was designed in order to query over dynamic data in the RDF stream format. Thus, the language is extended to include physical (number of RDF triples) and logical (time-related) windows over which the query is applied.

Operation	Lukasiewivz Logic	Gödel Logic
intersection	$\alpha \otimes_L \beta = max\{\alpha + \beta - 1, 0\}$	$min\{\alpha, \beta\}$
union	$\alpha \oplus_L \beta = min\{\alpha + \beta, 1\}$	$max\{\alpha, \beta\}$
negation	$\alpha \ominus_L \beta = 1 - \alpha$	$\alpha \ominus_L \beta = 1,\ if\ a = 0,\ 0,\ otherwise - \alpha$
implication	$\alpha \Rightarrow_L \beta = min\{1, 1 - \alpha + \beta\}$	$1, if \alpha \leqslant \beta, \beta, otherwise$

Figure 2: Operators in Fuzzy Logics.

2.3 Theoretical aspects

Fuzzy Description Logic (FDL) has been proposed as an extension to classical description logic with the aim to deal with fuzzy and imprecise concepts, and it is based on the *SHIF(D)* version of the description logic. Some observations regarding the appliance of the fuzzy operators (figure 2) to argumentation follow:

The interpretation of Gödel operators maps the *weakest link principle* in argumentation. According to this principle, an argument supported by a conjunction of antecedents (α, β,...) is as good as the weakest premise ($min\{\alpha, \beta, ...\}$). Similarly, when several reasons are used to support a consequent, the strongest justification is chosen ($max\{\alpha, \beta, ...\}$). From the nature of the argumentative process itself, the subject of the debate cannot be easily categorised as true or false. The degree of truth for an issue and its negation ($1 - \alpha$) are continuously changed during the lifetime of the dispute. Thus, the different levels of trueness (and falseness) from fuzzy logic can be exploited to model argumentation.

The interpretation of Lukasiewics operators fits better to the concept of *accrual of arguments*. In some cases, independent reasons supporting the same consequent provide stronger arguments in favor of that conclusion ($min\{\alpha + \beta, 1\}$). Similarly, several reasons against a statement act as a form of collaborative defeat [7].

In the following paragraphs the differences introduced by fuzzy reasoning on top of classical description logic are presented. The complete formalisation of the fuzzy description logic can be found in [6].

A fuzzy knowledge base $K = \langle A, T, R \rangle$, consists of a fuzzy ABox *A*, a fuzzy TBox *T* and a fuzzy RBox *R* [6]. A fuzzy ABox A consists of a finite set of assertion axioms for fuzzy concepts $\langle x : C, \alpha \rangle$, and fuzzy roles $\langle (x, y) : R, \alpha \rangle$, where $\alpha \in [0, 1]$, C is a concept, and R a role. For instance, $\langle david : SmallPerson, 0.8 \rangle$ states that $david$ is a SmallPerson with degree at least 0.8, whilst $\langle (david, goliat) : attack, 0.7 \rangle$ says that *david* has attacked *goliat* with degree at least 0.7. If α is omitted, the maximum degree of 1 is assumed.

A fuzzy TBox T is a finite set of inclusion axioms $\langle C \sqsubseteq_S D, \alpha \rangle$, where $\alpha \in [0, 1]$, C, D are concepts, and S specifies the implication function (Lukasiewicz, Gödel) to be used. The axioms state that the subsumption degree between C and D is at least α.

A fuzzy RBox is a finite set of role axioms of the form: $(fun\ R)$, stating that the role R is functional; $(trans\ R)$, stating the role R is transitive, $R_1 \sqsubseteq R_2$, meaning the role R_1 is subsumed by the role R_2; and $(inv\ R_1\ R_2)$, stating the role R_1 is the inverse of the role R_2.

The main idea of *semantics* of FDL is that concepts and roles are interpreted as fuzzy subsets of an interpretation's domain [6]. A fuzzy interpretation $I = (\Delta^I, \bullet^I)$ consists of a non empty set Δ^I (the domain) and a fuzzy interpretation function $\bullet^I$. The mapping $\bullet^I$ is extended to roles and complex concepts as specified in figure 3.

[2]http://gaia.isti.cnr.it/straccia/software/fuzzyDL/intro.html

[3]http://gaia.isti.cnr.it/straccia/software/FuzzyOWL/index.html

$\bot^I(x) = 0$	$(\forall R.C)^I(x) = inf_{y \in \Delta^I} R^I(x,y) \Rightarrow C^I(y)$
$\top^I(x) = 1$	$(\exists R.C)^I(x) = sup_{y \in \Delta^I} R^I(x,y) \otimes C^I(y)$
$(\neg C)^I = \ominus C^I(x)$	$(\forall T.d)^I(x) = inf_{y \in \Delta^I} R^I(x,v) \Rightarrow d^I(y)$
$(C \sqcap_S D)^I(x) = C^I(x) \otimes_S D^I(x)$	$(\exists R.C)^I(x) = sup_{y \in \Delta^I} R^I(x,v) \otimes d^I(y)$
$(C \sqcup_S D)^I(x) = C^I(x) \oplus_S D^I(x)$	$(x : C)^I = C^I(x^I)$
$(C \rightarrow_S D)^I(x) = C^I(x) \Rightarrow_S D^I(x)$	$((x,y) : R)^I = R^I(x^I, y^I)$
$(m(C))^I(x) = f_m(C^I(x))$	$(C \sqsubseteq D)^I(x) = inf_{x \in \Delta^I} C^I(x) \Rightarrow_S D^I(x)$

Figure 3: Semantics of fuzzy concepts.

3 Implementation

This section describes in detail the most significant aspects of the implementation process. The focus here is on the tweet parsing algorithm, the design of the fuzzy ontology and the two reasoning steps.

3.1 Modelling fuzzy symptoms

The advantages of fuzzy logic for the representation of symptoms, such as presented in this work, are numerous. First of all, such a representation yields much more accurate results when trying to detect a disease given a number of symptoms. Secondly, the linguistic variables provide an appropriate method of describing a symptom, in a manner similar to how humans verbally express them (e.g. high fever, mild stomach pain). Last but not least, fuzzy logic applied to medical diagnosis has a long history, given by the amount of literature that can be found on the subject.

First of all, three linguistic variables were created: weak, moderate and strong. Each of these is a datatype, mapped on the [0,1] domain. A data property named *isManifested* was created, with the domain being any symptom in the ontology and the range one of the three fuzzy datatypes. Thus, it can be stated that a symptom has three possible degrees of manifestation.

In order to create relevant rules for detecting influenza in a message, we make use of the medical literature through the research presented in [9]. Table 1 presents some of the data based on which the rules from ontology have been created.

Symptom	Percentage(%)
fever	76
cough	69
fever+cough	82
fever+cough+headache	78
fever+cough+runny nose	81

Table 1: Symptoms and their power of predicting influenza infection.

One rule is designed as an equivalence class in the ontology as shown in table 2. There are nine such classes and therefore nine rules.

In order to make use of these rules, a concept of type Weighted Complex Concept(WCC) is created, called InfluenzaWeightedConcept (IWC). A WCC is a class which is composed of two or more classes, with different weights attached to them. The weighting type chosen is weighted maximum, which means that the class with the greatest weight will be chosen. This matches the Gödel interpretation of fuzzy logic, which states that the strongest justification is chosen when there are several reasons to support a consequent.

Class Name	Rule
I1	fever and (isManifested some strong)
I2	fever and ((isManifested some moderate) or (isManifested some weak)))
I3	cough and (isManifested some strong)
I4	cough and ((isManifested some moderate) or (isManifested some weak)))
I5	(chills and (isManifested some moderate)) or (fever and ((isManifested some moderate) or (isManifested some weak)))
I6	fever and cough and headache
I7	fever and cough
I8	(fever or cough) and (pain or sneezing or weakness or 'runny nose' or 'muscle pain')
I9	headache or pain or sneezing or weakness or 'runny nose' or 'muscle pain'

Table 2: Echivalence classes act as rules for detecting influenza.

Practically, if more than one of the nine rules is active (for example, a Twitter message contains both the fever and cough symptoms), the final value in detecting the influenza probability is set by the active rule with the greatest weight.

The definition for the IWC concept is the following:

$$\begin{aligned} InfluenzaWeightedConcept = \quad & 0.68 * I1 + 0.76 * I2 + 0.69 * I3 + 0.62 * I4 + 0.76 * I5 + \\ & 0.78 * I6 + 0.82 * I7 + 0.81 * I8 + 0.1 * I9 + 1 * I10 \end{aligned}$$

Note that the I10 class has been added by necessity. A WCC with the weighting type maximum needs at least one class with weight 1. I10 will not affect the final result, the term is 0 in all scenarios. The weight for each class was set according to the predictive value of the symptoms they entail. Since this is a fuzzy context, an instance a of class C can have a value between (0,1], which signifies the fuzzy degree to which that instance a is a member of class C. Therefore, in practice, the value associated with the instances of the class IWC can give us the degree to which that individual can be said to be a member of the class which represents positive influenza tweets.

For the fuzzyDL reasoner to compute the degree of an instance a of class IWC, the following query is used: $(min - sat?\ InfluenzaWeightedConcept\ [a])$. This query returns the minimum satisfiability degree of the instance a in the class IWC and gives the probability that a Twitter message, abstracted by instance a, entails relevant information regarding influenza.

3.2 Twitter message parsing

Before parsing a tweet using the Stanford Dependencies method, some filtering needs to be applied. First of all, only Twitter messages in English which have the geolocation property set are used. That is because without the geolocation property, it is impossible to detect influenza rates in a geographical region.

Furthermore, only tweets which contain words indicative of influenza, such as flu or symptoms, are processed by SD. The other ones are discarded. The parsing algorithm which extracts the data needed for the reasoning tasks consists of four important steps.

(1) First of all, vocabularies of words are initialised. These are used to match if a Twitter message has symptoms or words referring to influenza in its body, and to match a symptom's attribute to their corresponding fuzzy datatype (weak, moderate or strong). If a word like *flu* matched, the symptoms are no longer searched in the message. It is considered that the message entails information regarding influenza, and the protocol for detecting influenza is not used. Otherwise, if one or more symptoms matched, they need to be extracted, along with their possible attributes, and passed to the fuzzyDL reasoner.

```
REGISTER QUERY DetectInfluenza AS
SELECT (COUNT(?s) as ?NoOfTweets)
FROM STREAM < http : //myexample.org/streamInfluenza > [RANGE 24 HOUR TUMBLING]
WHERE { ?s < http : //myexample.org/streamInfluenza/hasInfluenza/ > ?o.
?s2 < http : //myexample.org/streamInfluenza/hasLongitude/ > ?long.
?s3 < http : //myexample.org/streamInfluenza/hasLatitude/ > ?lat.
FILTER( ?s = ?s2 && ?s = ?s3
&& lat >= 39 && lat <= 42
&& ?long >= 73 && ?long <= 75 )}
```

Figure 4: C-SPARQL query for determining the number of positive influenza tweets.

(2) In the case the matched words are *flu* or $influenza$, the result from SD parsing is used to make sure that the meaning of the message is positive, rather than negative, such as in the tweet *"As it turns out, I do not have theflu!"*. To detect a negative tweet, the formula used is:

$$if\ neg(verb, negation)\ and\ dobj(verb, flu) \Rightarrow negative$$

If it is found to be negative, the tweet is discarded.

(3) If the body of the message is matched against one or more symptoms, the grammatical relationship $amod$ determines if a symptom has an attribute that can help model the fuzzy degree of manifestation. For example, parsing the message *"I have a terrible fever, no work for me today."* determines the grammatical relationship $amod(terrible, fever)$. Therefore, the symptom fever has associated a fuzzy degree of manifestation.

(4) If an attribute is found, check against the vocabularies in order to establish to which fuzzy datatype the attribute belongs: weak, moderate or strong. In this case, terrible is matched against the vocabulary of words representing the $strong$ fuzzy datatype.

3.3 Stream reasoning

The fuzzyDL reasoner, for a Twitter message, gives the fuzzy degree of membership to the class of relevant influenza messages. This result, along with the geolocation property of the tweet, is converted in the RDFs format. A RDF stream is made up of triples (subject, predicate, object), to which a timestamp is attached. The application generates information in the following format:

$(\prec: Tweet_i : hasInfluenza : val \succ, 2013 - 07 - 12T12 : 34 : 40)$
$(\prec: Tweet_i : hasLatitude : lat \succ, 2013 - 07 - 12T12 : 34 : 40)$
$(\prec: Tweet_i : hasLongitude : long \succ, 2013 - 07 - 12T12 : 34 : 40)$

In order to determine the number of influenza cases detected by the system over a period of 24 hours, the query in figure 4 is used.

In this example, we compute the number of tweets for the city of New York and its surroundings. This is accomplished by filtering the results according to the right geographical coordinates.

4 Experimental results and discussions

There are two important aspects in regard to the functionality of the system which need to be verified experimentally. First of all, the efficiency of the parsing method needs to be tested, in order to check the number of correctly categorised messages which enter the reasoning process. Table 3 presents the

percentage of positively detected influenza messages, when a given word has been matched during the parsing algorithm.

Word found	Number of correctly categorised messages (in 100)
Flu	37%
Fever	48%
Cough, coughing	66%
Headache	69%
Sneeze, sneezing	33%

Table 3: Statistics of correctly categorised messages, by expression matched.

While cough and headache have a good percentage, other words have low detection efficiency. However, most of these are limitations of the currently used modules. For example, more than half of the incorrectly processed flu messages are introduced because the language used is not English. This is due to the fact that the Twitter Streaming API does not set the language property at the message body level, but at the user level. Therefore, a Twitter user can have the language set to English on his or her profile, yet write messages in other languages.

Moreover, as future improvements, the Stanford Dependencies parser can be used to filter common expressions which lead to incorrectly categorised messages. One such expression uses the word fever figuratively, in terms like $Bieberfever$ or $basketballfever$, found in more than 20 of the incorrectly detected tweets.

In order to test the functional aspect of the proposed solution, which is the detection of influenza rates, some experiments need to be performed. Over a course of one week (12-21 August), our application detected 8 influenza cases.

Unfortunately, it is difficult to compare these results with official reports. The New York State Department of Health monitors influenza rates only in the influenza season, which lasts from October to May. Therefore, there are no official weekly reports for August. However, from the archive of the Department of Health, for a period close to August, the week ending on June 30 2012[4], there were 13 laboratory confirmed cases of the influenza virus. The report for the week ending on the 11th of May 2013[5], 39 cases were laboratory confirmed.

Taking into consideration these results, it can be concluded that 8 cases for a week in August is a good approximate. However, for a more precise evaluation, the application needs to be tested in the winter period, when the influenza cases rise to the order of thousands per month. Moreover, the application has certain limitations which makes it difficult to simply compare the results with the official reports. For example, the application only receives 1% of the Twitter posts, and only processes messages which have the geolocation property set.

As future work, the application needs to be tested for larger periods of time. To properly compare results, the exact number of official cases should not be used. That is because there is no reason to believe that the officially reported number has to be the same as the number reported by the application. Instead, it should be seen if the fluctuation in official reports is also expressed in Twitter messages. For example, if the number of influenza cases rises by 10%, a similar trend should be reported by the application. Only after that, the solution can be said to correctly detect influenza epidemics, by reporting a number of cases over the epidemics threshold, in accordance to official reports (which are later generated). After that, the efficacy of the solution can be compared with the already existing ones, presented in the future section.

[4]http://www.health.ny.gov/diseases/communicable/influenza/surveillance/2011-2012/archive/2012-06-30/

[5]http://www.health.ny.gov/diseases/communicable/influenza/surveillance/2012-2013/archive/2013-05-11_flu_report.pdf

5 Related work

The most influent approach which makes use of the user generated data for detecting influenza epidemics is Google Flu Trends [1]. The data from five years of Google search archives and the official reports of CDC (Centers for Disease Control and Prevention) are used to create a machine learning model. The model can detect, based on the words used in user queries, influenza rates in a specific region. The result of applying this method, with a correlation between 0.85 and 0.96, is a good baseline for similar approaches.

Besides search engines, social networking platforms such as Twitter are becoming a very popular option for data provenance. Twitter messages are used for detecting influenza epidemics in [10]. The information is extracted using the Twitter Streaming API and, using a Support Vector Machine classifier, a tweet is labeled as negative (no useful information such as ambiguous or generic messages) and positive. The final correlation value of 0.89 is similar to the one in [1].

An influential work, based on the semantic analysis of Twitter messages is [11]. Linked Open Social Signals annotates tweets with semantic content and queries over them using the SPARQL language. The messages are extracted using the Twitter Streaming API, followed by a step of extracting relevant information. At the end of the extraction process, a micropost consists of the original text message, the author, the date of the post, its geolocation and a collection of entities, hashtags and URLs. This information is modelled under RDF(S)/OWL, by semantically annotating messages with concepts from the Linked Open Data Cloud. For example, the FOAF ontology (Friend of a Friend) is used to model Twitter users. The users have the possibility of querying the semantic information using SPARQL.

Another semantic approach to analyzing data is [12]. The data source used is Glue, a social network which allows its subscribers to connect with other users and share information regarding their favourite movies or sports. The platform makes use of semantic techniques for publishing topics in the form of RDF streams. C-SPARQL queries are used to reason over the data, using two methods of reasoning: deductive and inductive.

Twitter has been used in order to provide data regarding other public health issues, such as sentiment towards new tobacco products [13]. Again, the methods employed for detecting and classifying the relevant messages belong to the area of machine learning.

The research presented in this paper differs from the similar approaches presented in this section in two important aspects. One is the choice of semantic technologies for detecting influenza epidemics, instead of machine learning. The other one is the use of fuzzy semantic concepts, instead of just the classic OWL representation of the knowledge ontologies.

6 Conclusion

A novel approach to detecting influenza epidemics was presented. The fuzzy concepts employed help better shape the diagnosis protocol, by using linguistic variables appropriate to human language. Moreover, the semantic approach enables the solution to be further extended to concepts in the Symptom Ontology and to be useful in the broader context of the Semantic Web.

Acknowledgment: Part of this work was supported by the PN-II Romania-Moldova bilateral agreement ”ASDEC: Structural Argumentation for Decision Support with Normative Constraints”, 2013-2014.

References

[1] J. Ginsberg, M. Mohebbi, R. Patel, L. Brammer, M. Smolinski, and L. Brilliant, Detecting infuenza epidemics using search engine query data. *Nature*, vol. 457, no. 7232, pp. 1012-1014, 2009.

[2] E. D. Valle, S. Ceri, F. van Harmelen, and D. Fensel, It's a streaming world! reasoning upon rapidly changing information. *IEEE Intelligent Systems*, vol. 24, no. 6, pp. 83-89, 2009.

[3] M.-C. de Marneffe and C. D. Manning, The stanford typed dependencies representation. *Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pp. 1-8, 2008.

[4] M. J. Garcia-Godoy, I. Navas-Delgado, and J. Aldana-Montes, Bioqueries: a social community sharing experiences while querying biological linked data. *Proceedings of the 4th International Workshop on Semantic Web Applications and Tools for the Life Sciences*, SWAT4LS 2011.

[5] D. F. Barbieri, D. Braga, S. Ceri, E. D. Valle, and M. Grossniklaus, Querying RDF streams with C-SPARQL. *SIGMOD Record*, vol. 39, no. 1, pp. 20-26, Sep. 2010.

[6] Bobillo, F., Straccia, U., fuzzyDL: An expressive fuzzy description logic reasoner. *2008 International Conference on Fuzzy Systems (FUZZ-08)*, pages 923-930.

[7] Pollock, J.L. Defeasible reasoning with variable degrees of justification. *Artif. Intelligence* 133(1-2), pp. 232-282, 2001.

[8] Groza A. and Letia, I.A. Plausible Description Logic Programs for Stream Reasoning. *Future Internet* 4(4), pp. 865-881, 2001.

[9] Arnold, M. D. Stefan Gravenstein, M. D. Michael Elliott, P. Michael Colopy, and M. D. Jo Schweinle. Clinical Signs and Symptoms Predicting Infuenza Infection. *Archives of International Medicine*, Vol. 160, no. No. 21, pp. 3243-3247, Nov. 2000.

[10] E. Aramaki, S. Maskawa, and M. Morita,Twitter catches the flu: detecting infuenza epidemics using twitter,*Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '11, Association for Computational Linguistics, pp. 1568-1576, 2011.

[11] P. N. Mendes, A. Passant, P. Kapanipathi, and A. P. Sheth, Linked open social signals, *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Volume 01, pp. 224-231, 2010.

[12] D. Barbieri, D. Braga, S. Ceri, E. D. Valle, Y. Huang, V. Tresp, A. Rettinger, and H. Wermser, Deductive and inductive stream reasoning for semantic social media analytics, *IEEE Intelligent Systems*, vol. 25, no. 6, pp. 32-41, 2010.

[13] Mark Mysln; Shu-Hong Zhu, Wendy Chapman, Mike Conway, Using Twitter to Examine Smoking Behavior and Perceptions of Emerging Tobacco Products, *Journal of Medical Internet Research*, 15(8):e174, 2013.

Radu Balaj
Technical University of Cluj-Napoca
Department of Computer Science
Intelligent Systems Group
Baritiu 28, RO-400391, Cluj-Napoca, Romania
E-mail: *radu.balaj@student.utcluj.ro*

Adrian Groza
Technical University of Cluj-Napoca
Department of Computer Science
Intelligent Systems Group
Baritiu 28, RO-400391, Cluj-Napoca, Romania
E-mail: *Adrian.Groza@cs.utcluj.ro*

Third International Conference
Modelling and Development of Intelligent Systems
October 10 - 12, 2013
"Lucian Blaga" University
Sibiu - Romania

Theoretical and practical approaches for time series prediction

Alina Bărbulescu, Dana Simian

Abstract

The goal of this paper is to discuss two different modern approaches for modeling and prediction of time series – general regression neural networks and support vector regression. It is known that the performances of different approaches from machine learning field are strongly dependent on data. We apply and evaluate our methods on eight different real meteorological series. In order to increase the SVR performances we develop a method for obtaining a SVR optimal multiple kernel.

1 Introduction

Modeling the time series evolution is of main importance for the prediction of real life processes as temperature, precipitation, earthquakes, due to their economic and social implications, that could be dramatic (drought, disasters, famine etc). The main problems in modeling and prediction of this type of series are their non – linearity, high variability, correlation and/or persistence, making them inappropriate for the use of classical regression methods. Therefore, new methods, belonging to machine learning, artificial intelligence and optimization techniques, as artificial neural networks (ANN), gene expression programming (GEP), support vector regression (SVR) can be successfully used for this aim. It is known that generally a SVM based approach is strong dependent of data type. In [3] we have studied the problem of forecasting the meteorological time series. We used for this purpose an adaptive GEP algorithm, AdaGEP, and a ε-SVR algorithm with RBF kernel and we performed an empirical comparison of these methods on many series of temperature and precipitations from different meteorological stations in the Black Sea region. The comparison revealed that there is not one method with best results for all studied data series, but AdaGEP dominates SVR models in the most cases. It would be ideal to find a method capable to be used with almost the same level of performances for meteorological series obtained from any stations and for all variables taken into account (temperatures, precipitations etc.). This is a difficult task and it is not sure that it can be accomplished.
The aim of this paper is to analyze two models obtained for meteorological time series using General Regression Neural Networks (GRNN) and SVR. We detect models and we evaluate their performances for six different real meteorological series, using DTREG software [6]. We also propose a theoretical approach for improving the performances of SVR. This is a method for the choice of a multiple SVR kernel such that the measure of the prediction error is minimized. We consider two such kinds of measures: Mean Absolute Prediction Error and Mean Squared Error.

2 Problem model

2.1 Time series prediction

The problem we face with is the following one: given a set of measured values for some meteorological characteristics (temperature, precipitations) in a period of time, predict the future values based on the past values. Practical a number *d* of past values is chosen to predict the future one. The choice of *d* is not the object of this article. Usually additional information and experiments are necessary to accomplish this task.

The classical problem of time series prediction is: find a function *f* which predicts future values, in a given prediction horizon *p*, of the series $X_t = (x_t, x_{t-\tau}, ..., x_{t-(d-1)\tau})$, i.e. express $x_{t+p} = f(X_t)$.

The series studied are annual and monthly precipitations and temperature data collected at four main meteorological stations from Dobrudja region, Romania, presented in Table 1.

Series	Station	Type	Variable	Period
CAT	Constanta	Annual	Temperature	01.1965 – 12.2005
TAT	Tulcea	Annual	Temperature	01.1965 – 12.2005
SAT	Sulina	Annual	Temperature	01.1965 – 12.2005
JAT	Jurilovca	Annual	Temperature	01.1965 – 12.2005
CMP	Constanta	Monthly	Precipitation	01.1961 – 12.2009
SMP	Sulina	Monthly	Precipitation	01.1965 – 08.2007
CMT	Constanta	Monthly	Temperature	01.1961 – 12.2009
SMT	Sulina	Monthly	Temperature	01.1961 – 08.2003

Table 1. Data series

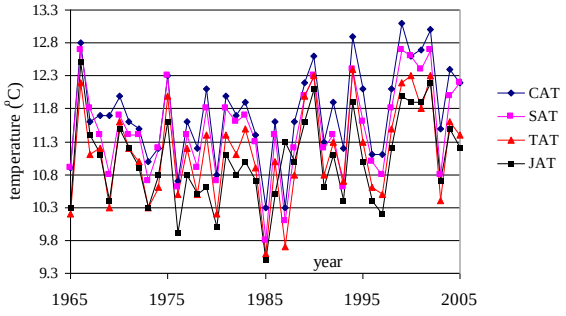


Fig.1. Series of annual temperatures

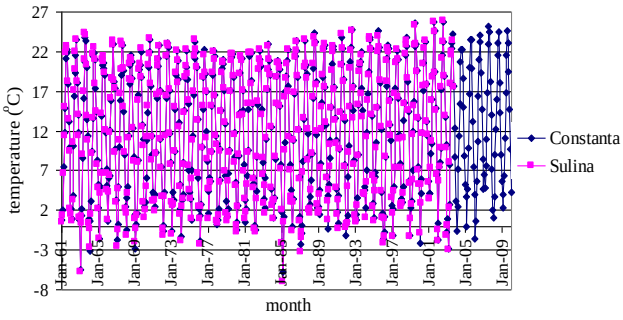


Fig.2. Series of monthly temperatures

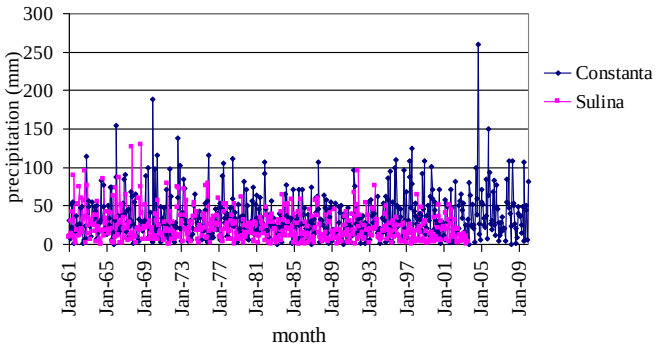


Fig.3. Series of monthly precipitation

The charts of series of annual temperatures are presented in Fig.1, those of monthly temperatures in Fig. 2 and of monthly precipitation in Fig. 3.

2.2 General Regression Neural Networks (GRNN)

GRNN introduced by Specht [14] is a feedforward network that allows the estimation of a dependent vector Y= (Y_i) from an independent vector **X**= (X_j) obtained by measurements. GRNN have the same architecture as the Probabilistic Neural Networks (PNN). The difference is that PNN act on categorical target variables and GRNN act for continuous target variables. That means that PNN performs classification while GRNN performs regression. PNN estimate the probability density function f(**X**,Y) for each class based on the training samples using Parzen or a similar probability density function. In both cases (PNN and GRNN) the operations are organized into a multilayered feed-forward network with four layers: Input layer, Pattern layer, Summation layer, Output layer [13]. In the case of GRNN the input layer contains one neuron for each predictor variable Y_i. The pattern layer contains one neuron for each case in the training data set, and stores the predictor variables values and target values [2]. The summation layer computes the numerator and the denominator of the estimator using two neurons: the numerator and the denominator summation neurons. Output layer contains one neuron that contains the result of division of the values in the numerator and the denominator of the previous layers. Both GRNN and PNN use nonparametric estimators of probability density function. The measure of how well each training sample can represent the position of prediction is the Euclidian or the city block distance between the training sample and the point of prediction [14].

2.3 Support Vector Regression (SVR)

SVR is a category of Support Vector Machines (SVM). SVM represents a powerful tool for solving learning tasks like classification and regression tasks. They are supervised learning methods introduced first by Vapnik [15]. The goal of SVM is to build a model, *f*, which predicts the output of a system depending of a set of variables, using a set of training data for which the output is known. The main characteristic of SVM is that the prediction function is expanded on a subset of support vectors as will be seen in relation (1). Support Vector based algorithms were extended from classification tasks to regression ones using various loss functions [1]. Traditional statistical regression techniques aim to minimize the deviation of *f(x)* from the known outputs for all training examples. The ε - SVR introduced in [15] uses the so called ε – insensitive loss function. It minimizes the generalized error bound instead of minimizing the observed training error, being based on the structural risk minimization principle. ε - SVR is searching for a function *f* that has at most ε deviation from the target outputs on all the training data and is as flat as possible. These requirements lead to a convex optimization problem. Next we present this formulation following the presentation from [15,1].
We consider first the case of a linear function *f:*

$$f(x) = \langle w, x \rangle + b;\ b \in \mathbb{R}, x \in X$$

where we denoted by *X* the space of all input instances and $\langle \cdot, \cdot \rangle$ represents the dot product in *X*.

Suppose that the training data are denoted by $(x_i, y_i) \subset X \times \mathbb{R}$, *i* =1, ..., *m*.

To take into account the possibility of an infeasible convex optimization problem we introduce the slack variables ξ, ξ^* and the problem formulation becomes:

$$\text{minimize } \left\{ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*) \right\},$$

subject to the constraints:

$$\begin{cases} y_i - \langle w, x_i \rangle - b \le \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases}$$

The dual optimization problem is

$$\text{maximize } \left\{ -\frac{1}{2}\sum_{i,j=1}^{m}\left(\alpha_i-\alpha_i^*\right)\left(\alpha_j-\alpha_j^*\right)\left\langle x_i,x_j\right\rangle-\varepsilon\sum_{i=1}^{m}\left(\alpha_i+\alpha_i^*\right)+\sum_{i=1}^{m}y_i\left(\alpha_i+\alpha_i^*\right)\right\}$$

subject to the constraints:

$$\sum_{i=1}^{m}\left(\alpha_i-\alpha_i^*\right)=0 \text{ and } \alpha_i,\ \alpha_i^*\in[0,C]$$

The function *f* can be rewritten as:

$$f(x)=\sum_{i=1}^{m}\left(\alpha_i-\alpha_i^*\right)\left\langle x_i,x\right\rangle+b \tag{1}$$

Relation (1) represents the so - called support vector expansion of the function *f*. The examples corresponding to non-vanishing coefficients are called Support Vectors.
The constant $C > 0$ and ε are parameters of the method. An improved SVR technique, named ν – SVR considers ε itself as a variable in the optimization process introducing a new parameter $\nu\in(0,1)$ (see [1]). The new parameter is more convenient than ε.

In order to solve the non-linear problem we make a projection ϕ of the input data X in a higher dimensional feature Hilbert space F. Using the "kernel trick" we can rewrite (1) for a non - linear function, without knowing explicitly the form of the mapping ϕ :

$$f(x)=\sum_{i=1}^{m}\left(\alpha_i-\alpha_i^*\right)K(x_i,x)+b \tag{2}$$

The kernel function *K* represents the inner product in the feature Hilbert space *F*. Several kernels can be used: linear, polynomial, RBF, sygmoidal (see [12]). These kernel functions are defined in Table 7. Other functions satisfying the Mercer's conditions can be used as kernel functions ([13]).

2.4 Settings and performance evaluation

For performance evaluation of the model obtained using GRNN and SVR we divided the data into two parts, one for training and one for validation. The last one contained 10 values for the annual series and 24 values for the monthly ones. Both algorithms were applied using the default settings of DTREG software [6], the number of predictor variables being 1 for the annual data and 2 for the monthly data. We implemented the ν – SVR, with RBF kernel.
The models performance have been analysed using as indicators the Mean Squared Error (MSE) and the Mean Absolute Prediction Error (MAPE) defined by:

$$MSE=\frac{\sum_{i=1}^{n}(x_i-x_i^*)^2}{n} \tag{3}$$

$$MAPE=\frac{\sum_{i=1}^{n}\left|\frac{x_i-x_i^*}{x_i}\right|}{n} \tag{4}$$

where x_i is the *i* - th registered value in the data series and x_i^* is the *i* - th value predicted by the model. We can observe that the MSE is a scale-dependent accuracy measure while the MAPE is scale independent. For models resulted by GRNN application, the correlation between the actual and the predicted value was also reported.

3 Results and discussions

The MSE for the models on the training and validation dataset are presented in Tables 2 and 3. For the first dataset, GRNN performed better than SVM. We remark the big differences of MSEs corresponding to GRNN and SVR on the training sets for TAT, SAT, JAT and CMP. For the validation datasets, MSE are comparable in the case of CAT, JAT. The SVR algorithm performs better on validation data in the case of the series TAT, JAT and SMP.
The MAPE for the models on the training and validation dataset are presented in Tables 4 and 5.
For the validation datasets, MAPE are generally comparable for GRNN and SVR. GRNN performs better in the case of CAT, CMP, CMT and SMT. In the case of training data GRNN significantly outperforms SVR.

Series	GRNN	SVR
CAT	0.066	0.39
TAT	0.000148	0.31
SAT	0.0006	0.27
JAT	0.034	0.39
CMP	6.45	793.85
SMP	327	373.02
CMT	4.08	5.79
SMT	3.95	5.6

Table 2.

MSE for the models on the *training* datasets

Series	GRNN	SVR
CAT	0.434	0.45
TAT	0.99	0.49
SAT	0.503	0.93
JAT	0.518	0.46
CMP	887.17	899.93
SMP	105.09	97
CMT	1.59	3.27
SMT	9.27	11.75

Table 3.

MSE for the models on the *validation* datasets

The correlations between the actual and predicted values in the experiments performed with GRNN are presented in Table 6. GRNN performed better on the training datasets and in five of eight cases on the validation datasets. The exceptions are for CMT and SMT, for which the registered performances are very high

Series	GRNN	SVM
CAT	0.482	4.14
TAT	0.021	3.76
SAT	0.041	4.39
JAT	0.322	4.16
CMP	5.55	267.22
SMP	244.33	209.62
CMT	82.45	107.99
SMT	49.14	60.34

Table 4.

MAPE for the models on the *training* datasets

Series	GRNN	SVM
CAT	4.8	4.88
TAT	6.87	5.12
SAT	7.58	5.15
JAT	5.52	5.21
CMP	580.34	594.75
SMP	247.09	197.5
CMT	24.36	35.78
SMT	244.03	250.94

Table 5.

MAPE for the models on the *validation* datasets

Series	Training		Validation	
	GRNN	SVR	GRNN	SVR
CAT	0.913101	0.207317	0.312878	0.31407
TAT	0.999859	0.623083	-0.394385	0.250404
SAT	0.893635	0.605498	0.250289	-0.247702
JAT	0.958707	0.183927	0.332102	0.349899
CMP	0.996158	0.237336	0.078309	0.049084
SMP	0.394746	0.256092	0.206757	0.250852
CMT	0.96722	0.953139	0.986575	0.971232
SMT	0.969537	0.956431	0.968031	0.961067

Table 6. Correlations between actual and predicted values

We observe that the performances are highly dependent of data. The size and the structure of the training set strongly influences the modeling and forecasting.
Taking into account that SVR underperforms on the validation data set in most of the cases we conclude that SVR overfit the training examples. Possible causes of overfitting phenomenon could be the choice of parameters C and ν or the choice of the kernel. The choice of parameters C and ν in SVR was performed doing a grid search in a 10-fold cross-validation procedure in order to avoid the overfitting [7]. This leads us to the conclusion that the use of a single kernel is not capable to generate an accurate model and more complex kernel must be used.

A possible solution for obtaining best results using SVR for time series prediction is proposed in the next section.

4 Optimal SVR multiple kernels for time series prediction

In [10] we introduced a general frame for building optimal kernels for Support Vector Classification. We implemented many particular methods derived from this frame for data sets Leukemia and Vowel from the standard libsvm package [4] and the results were promising.
In the following we propose a new method for obtaining optimal multiple SVR kernels for time series prediction. The idea is to create a SVR model based on a multiple kernel in order to obtain better prediction results. Multiple kernels are built using the simple standard kernels presented in Table 7 and the set of operations {+, *, *exp*} which preserve Mercer's conditions (see [9] for many details on Mercer's Theorem).

Kernels	
Polynomial: $K_{pol}^{r,d}(x_1, x_2) = (x_1 \cdot x_2 + r)^d; r, d \in \mathbb{Z}_+$	(5)
RBF: $K_{RBF}^{\gamma}(x_1, x_2) = \exp\left(\frac{-1}{2\gamma^2}\lvert x_1 - x_2\rvert^2\right); \gamma \in \mathbb{R}_+$	(6)
Sigmoidal: $K_{sig}^{\gamma}(x_1, x_2) = \tanh(\gamma \cdot x_1 \cdot x_2 + 1); \gamma \in \mathbb{R}_+$	(7)

Table 7. Standard single kernels

Following the models implemented in [10] we proposed a multiple kernel composed by 4 single kernels, but our approach is not restricted to this number of single kernels. The tree formal representation of the multiple kernel K= (K_1 op_2 K_2) op_1 (K_3 op_3 K_4), is given in Fig. 4.

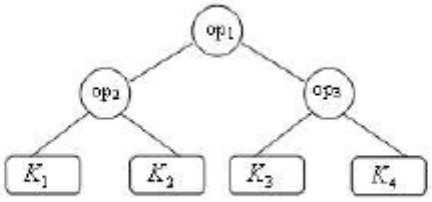


Fig. 4. – Formal representation of multiple kernel [10]

The choice of a multiple kernel supposes the choice of the type of the single kernels and the choice of the corresponding parameters.

To construct the multiple kernels we use an evolutionary method structured on two levels: the macro and the micro levels. In the macro level the multiple kernel is built using a genetic algorithm. The multiple kernels are coded into chromosomes. The chromosomes quality is computed in the micro level using a SVR algorithm. The main difference from the classification case is that we have to additional constants to be appropriately chosen: C and ε (or ν depending on the SVR approach). We propose two solutions for the choice of these parameters in the end of this section.

As structural representation of the multiple kernel we use a linear representation given in Fig.5. For coding the multiple kernel we use 78 genes: 6 genes for operations (2 genes for each operation op_i, i = 1, 2, 3); 6 genes for kernel types (2 genes for each type t_i, i = 1, 2, 3); if the single kernel K_i is polynomial we use 4 genes for the degree parameter d_i and 12 genes for r_i; if the single kernel K_i is not polynomial we use 16 genes to represent the real value v_i..

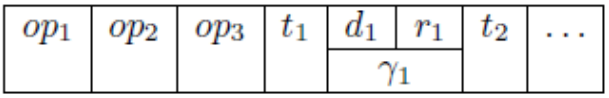


Fig. 5. Linear representation of multiple kernel [10]

In order to evaluate the quality of the chromosomes we use a SVR algorithm acting on a particular set of data. In order to make the evaluation we divide the dataset into two subsets: the training subset and the test subset. The training subset is used for problem modeling and the test subset is used for evaluation. The training subset is also randomly divided in two subsets: the learning and the validation subsets. The learning subset is used for training the SVR algorithm and the validation subset is used for computing the Mean Absolute Prediction Error (MAPE) defined in (4), which represents the fitness function for evaluation of chromosomes.

The performance of our predictive model based on the multiple kernel given by the genetic algorithm is evaluated using cross validation method.

There is two possibility for the choice of parameters C and ε (or ν) from the SVR algorithm. The first one is to adapt a grid choice of parameters which imply to run our method for each pair in the grid and then to chose the best triplet (C, ε, optimal multiple kernel). This method is huge time consuming in the training step. Due to the independence of the computation in each point of the grid it could give good results in the case of parallelization. The second method includes the

two parameters in the chromosome representation such that a choice of a multiple kernel is made together with a choice of the pair (C, ε).

Implementation and practical tests are necessary in order to validate our theoretical proposed method.

5 Conclusions and further directions of work

In this article we studied the performance of modeling meteorological time series using two different approaches – GRNN and SVR. It has been seen that the results are strong dependent on datasets. In terms of Mean Squared Error, GRNN performed better in all cases for the training sets and in five from eight cases for validation sets. Since we needed an invariant measure of performances, we also used Mean Absolute Prediction Error. In this case, GRNN performed better in seven from eight cases for the training sets and in four cases for validation sets. For two series (CMT and SMT) the correlation between predicted and actual values is very good both for training and validation datasets. For the other six series the correlation is under 0.5 and in the case of TAT validation data set for GRNN and SAT validation data set for SVR we obtained a negative correlation. The results suggest us the necessity of a more complex model, if possible a flexible one, able to fit well to different series of meteorological data. The proposed theoretical solution is a new SVR approach using an optimal multiple kernel. The optimal multiple kernel is obtained using an evolutionary algorithm structured on two levels. Further works will be oriented to the implementation and testing of this method on different meteorological datasets in order to validate it.

References

[1] D. Basak, S. Pal, D. C. Patranabis, Support Vector Regression, Neural Information Processing – Letters and Reviews, Vol. 11, No. 10, pp. 203 – 224, 2007.

[2] A. Bărbulescu, L. Barbeş, Mathematical models for inorganic pollutants in Constanța area, Romania. Revista de chimie vol. 64(7), pp. 747 - 753, 2013.

[3] E. Băutu, A. Bărbulescu. Forecasting meteorological time series using soft computing methods. Appl. Math. Inf. Sci., 7, No. 4, pp. 1297 – 1306, 2013.

[4] C-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, 2001, Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[5] H. Drucker, C. J.C. Burges, L. Kaufman. A. Smola, V. Vapnik, Support Vector Regression Machines, Neural Information Processing Systems, C.J. Burges, L. Kauffman, A. Smola, V. Vapnik (Eds.), MIT Press, pp. 155 – 161, 1997.

[6] DTREG, Software For Predictive Modeling and Forecasting, http://www.dtreg.com/

[7] C.-W. Hsu, C.-C. Chang, C.-J. Lin, A practical guide to support vector classification. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, 2003.

[8] H. N. Nguyen, S. Y. Ohn, W. J. Choi, Combined kernel function for support vector machine and learning method based on evolutionary algorithm, Neural Information Processing, 11th International Conference, ICONIP 2004, N. R. Pal et al. (Eds.), LNCS, vol. 3316 , Springer, pp. 1273 – 1278, 2004.

[9] Ha Quang Minh, Partha Niyogi, Yuan Yao, Mercers Theorem, Feature Maps, and Smoothing, http://people.cs.uchicago.edu/ niyogi/papersps/MinNiyYao06.pdf

[10] D. Simian, F. Stoica, A general frame for building optimal multiple SVM kernels, Lecture Notes in Computer Science, Vol. 7116, Large Scale Scientific Computation, I. Lirkov et al. (Eds.), pp. 256 – 263, 2012.

[11] D. Simian, F. Stoica, C. Simian, Optimization of Complex SVM kernels using a hybrid algorithm based on wasp behaviour, Lecture Notes in Computer Science, vol. 5910, I. Lirkov, S. Margenov, and J. Wasniewski (Eds.), pp. 361 – 368, 2010.

[12] A. J. Smola, B.Schölkopf, A tutorial on support vector regression, Statistics and Computing, 14, Kluwer Academic Publishers, pp. 199 – 222, 2004.

[13]D. F. Specht, "Enhancements to Probabilistic Neural Networks", International Joint Conference on Neural Networks, vol. I, pp. 761 – 768, June 1992.
[14]F. Sprecht. A General Regression Neural Network. IEEE Transactions on Neural Networks, vol. 2, no. 6, pp. 568 – 576, Nov. 1991.
[15]V. Vapnik, The Nature of Statistical Learning Theory, Springer Verlag, 1995, http://www.csie.ntu.edu.tw/cjlin/libsvmtools/datasets

BĂRBULESCU Alina
Doctoral School of Civil Engineering
Technical University of Civil Engineering
Bucharest
ROMANIA
alinadumitriu@yahoo.com

SIMIAN Dana
Lucian Blaga University
Sibiu
ROMANIA
dana.simian@ulbsibiu.ro

Third International Conference
Modelling and Development of Intelligent Systems
October 10 - 12, 2013
"Lucian Blaga" University
Sibiu - Romania

A Better Genetic Representation of a Fuzzy Controller Enabling the Determination of Its Parameters with the Help of a Genetic Algorithm

Stelian Ciurea

Abstract

Since 1975, fuzzy controllers have fully proved their usefulness in the most diverse applications. The design of such a controller involves setting up inference rules and values for a large number of parameters. There are situations where this is possible either through the expertise of a human operator or through a knowledge stock. If we cannot rely on such information, genetic algorithms are a good alternative to determine these values. The first condition in solving a problem by means of a genetic algorithm is the genetic representation of the solution. In this paper, we present the genetic algorithm that we designed with a view to determining the parameters of a fuzzy controller for the Truck Backer-Upper Problem (this problem is considered an acknowledged benchmark in nonlinear system identification). The genetic representation used in this algorithm belongs to us.

1 The Fuzzy Controller for the Truck Backer-upper Problem

1.1 The Truck Backer-Upper Problem

This problem, made famous by [3], has been investigated by many researchers. On the other hand, it is difficult not to notice that nearly anyone is able to drive the truck to the desired position if given some time to get used to the controls.

The truck corresponds to the cab part of the Nguyen-Widrow's truck and trailer, referred to as the simplified Nguyen-Widrow problem. The truck position is determined by the three state variables $x \in [-50, 50]$, $y \in [0, 80]$ and $\varphi \in [-90°, 270°]$ - the angle between the truck's onward direction and the x-axis (Fig. 1). The width and length of the truck are 5 and 2 meters, respectively.

The truck sets out from an initial position with the three state variables xi, yi and φi and must reach the loading dock with $xf = 0$, $yf = 0$, $\varphi f = 90°$. The truck only moves backwards with the fixed speed. To control the truck at every stage, an appropriate steering angle $\theta \in [-45°,45°]$ must be provided. Thus, the controller is a function of state variables $\theta = f(x, y, \varphi)$.

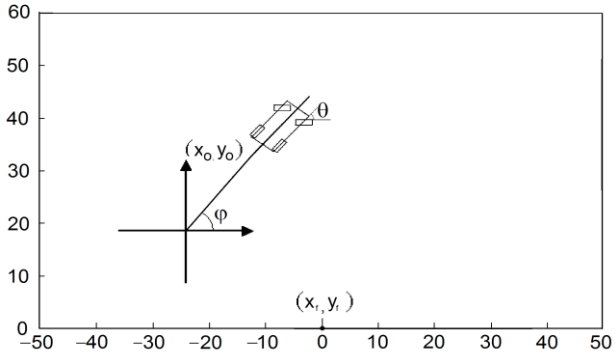


Figure 1. Truck backer-upper system

Typically, it is assumed that there is enough clearance between the truck and the loading dock so that the truck position coordinate y can be ignored, simplifying the controller function to: $\theta = f(x, \varphi)$. For obvious reasons, such a controller does not perform very well if the distance between the truck position and the loading dock is small.
The movements of the truck are described by the following system of equations:

$$\begin{cases} \dot{x} = -v\cos\varphi \\ \dot{y} = -v\sin\varphi \\ \dot{\varphi} = -\frac{v}{l}\tan\theta \end{cases} \quad (1)$$

where l is the length of the truck and v is the backing up speed of the truck. These equations are applied to the current state, and the truck moves on until one of the following stopping conditions is met:

- $y \leq 0$ (the truck reached the loading dock);
- x, y or φ have an unacceptable value: $y>100$, $x \notin [-50, 50]$ or $\varphi \notin [-90°, 270°]$

1.2 The Fuzzy Controller

We implemented a Mamdani-type fuzzy controller. The input data are *x* and φ, and the output data is the steering angle θ. For *x*, we have opted for 5 fuzzy sets with the following linguistic variables: left - LE, left center - LC, center - CE, right center- RC, and right – RI. For φ, we have settled on 7 sets: RB (right below), RU (right upper), RV (right vertical), VE (vertical), LV (left vertical), LU (left upper), and LB (left below). For θ, we have also selected 7 sets: NL (negative large), NM (negative medium), NS (negative small), ZE (zero), PS (positive small), PM (positive medium), and PL (positive large). The membership functions we have employed are trapezoidal or triangular (Fig. 2, Fig. 3 and Fig. 4).

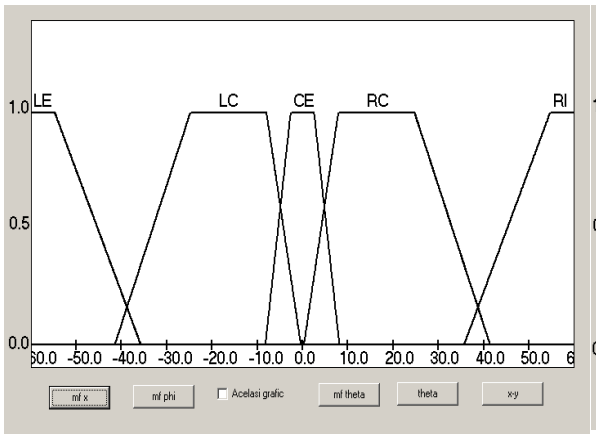


Figure 2. Example of trapeziodal for variable *x*.

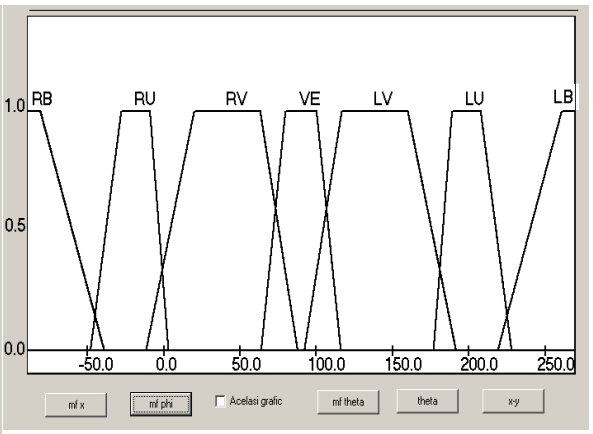


Figure 3. Example of trapezoidal membership functions for variable φ.

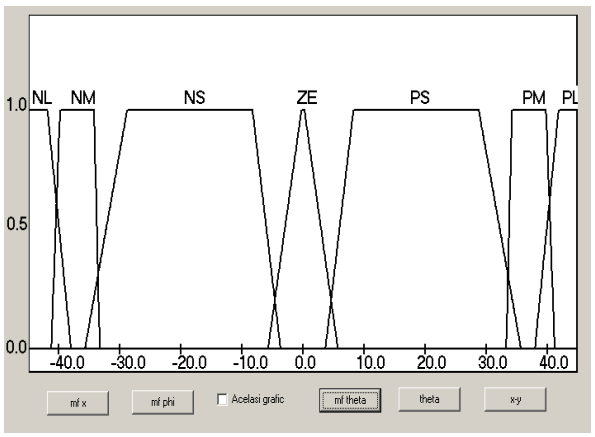


Figure 4. Example of trapezoidal and triangular membership functions for θ.

Starting from these sets, a fuzzy controller needs 5×7=35 inference rules based on fuzzy arguments. For example, if the *x* position is right centre, and the angle φ is vertical, then we want to steer positive medium. Symbolically, IF x is RC AND φ is VE, THEN θ is PM. Table I illustrates a possibility of defining these 35 rules.

Table I: Matrix of the Rules for the control of the Truck Backer-Upper System:

φ / *x*	**RB**	**RU**	**RV**	**VE**	**LV**	**LU**	**LB**
LE	NL	NL	NL	NM	NM	NS	PS
LC	NL	NL	NM	NM	NS	PS	PM
CE	NM	NM	NS	ZE	PS	PM	PM
RC	NM	NS	PS	PM	PM	PL	PL
RI	NS	PS	PM	PM	PL	PL	PL

For the two input data variables – *x* and φ, the membership function is calculated for each of the 35 rules; we group the two results of these memberships function by means of an AND fuzzy operation, and then by means of an implication operation we obtain fuzzy sets for θ. We will have 35 θ sets corresponding to the 35 rules. These are grouped by means of the aggregation operation

to get the output θ-set. Then, to find the actual control value, we must convert the output fuzzy set into a numerical value for θ by means of the defuzzification operation. There are various formulae for the fuzzy AND, implication, aggregation and defuzzification operations. It follows that, in order to fully define fuzzy controller, we need 35 inference rules and parameters that define the type and positioning of the fuzzy functions on the universe of discourse axis for the three variables (x, φ, θ), as well as the implementation of the fuzzy operations that occur in the calculating the response of the controller. There are no mathematical formulae to provide the values for these parameters.

3 The Genetic Algorithm

Genetic algorithms belong to the category of probabilistic algorithms. Such an algorithm starts from a set (population) of possible solutions (chromosomes). The performance of each chromosome is calculated by means of an evaluation function which appraises the accuracy of the solution provided by that chromosome to the studied problem. The new population is formed by selecting the fitter individuals. Some members of the new population recombine by means of "genetic" operators to form new solutions. There are unary transformations like mutations, which create new individuals by a small change in single individual and binary transformations, such as the crossover, which create new individuals by mixing traits from the two parents. After a number of iterations (generations) the search converges and is successful if the best individual obtained at a given time represents the optimum solution.

3.1 Genetic representation of the controller

The purpose of the genetic algorithm we have designed is to determine the parameters of the fuzzy controller optimal for solving the truck backer-upper problem. Because of the intrinsic symmetry of the problem, we have selected the member functions that are symmetrical to the median axis of the universe of discourse for each of the three variables. Thus, in a chromosome, for x, we have retained parameters for variables CE, RC and RI; variable LE mirrors RI symmetrical to axis x=0 and LC mirrors RC. Similarly, for φ we have represented parameters for variables VE, LV, LU and LB, and for θ, variables ZE, PS, PM and PL.
We coded the following parameters that typify a fuzzy controller:

- The interference matrix as a matrix with 5 rows and 7 columns, where each item can range between 1 and 7 corresponding to the 35 interference rules (1 means NL, 2 means NM, etc.);
- the type of fuzzy operations through 5 integer numbers for the type of fuzzy operations AND (0=min, 1=product), OR (0=max, 1=a+b-ab), the type of implication operation (0=min, 1=product), the type of the aggregation operation (0=max, 1=sum, 2=a+b-ab) and the type used in the defuzzification operation: a value between 0 and 3 corresponds to the methods based on integral calculus, while values between 4 and 6 are for elitist methods;
- The shape and position of the fuzzy sets in the universe of discourse. We point out that we chose the trapezoidal for the trapezoidal membership functions corresponding to these sets (we assumed that the triangles are special cases of trapeziums with smaller bases of negligible length).

3.1.1 Representing classical trapezoidal membership functions

With the release of MATLAB 6, the "Fuzzy Logic Toolbox" library has been implemented [14]. Within it, a number of membership functions were defined for fuzzy sets, among which trapezoidal and triangular-shaped ones. Due to the popularity of MATLAB, the method by which

fuzzy sets are represented in this software has become traditional, and most papers determining the parameters of fuzzy controllers by means of genetic algorithms use it. Thus, a trapezoidal membership function is represented with the help of four parameters, marked a, b, c and d, with the mathematical expression (2) and shape illustrated in Figure 5:

$$f(x;a,b,c,d) = \begin{cases} 0 & x \le a \text{ or } x \ge d \\ \dfrac{x-a}{b-a} & a \le x \le b \\ 1 & b \le x \le c \\ \dfrac{d-x}{d-c} & c \le x \le d \end{cases} \quad (2)$$

where

$$a < b \le c < d \quad (3)$$

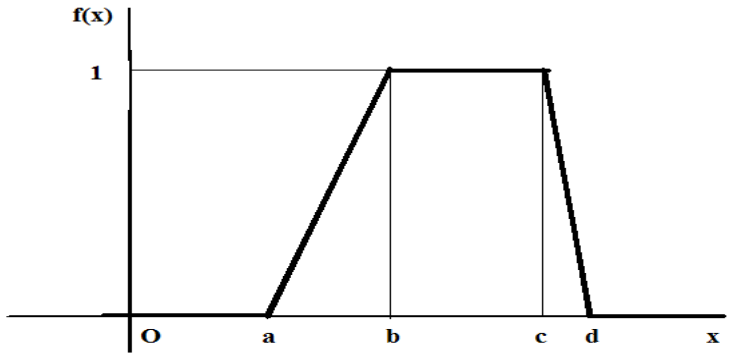


Figure 5. Trapezoidal membership function.

This representation is useful when we do the calculations required to determine the value of the output size of the controller or when we graphically represent memberships functions. However, it is not helpful when we apply the specific operations of the genetic algorithm because we need to introduce a number of restrictions where in the case of both mutation and crossover, so that for the resulting individuals relationship (3) is maintained, which burdens and slows down the genetic algorithm. We implemented this method of representation by means of the structure termed **cromozom**:

```
struct cromozom
{  double parammfx[5][4];      /* parameters for the 5 memberships functions of x */
   double parammfphi[7][4];    /* parameters for the 7 memberships functions of phi*/
   double parammfteta[7][4];   /* parameters for the 7 memberships functions of theta*/
   int typeand, typeor, typeimplication, typeagregation, typedefuzz, rules[5][7] ;
};
```

3.1.2 Our own representation of trapezoidal membership functions

We implemented the representation of trapezoidal membership functions as follows:

- for the membership functions for *x* 9 real parameters:
  - 3 values representing the ratio between the larger base of the trapezium and the average value of the universe of discourse – a value within the range [0.25 ; 2.0] for variables CE, RC and RI.
  - 2 values representing the percentage of the larger base of the trapezium overlapping the larger base of the trapezium placed on its left - a value within the range [0.05 ; 0.4] for pairs CE and RC, and respectively RC and RI

- 3 values representing the ratio smaller base/larger base – a value within the range [0.01; 0.65] for CE, RC and RI.
- a value representing the position of the smaller base for CE – a value within the range [0.05 ; 0.95] out of the available interval calculated depending on the large basis for RC (for RI the shape of the function is a rectangular trapezium, and for CE the trapezium is isosceles).

- 13 real parameters that are determined in a similar manner, but adding one value to each parameter to represent the functions for the other two variables (φ and θ).

The advantage of this representation is that, regardless of the method chosen for the operation of mutation or crossover, the resulting values will belong to the intervals considered, so no further validation tests are needed. In our application, we used a structure called **cromozom2** corresponding to this way of representation:

```
struct cromozom2
{  int typeand, typeor, typeimplication, typeagregation, typedefuzz, rules[5][7] ;
   /* memberships functions for x*/
   double rapx[3];        /* ratio larger base / average value of the universe of discourse */
   double suprax[2];      /* percentage of the overlapping of larger bases: 0.05-0.4 */
   double procx[3];       /* ratio smaller base/larger base: 0.05-0.65*/
   double pozrelx;        /* position of the smaller: 0.05-0.95 */
   double rapphi[4], supraphi[3], procphi[4], pozrelphi2,pozrelphi1;     /*rapphi1 results */
   double raptheta[4], supratheta [3], proctheta [4], pozreltheta2,pozreltheta1;
};
```

3.1.3 Conversion from our own representation of transfer functions in their traditional form

In our application, we used our own representation for the operations that are specific to the genetic algorithm, whereas for simulating fuzzy controllers, for the graphic representation and for the output files generated by the application, we used the traditional form. It was necessary to implement a function converted into the traditional. For reasons of space, we only present the conversion for member functions of input x (those for φ and θ are similar).

```
struct cromozom conversie(struct cromozom2 c2, double xmin, double xmax, double phimin, double
phimax, double tetamin, double tetamax)
   {  struct cromozom c1;
      double bazamare[7], a[7][4], bazamica; int i,j;
      c1. typeand = c2. typeand; c1. typeor = c2. typeor; c1. typeimplication = c2. typeimplication;
      c1. typeagregation = c2. typeagregation; c1. typedefuzz = c2.typedefuzz;
      for (i=0;i<5;i++)
            for (j=0;j<7;j++)
                  c1.rules[i][j] = c2.rules[i][j];
      double medianax = (xmax + xmin)/2; double mediax = (xmax - xmin) / 5;
      c1.parammfx [2][3] = medianax + mediax * c2.rapx[0] / 2;                    /* CE */
      c1.parammfx [2][2] = medianax + c1.parammfx [2][3] * c2.procx[0];
      c1.parammfx [2][0] = medianax - c1.parammfx [2][3];
      c1.parammfx [2][1] = medianax - c1.parammfx [2][2];
      bazamare[2] = a[2][3] - a[2][0];
      c1.parammfx [4][0] = xmax - mediax * c2.rapx[2];                            /* RI */
      bazamare[4] = xmax - c1.parammfx [4][0];
      c1.parammfx [4][1] = xmax - bazamare[4] * c2.procx[2];
      c1.parammfx [4][2] = xmax + xmax/4;
      c1.parammfx [4][3] = xmax + xmax/2;
      c1.parammfx [3][0] = c1.parammfx [2][3] - bazamare[2] * c2.suprax[0];   /* RC */
      c1.parammfx [3][3] = c1.parammfx [4][0] + bazamare[4] * c2.suprax[1];
      bazamica = (c1.parammfx [3][3] - c1.parammfx [3][0]) * c2.procx[1];
      c1.parammfx [3][1] = c1.parammfx [3][0] + (c1.parammfx [3][3] - c1.parammfx [3][0] - bazamica) *
c2.pozrelx;
      c1.parammfx [3][2] = c1.parammfx [3][1] + bazamica;
      for (i=0;i<=1;i++)                                                          /* LE(i=0) and LC(i=1) */
            for (j=0;j<=3;j++)
                  c1.parammfx [i][j] = medianax - c1.parammfx [4-i][3-j];
      ….
      return c1 ;}
```

3.2 The Genetic Algorithm

In the genetic algorithm that we have implemented, one chromosome is a controller and it is a structure comprising a matrix with five rows and seven columns of integers values, 5 integer values and 35 real values that correspond to the parameters described in the previous paragraph. The genetic algorithm attempts to determine the optimum values of all these 75 parameters.

Input data:

for the fuzzy controller: the length and speed of the truck, the minimum and maximum values for x, y, φ, θ and set of fuzzy rules;

for the proper genetic algorithm: the number of ages, the number of chromosomes, the selection method (we have implemented three methods: Monte Carlo, "Tournament", and Michalewicz [5]), the probability of crossover, the probability of selecting a chromosome for the mutation, the probability of mutation for a gene, the number of tests for the fitness of chromosome and the three state variables x0, y0 and φ0 for each test. To assess a chromosome, we have simulated its route from the initial position to its final position (xf, yf, φf), and we compute the fitness using the following function:

$$fitness = \sum_{\text{test cases}} [2x_f^{\,2} + y_f^{\,2} + 5(\varphi_f - \frac{\pi}{2})^2] \tag{4}$$

Since the aim of the controller is to bring the truck to the coordinates point (0, 0, π/2), the function we have used is a penalizing one in relation to each of the three parameters that characterize the final state of the controlled system: the lower the value of the function, the better the chromosome.

Output data: the average performance of the population and the performance of the best chromosome following each generation; the parameters of all the controllers represented by the last generation chromosomes.

The implemented algorithm:

```
1. Read initial data
2. Initialize the parallel work mode
3. If the process is root then
      Randomly generate the initial population
4. For each age do
5.    If the process is root then
6.       For id←1, nr_procs do
7.             Send to the id process the data needed to assess the id chromosome
8.       For id←1, nr_procs do
9.          Receive fitness value calculated for the id process
10.      Selection; Crossover; Mutation
11.      If it is the last generation then
12.         Write the output data          //in text files
13.   Else                                 //the process is slave
14.      Receive the data needed to assess the chromosome
15.      For each assessment test do
16.         Compute the fitness
17.      Send the fitness value calculated at the root
18. Close the parallel work mode;
19. Stop
```

4 Experimental Results

We wrote the application in the C language, and we have used the mpich2-1.4.1 library for the parallel mode. We have run the application on an Intel HPC System, located at Lucian Blaga University, in Sibiu, comprising 14 nodes, each equipped with 4 Dual Core Intel Xeon processors. As parameters of the genetic algorithm, we have used the following: number of generations = 400, number of chromosomes = 200, crossover probability = 0.50, probability of selecting a chromosome for mutation = 0.25 and probability of mutation for a gene = 0.05. For the truck, we have chosen the following characteristics: length l = 5 m and backing up speed v = 1.4 m/s. The genetic algorithms with the parameters thus established were run for 14 randomly generated populations.

To assess the controllers we have obtained, we have used other 60 items of test data. The best controllers obtained in each of the 14 cases have been assessed with these tests. As assessment function, we have used the following measure of error from [6]:

$$\varepsilon = \sum_{i=1}^{11}\left(\left|x_{fi}\right| + \frac{0.4}{15}\left|90 - \varphi_{fi}\right|\right) \tag{5}$$

where x_{fi} and φ_{fi} are the values of x and φ in the final state from each of the 60 tests.
In Table II we present the average error of the best controllers resulted from the Genetic Algorithms using the three methods of selection we mentioned above.

Table II: Average error of the best controllers

Initial pop.	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14
Monte-Carlo	1.10	1.46	1.41	1.28	1.44	1.21	1.16	0.92	1.27	2.06	1.08	1.27	0.96	0.92
Tourna-ment	0.79	1.21	**0.60**	1.44	1.05	1.21	1.54	1.21	2.26	0.66	1.29	0.98	0.79	1.04
"Micale-wicz"	1.20	1.02	1.51	1.30	1.31	0.87	2.0	1.23	1.13	0.80	1.38	1.25	**0.60**	1.29

The results of each simulation for these tests were included into one of following three categories [9]: Good if $\varepsilon \leq 0.4$; average if $\varepsilon > 0.4$ but the controller led the truck to the loading dock (y=0); missed if the truck failed to reach the loading dock. Table III presents the number of tests for which the result was "good". The number of "missed" tests was zero for each best controller.

Table III: Number of "good" tests

Initial pop.	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14
Monte-Carlo	8	10	8	8	0	6	4	11	6	6	10	8	12	14
Tourna-ment	12	6	**30**	4	8	13	4	8	0	24	6	14	22	10
"Micale-wicz"	4	12	2	8	8	12	2	7	2	16	8	0	28	6

Fig. 6 illustrates the average fitness of the population that provided the best controller. Fig. 7 illustrates the fitness of the best chromosome obtained.

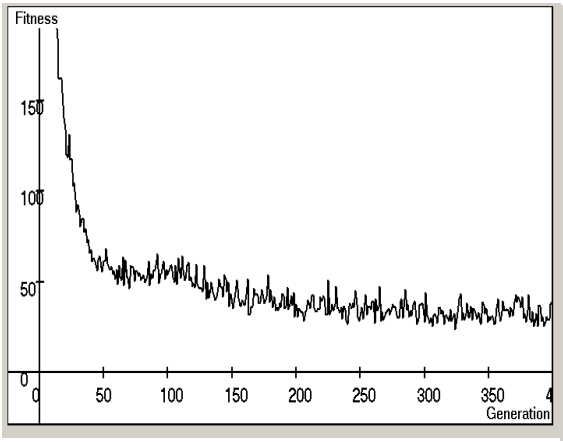


Figure 6. The average fitness of the population #1 "Tournament"

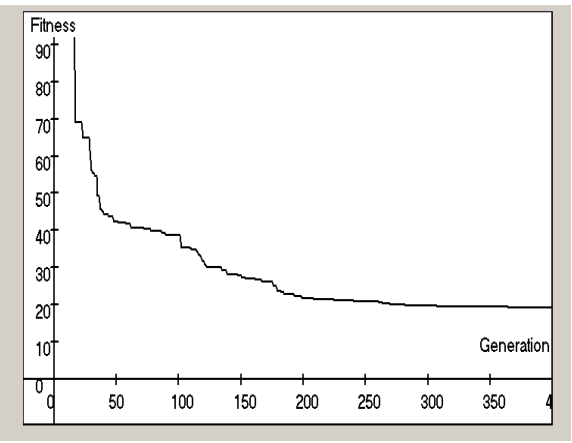


Figure 7. Best fitness of the population #3, "Tournament"

For 6 of the 60 starting items of test data we have illustrated in figure 8, the trajectories obtained by simulating the behavior of the best controller obtained.

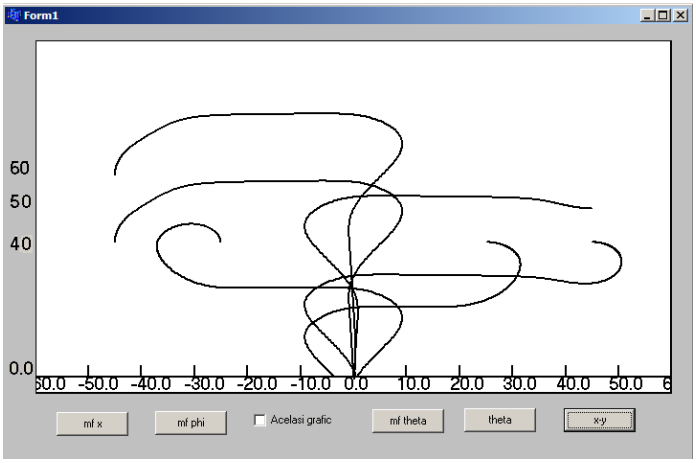


Figure 8. Truck trajectories for 6 initial positions obtained with best

The parameters of the best controller obtained are the following: product for fuzzy-AND, maximum for fuzzy-OR, product for the fuzzy implication, sum for the aggregation of the fuzzy rules and defuzzification using the sub-unitary weighted centroid method. The member functions for this controller are those represented in Fig. 2, Fig. 3 and Fig. 4. The Matrix of the Rules is similar to the one illustrated in Table I.

5 Conclusion

We can notice that:

The genetic algorithm proved its efficiency in all 42 tests we have performed. Thus, the ratio between the average fitness of the initial population and the average fitness of the final population equaled:

- 18.96 with the "Monte Carlo" method;
- 47.98 with the "Tournament" method;
- 46.03 with the "Micalewicz" method;

The ratio between the fitness of the best controller of the initial population and the fitness of the best controller of the final population equaled:

- 6.29 with the "Monte Carlo" method;
- 8.73 with the "Tournament" method;
- 7.83 with the "Micalewicz" method;

The best results have been obtained in the case of the algorithms that have used the "Tournament" selection method. The error of the best overall chromosome is 0.598 (coresponding to the best chromosome obtained from the initial population #3).

References

[1] Masood Anzar, Mohammad Fazle Azeem, Tanveer Chauhan & Anil Kumar Yadav, Generalized Approach for GA Based Learning of FLC Design Parameters, IICPE-2010
[2] Danilo Pelusi, Optimization of a Fuzzy Logic Controller using Genetic Algorithms, 2011 Third International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC) 2011
[3] D. Nguyen, B. Widrow, "The truck Backer Upper: An example of self learning in neural networks, in Neural networks for Control", The MIT Press, Cambridge MA, 1990.
[4] Z. Michalewicz, "Heuristic Methods for Evolutionary Computation Techniques", Journal of Heuristics, Vol.1, No.2, 1995, pp.177-206
[5] Z. Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs", Springer, 1998
[6] A. Riid, and E. Riistem, "Fuzzy logic in control: Truck Backer-Upper problem revisited," Proc. IEEE 10" Int. Conf. Fuzzy Systems, Melboume, Vol. 1, pp. 513-516, 2001.
[7] J. R. Koza, "A Genetic Approach to The Truck backer-upper Problem and the Inter-Twined Spiral Problem", IJCNN Intl. Conf. on Neural Networks, vol. 4, pp. 310-318, NY: IEEE Press, USA, 1992.
[8] S. Ciurea, I. Mihut, „Fuzzy Controller Design Using Genetic Algorithm Optimization", 8th International Symposium on Automatic Control and Computer Science - SACCS 2004, October 22 - 23, 2004, Iasi, Romania.
[9] S. Ciurea, „Determining the Parameters of a Sugeno Fuzzy Controller Using a Parallel Genetic Algorithm", Proceedings of the „9th International Conference on Control Systems and Computer Science, University Politehnica of Bucharest, Romania, 2013", ISBN 798-0-7685-4980-4, pg 36-43.
[10] Pintu Chandra Shill, Kishore Kumar Pal, Md. Faijul Amin, Kazuyuki Murase, "Genetic Algorithm Based Fully Automated and Adaptive Fuzzy Logic Controller", IEEE International Conference on Fuzzy Systems, June 27-30, 2011, Taipei, Taiwan
[11] Zhi-Long Wang, Chih-Hsiung Yang, Tong-Yi Guo*, The Design of An Autonomous Parallel Parking Neuro-Fuzzy Controller for A Car-like Mobile Robot, SICE Annual Conference 2010, Taipei, Taiwan
[12] http://www.mpitutorial.com.
[13] http://www.open-mpi.org/doc
[14] "Fuzzy Logic Toolbox™ User's Guide R2011b", The MathWorks, Inc., 2011.

STELIAN CIUREA
"Lucian Blaga" University of Sibiu
Faculty of Engineering, Department of Computer and Electrical Engineering
E. Cioran Str, No. 4, Sibiu-550025, ROMANIA,
E-mail: stelian.ciurea@ulbsibiu.ro

Third International Conference
Modelling and Development of Intelligent Systems
October 10 - 12, 2013
"Lucian Blaga" University
Sibiu - Romania

Splitting the structured paths in stratified graphs

Dănciulescu Daniela, **Nicolae Ţăndăreanu**

Abstract

The concept of stratified graph introduce some method of knowledge representation ([7], [4]). The inference process developed for this method uses the paths of the stratified graphs, an order between the elementary arcs of a path and some results of universal algebras. The order is defined by considering a structured path instead of a regular path. In this paper we give two splitting properties. First property shows that every structured path can be uniquely decomposed by means of two structured subpaths. A similar decomposition is shown for the accepted structured paths. The decomposition of the accepted structured paths is used to define the inference process allowed by a stratified graph. This process is restated in the vision of the new results presented in this paper. This description is included in a separate section, where we define the concept of knowledge processing system based on stratified graphs. We give a formalism for the inference process in such systems.
Keywords: Peano algebra, labeled graph, stratified graph, structured path, accepted structured path, inference process

1 Introduction

The concept of stratified graph provides a method of knowledge representation. This concept was introduced in paper [7]. The resulting method uses concepts from graph theory redefined in the new framework and elements of universal algebra. Intuitively, a stratified graph is built over a labeled graph G_0, placing on top a subset of a Peano algebra generated by the label set of G_0.

The concept of structured path over a labeled graph was introduced in [4]. In the same paper was introduced the concept of accepted structured path over a stratified graph. The inference process was defined by means of a decomposition property of the accepted structured path, described in an intuitive manner in [4].

In this paper we define in a mathematical manner the concept of decomposition and obtain two splitting properties: one for a stratified graph and the other for an accepted structured path.

The inference process developed by a stratified graph is based on the decomposition of an accepted structured path into two accepted structured paths. These two components are subpaths of the initial path and the decomposition is iterated until we obtain atomic accepted paths. A subpath of a path defines a continuous path which consists of different kinds of elementary arcs of the initial path. Further we use the order induced by the structure of the accepted path and some meaning attached to every elementary arc.

The paper is organized as follows: Section 2 contains basic concepts as labeled graph and stratified graph. In section 3 we define the concept of structured path in a labeled graph, the concept of accepted structured path in a structured graph and we establish an useful result concerning the existence of some morphism of universal algebras obtained from the labels of the structured paths to Peano algebra generated by the elementary labels the structured graph (Proposition 4); Section 4 treats two decompositions of structured paths and accepted structured paths respectively; Section 5 defines the concepts of knowledge processing system based on stratified graphs and we give the formalism of the coresponding inference process. Last section includes conclusions of our study.

2 Basic concepts

We begin this section by a short presentation of two concepts: labeled graph and stratified graph. Various papers ([4], [3], [6]) present in their introduction these concepts. By *a labeled graph* we understand a tuple $G = (S, L_0, T_0, f_0)$, where S is a finite set of nodes, L_0 is a set of elements named *labels*, T_0 is a set of binary relations on S and $f_0 : L_0 \longrightarrow T_0$ is a surjective function. Such a structure admits a graphical representation. Each element of S is represented by a rectangle specifying the corresponding node. We draw an arc from $x_1 \in S$ to $x_2 \in S$ and this arc is labeled by $a \in L_0$ if $(x_1, x_2) \in f_0(a)$. This case is shown in Figure 1.

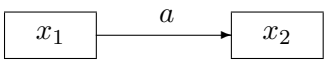


Figure 1: A labeled arc

We consider a symbol σ of arity 2 and take the sets defined recursively as follows:

$$\begin{cases} B_0 = L_0 \\ B_{n+1} = B_n \cup \{\sigma(x_1, x_2) \mid (x_1, x_2) \in B_n \times B_n\}, n \geq 0 \end{cases}$$

where L_0 is a finite set that does not contain the symbol σ. The set $\mathcal{B} = \bigcup_{n \geq 0} B_n$ is the Peano σ -algebra ([1]) generated by L_0. We can understand that $\sigma(x, y)$ is the word σxy over the alphabet $L_0 \cup \{\sigma\}$. Often this algebra is denoted by $\overline{L_0}$.

By $Initial(\overline{L_0})$ we denote a collection of subsets of B satisfying the following conditions: $M \in Initial(\overline{L_0})$ if

- $L_0 \subseteq M \subseteq B$
- if $\sigma(u, v) \in M$, $u \in \overline{L_0}$, $v \in \overline{L_0}$ then $u \in M$ and $v \in M$

We define the mapping $prod_S : dom(prod_S) \longrightarrow 2^{S \times S}$ as follows:

$$dom(prod_S) = \{(\rho_1, \rho_2) \in 2^{S \times S} \times 2^{S \times S} \mid \rho_1 \circ \rho_2 \neq \emptyset\}$$

$$prod_S(\rho_1, \rho_2) = \rho_1 \circ \rho_2$$

where $\circ$ is the usual operation between the binary relations:

$$\rho_1 \circ \rho_2 = \{(x, y) \in S \times S \mid \exists z \in S : (x, z) \in \rho_1, (z, y) \in \rho_2\}$$

We denote by $R(prod_S)$ the set of all the restrictions of the mapping $prod_S$:

$$R(prod_S) = \{u \mid u \prec prod_S\}$$

where $u \prec prod_S$ means that $dom(u) \subseteq prod_S$ and $u(\rho_1, \rho_2) = prod_S(\rho_1, \rho_2)$ for $(\rho_1, \rho_2) \in dom(u)$.

If u is an element of $R(prod_S)$ then we denote by $Cl_u(T_0)$ the *closure* of T_0 in the partial algebra $(2^{S\times S}, \{u\})$. This is the smallest subset Q of $2^{S\times S}$ such that $T_0 \subseteq Q$ and Q is closed under u. It is known that this is the union $\bigcup_{n\geq 0} X_n$, where

$$\begin{cases} X_0 = T_0 \\ X_{n+1} = X_n \cup \{u(\rho_1, \rho_2) \mid (\rho_1, \rho_2) \in dom(u) \cap (X_n \times X_n)\}, n \geq 0 \end{cases}$$

If $L \in Initial(L_0)$ then the pair $(L, \{\sigma_L\})$, where

- $dom(\sigma_L) = \{(x, y) \in L \times L \mid \sigma(x, y) \in L\}$
- $\sigma_L(x, y) = \sigma(x, y)$ for every $(x, y) \in dom(\sigma_L)$

is a partial algebra. This property is used to define the concept of stratified graph.

Consider a labeled graph $G_0 = (S, L_0, T_0, f_0)$. A *stratified graph* ([7]) $\mathcal{G}$ over G_0 is a tuple (G_0, L, T, u, f) where

- $L \in Initial(\overline{L_0})$
- $u \in R(prod_S)$ and $T = Cl_u(T_0)$
- $f : (L, \{\sigma_L\}) \longrightarrow (2^{S\times S}, \{u\})$ is a morphism of partial algebras such that $f_0 \prec f$, $f(L) = T$ and if $(f(x), f(y)) \in dom(u)$ then $(x, y) \in dom(\sigma_L)$

The existence of this structure, as well as the uniqueness is proved in [7]:

Proposition 1 *For every labeled graph $G_0 = (S < L_), T_0, f_0)$ and every $u \in R(prod_S)$ there is just one stratified graph (G_0, L, T, u, f) over G_0.*

3 Accepted structured paths

We consider a labeled graph $G_0 = (S, L_0, T_0, f_0)$. A *regular path* over G_0 is a pair $([x_1, \ldots, x_{n+1}], [a_1, \ldots, a_n])$ such that $(x_i, x_{i+1}) \in f_0(a_i)$ for every $i \in \{1, \ldots, n\}$.

Definition 1 *We denote by $STR(G_0)$ the smallest set satisfying the following conditions:*

- *For every $a \in L_0$ and $(x, y) \in f_0(a)$ we have $([x, y], a) \in STR(G_0)$.*
- *If $([x_1, \ldots, x_k], u) \in STR(G_0)$ and $([x_k, \ldots, x_n], v) \in STR(G_0)$ then $([x_1, \ldots, x_k, \ldots, x_n], [u, v]) \in STR(G_0)$.*

The concept of structured path introduces some order between the arcs taken into consideration for an regular path. To highlight the role of structured paths we consider the following example presented in Figure 2. We relieved here two structured paths: one of them is denoted by (1) and represents the structured path $([x_1, x_2, x_3, x_4], [[a_1, b_1], c_1])$; the other is denoted by (2) and represents the structured path $([x_1, x_2, x_3, x_4], [a_1, [b_1, c_1]])$. In order to explain in an intuitive manner the inference process we assign an algorithm to every arc symbol. For example, consider the following simple case: each arc symbol designates the following algorithm:

Alg

Input: x, y

Output: If $x \geq y$ then $x + y$; otherwise $x - y$

end

Each node of the labeled graph represents a natural number. In order to make a choice we take $x_1 = 7$, $x_2 = 2$, $x_3 = 5$ and $x_4 = 4$. For example the output of the algorithm for x_1 and x_2 is 9. We write $Alg(x_1, x_2) = 9$. For the paths (1) and (2) from Figure 2 we obtain

$$Alg(Alg(x_1, x_2), Alg(x_2, x_3)) = 14;\ Alg(Alg(Alg(x_1, x_2), Alg(x_2, x_3)), x_4) = 18$$

$$Alg(x_2, x_3) = -3;\ Alg(x_3, x_4) = 9;\ Alg(Alg(x_2, x_3), Alg(x_3, x_4)) = -12;$$

$$Alg(x_1, Alg(Alg(x_2, x_3), Alg(x_3, x_4))) = Alg(7, -12) = -5$$

Thus the inference process gives 18 for the first path and -5 for the second path.

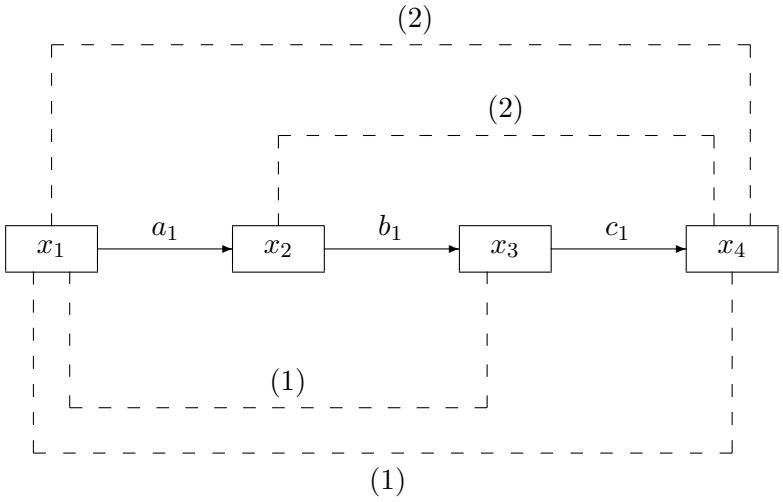


Figure 2: Intuitive representation of structured paths

Let us consider the set $\mathcal{L}(X) = \{[x_1, \ldots, x_n] \mid n \geq 1, x_i \in X, i = 1, \ldots, n\}$, the set of all nonempty lists over X. We denote $first([x_1, \ldots, x_n]) = x_1$ and $last([x_1, \ldots, x_n]) = x_n$.

We define the mapping

$$\otimes : STR(G_0) \times STR(G_0) \longrightarrow STR(G_0)$$

as follows:

- $dom(\otimes) = \{((\alpha_1, u_1), (\alpha_2, u_2)) \mid (\alpha_1, u_1) \in STR(G_0), (\alpha_2, u_2) \in STR(G_0),$
  $last(\alpha_1) = first(\alpha_2)\}$
- If $([x_1, \ldots, x_k], u) \in STR(G_0)$ and $([x_k, \ldots, x_n], v) \in STR(G_0)$ then

$$([x_1, \ldots, x_k], u) \otimes ([x_k, \ldots, x_n], v) = ([x_1, \ldots, x_n], [u, v])$$

Proposition 2 *Consider a labeled graph $G_0 = (S, L_0, T_0, f_0)$ and the set*

$$K(G_0) = \{([x, y], a) \mid (x, y) \in f_0(a)\} \tag{1}$$

The set $STR(G_0)$ is the $\otimes$-Peano algebra generated by $K(G_0)$.

Proof. From Definition 1 we deduce that $STR(G_0)$ is the smallest set containing $K(G_0)$ and closed under $\otimes$ operation. It follows that $STR(G_0)$ is the $\otimes$-Peano algebra generated by $K(G_0)$. ■

We define

$$STR_2(G_0) = \{w \mid \exists(\alpha, w) \in STR(G_0)\}$$

In fact, $STR_2(G_0)$ represents the projection of the set $STR(G_0)$ on the second axis: in a classical notation we write $STR_2(G_0) = pr_2(STR(G_0))$.

We define the mapping $* : STR_2(G_0) \times STR_2(G_0) \longrightarrow STR_2(G_0)$ as follows:

- $dom(*) = \{(\beta_1, \beta_2) \mid \exists\alpha_1, \alpha_2 : (\alpha_1, \beta_1) \in STR(G_0), (\alpha_2, \beta_2) \in STR(G_0),$
  $last(\alpha_1) = first(\alpha_2)\}$
- If $\beta_1, \beta_2 \in dom(*)$ then $\beta_1 * \beta_2 = [\beta_1, \beta_2]$

Remark 1 *The pair $STR_2(G_0), *)$ becomes a partial algebra.*

Proposition 3 *$STR_2(G_0)$ is the $*$-Peano algebra generated by L_0.*

Proof. The set $STR(G_0)$ is the $\otimes$-Peano algebra generated by $K(G_0)$. This means that $STR(G_0) = \bigcup_{n\geq 0} M_n$, where

$$\begin{cases} M_0 = K(G_0) \\ M_{n+1} = M_n \cup \{\gamma \mid \exists(\alpha, \beta) \in dom(\otimes) \cap (M_n \times M_n), \gamma = \alpha \otimes \beta\} \end{cases} \tag{2}$$

It follows that

$$STR_2(G_0) = pr_2 STR(G_0) = pr_2(\bigcup_{n\geq 0} M_n) = \bigcup_{n\geq 0} pr_2 M_n =$$

$$pr_2 M_0 \cup \bigcup_{n\geq 0} pr_2 M_{n+1} = pr_2 K(G_0) \cup \bigcup_{n\geq 0} pr_2 M_{n+1} =$$

therefore

$$STR_2(G_0) = L_0 \cup \bigcup_{n\geq 0} pr_2 M_{n+1} \tag{3}$$

Based on (2) we obtain

$$pr_2 M_{n+1} = pr_2 M_n \cup pr_2 X_n \tag{4}$$

where $X_n = \{\gamma \mid \exists(\alpha,\beta) \in dom(\otimes) \cap (M_n \times M_n), \gamma = \alpha \otimes \beta\}$.
From (4) we find that

$$pr_2M_{n+1} = pr_2M_n \cup \{pr_2\gamma \mid \exists(\alpha,\beta) \in dom(\otimes) \cap (M_n \times M_n), \gamma = \alpha \otimes \beta\} \tag{5}$$

Consider an element $\gamma \in X_n$. There are $(\alpha,\beta) \in dom(\otimes) \cap (M_n \times M_n)$ such that $\gamma = \alpha \otimes \beta$. This means that $\alpha = ([x_1,\ldots,x_k],u_1)$, $\beta = ([x_k,\ldots,x_m],v_1)$ and $\gamma = ([x_1,\ldots,x_k,\ldots,x_m],[u_1,v_1])$. It follows that $pr_2\gamma = [u_1,v_1]$ and by the definition of the operation $*$ we have $[u_1,v_1] = u_1 * v_1$. Thus, if $\gamma = \alpha \otimes \beta$, where $(\alpha,\beta) \in M_n \times M_n$ then $pr_2\gamma = pr_2\alpha * pr_2\beta$. This property allows to rewrite (6) as follows

$$pr_2M_{n+1} = pr_2M_n \cup \{w \mid \exists(\alpha,\beta) \in dom(\otimes) \cap (M_n \times M_n), w = pr_2\alpha * pr_2\beta\} \tag{6}$$

Let us denote $Y_n = pr_2M_n$ for every $n \geq 0$. We have $Y_0 = pr_2M_0 = pr_2K(G_0) = L_0$ and from (6) we obtain Let us prove that

$$\begin{aligned}\{w \mid \exists(\alpha,\beta) \in dom(\otimes) \cap (M_n \times M_n), w = pr_2\alpha * pr_2\beta\} \quad = \\ \{\omega \mid \exists(u,v) \in (Y_n \times Y_n) \cap dom(*) : \omega = u * v\}\end{aligned} \tag{7}$$

Take $w = pr_2\alpha * pr_2\beta$ for some $(\alpha,\beta) \in dom(\otimes) \cap (M_n \times M_n)$. It follows that $\alpha = ([x_1,\ldots,x_k],pr_2\alpha)$, $\beta = ([y_1,\ldots,y_r],pr_2\beta$ and $x_k = y_1$. Denote $pr_2\alpha = u$ and $pr_2\beta = v$. Because $\alpha = ([x_1,\ldots,x_k],pr_2\alpha) \in M_n$ we obtain $u = pr_2\alpha \in pr_2M_n$. Similarly we have $v \in pr_2M_n$. But $pr_2M_n = Y_n$, therefore $u \in Y_n$ and $v \in Y_n$. We have $w = u * v$ therefore we proved the inclusion

$$\begin{aligned}\{w \mid \exists(\alpha,\beta) \in dom(\otimes) \cap (M_n \times M_n), w = pr_2\alpha * pr_2\beta\} \subseteq \\ \{\omega \mid \exists(u,v) \in (Y_n \times Y_n) \cap dom(*) : \omega = u * v\}\end{aligned} \tag{8}$$

We prove now the converse inclusion. To prove this property we consider an element $\omega = u * v$ for some $(u,v) \in (Y_n \times Y_n) \cap dom(*)$. But $Y_n = pr_2M_n$ and $u \in Y_n$. It follows that there is $\alpha = ([x_1,\ldots,x_k],u) \in M_n$ and $\beta \in ([y_1,\ldots,y_m],v) \in M_n$ $x_k = y_1$. We deduce that $(\alpha,\beta)dom(\otimes) \cap (M_n \times M_n)$ such that $\omega = pr_2\alpha * pr_2\beta$. This shows that

$$\begin{aligned}\{w \mid \exists(\alpha,\beta) \in dom(\otimes) \cap (M_n \times M_n), w = pr_2\alpha * pr_2\beta\} \supseteq \\ \{\omega \mid \exists(u,v) \in (Y_n \times Y_n) \cap dom(*) : \omega = u * v\}\end{aligned} \tag{9}$$

Now, from (8) and (9) we obtain (7).
From (6) and (7) we obtain

$$pr_2M_{n+1} = pr_2M_n \cup \{\omega \mid \exists(u,v) \in (Y_n \times Y_n) \cap dom(*) : \omega = u * v\}$$

equivalently we can write that

$$Y_{n+1} = Y_n \cup \{\omega \mid \exists(u,v) \in (Y_n \times Y_n) \cap dom(*) : \omega = u * v\} \tag{10}$$

From $Y_0 = L_0$ and (10) we obtain that $\bigcup_{n\geq 0} Y_n = \overline{L_0}$, where $\overline{L_0}$ is taken under operation $*$. From (3) we obtain $STR_2(G_0) = \bigcup_{n\geq 0} Y_n$, therefore $STR_2(G_0) = (\overline{L_0})_*$ and the proposition is proved. ∎

Proposition 4 *The mapping* $h : (STR_2(G_0),*) \longrightarrow ((\overline{L_0})_\sigma,\sigma)$ *defined by*

$$h(p) = \begin{cases} p \text{ if } p \in L_0 \\ \\ \sigma(h(u),h(v)) \text{ if } p = [u,v], u \in STR_2(G_0), v \in STR_2(G_0) \end{cases}$$

is a morphism of partial algebras. In other words, the diagram from Figure 3 is commutative.

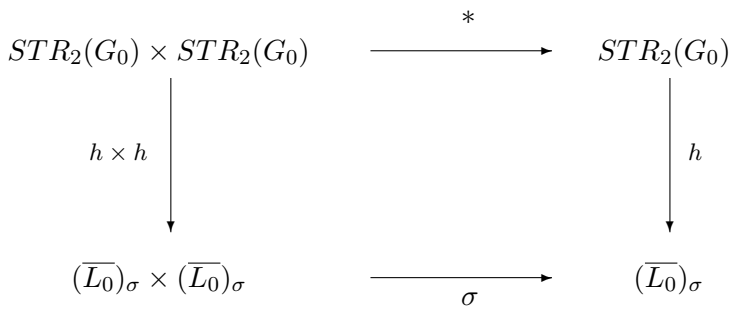


Figure 3: Commutative diagram

Proof. Consider $(u, v) \in dom(*)$. There are $([x_1, \ldots, x_k], u) \in STR(G_0)$ and $([x_k, \ldots, x_n], v) \in STR(G_0)$. If this is the case then $u * v = [u, v] \in STR_2(G_0)$ and $h([u, v]) = \sigma(h(u), h(v))$. Thus the diagram is commutative. ∎

Definition 2 *We define the set $ASP(\mathcal{G})$ as follows: $([x_1, \ldots, x_{n+1}], c) \in ASP(\mathcal{G})$ if and only if $([x_1, \ldots, x_{n+1}], c) \in STR(G_0)$ and $h(c) \in L$.*
*An element of $ASP(\mathcal{G})$ is named **accepted structured path** over $\mathcal{G}$.*

4 Splitting properties

In this section we obtain two splitting properties: one of them refers to the decomposition of a structured path; the other gives the decomposition of an accepted structured path. The first splitting property is used to prove the second property.

Proposition 5 *(splitting property I)*
If $([x_1, \ldots, x_{n+1}], c) \in STR(G_0)$ and $n \geq 2$ then there are $u, v \in STR_2(G_0)$ and $k \in \{2, \ldots, n\}$, uniquely determined, such that
$c = [u, v]$
$([x_1, \ldots, x_k], u) \in STR(G_0)$
$([x_k, \ldots, x_{n+1}], v) \in STR(G_0)$

Proof. We denote by $(\overline{L_0})_*$ the *-Peano algebra generated by L_0. By Proposition 3 we have $STR_2(G_0) = (\overline{L_0})_*$. In a similar manner we consider the $\otimes$-Peano algebra generated bu $K(G_0)$, denoted by $(\overline{K(G_0)})_\otimes$. By Proposition 2 we have $STR(G_0) = (\overline{K(G_0)})_\otimes$.
Take $([x_1, \ldots, x_{n+1}], c) \in STR(G_0)$, $n \geq 2$. This implies that $c \in STR_2(G_0) = (\overline{L_0})_*$, therefore there are $u, v \in STR_2(G_0)$, uniquely determined, such that $c = [u, v]$. Thus $([x_1, \ldots, x_{n+1}], [u, v]) \in STR(G_0) = (\overline{K(G_0)})_\otimes$. It follows that there are the elements, uniquely determined, $d_1 = ([y_1, \ldots, y_s], \gamma_1) \in STR(G_0)$, $d_2 = ([z_1, \ldots, z_p], \gamma_2) \in STR(G_0)$ such that $(d_1, d_2) \in dom(\otimes)$ and

$$([x_1, \ldots, x_{n+1}], [u, v]) = d_1 \otimes d_2 \tag{11}$$

From $(d_1, d_2) \in dom(\otimes)$ we deduce that $y_s = z_1$ and

$$d_1 \otimes d_2 = ([y_1, \ldots, y_s, z_2, \ldots, z_p], [\gamma_1, \gamma_2]) \tag{12}$$

From (11) and (12) we deduce that

$$[x_1, \ldots, x_{n+1}] = [y_1, \ldots, y_s, z_2, \ldots, z_p] \tag{13}$$

$$[u, v] = [\gamma_1, \gamma_2]$$

We have $u, v, \gamma_1, \gamma_2 \in STR_2(G_0))$, $STR_2(G_0)$ is a $*$ -Peano algebra and from $[u, v] = [\gamma_1, \gamma_2]$ we deduce $u = \gamma_1$ and $v = \gamma_2$. From (13) we deduce that $n+1 = s+p-1$ and $x_1 = y_1$, ..., $x_s = y_s$, $x_{s+1} = z_2$, ..., $x_{n+1} = z_p$. It follows that $d_1 = ([x_1, \ldots, x_s], u)$ and $d_2 = ([x_s, \ldots, x_{n+1}], v)$. But $d_1 \in STR(G_0)$ and $d_2 \in STR(G_0)$. We remark that s is uniquely determined. Thus the proposition is proved. ∎

Proposition 6 *(splitting property II)*
If $([x_1, \ldots, x_{n+1}], c) \in ASP(\mathcal{G})$ *and* $n \geq 2$ *then there are* $u, v \in STR_2(G_0)$ *and* $k \in \{2, \ldots, n\}$, *uniquely determined, such that*

$c = [u, v]$
$([x_1, \ldots, x_k], u) \in ASP(\mathcal{G})$
$([x_k, \ldots, x_{n+1}], v) \in ASP(\mathcal{G})$

Proof. Consider $([x_1, \ldots, x_{n+1}], c) \in ASP(\mathcal{G})$ and $n \geq 2$. Because $ASP(\mathcal{G}) \subseteq STR(G_0)$ we can apply Proposition 5. Thus, there are $u, v \in STR_2(G_0)$ and $k \in \{2, \ldots, n\}$, uniquely determined, such that

$c = [u, v]$
$([x_1, \ldots, x_k], u) \in STR(G_0)$
$([x_k, \ldots, x_{n+1}], v) \in STR(G_0)$

But $h(c) \in L$, therefore from the definition of the mapping h we deduce that $\sigma(h(u), h(v)) \in L$. We have $h(u) \in (\overline{L_0})_\sigma$ and $h(v) \in (\overline{L_0})_\sigma$. From $L \in Initial((\overline{L_0})_\sigma)$ we deduce that $h(u) \in L$ and $h(v) \in L$. This shows that $([x_1, \ldots, x_k], u) \in ASP(\mathcal{G})$ and $([x_k, \ldots, x_{n+1}], v) \in STR(G_0)$. ∎

5 Inference process based on accepted structured paths

We consider a stratified graph $\mathcal{G} = (G_0, L, T, u, f)$ over $G_0 = (S, L_0, T_0, f_0)$. Let $\mathcal{Y} = (Y, \odot)$ be a binary algebra and an injective mapping $ob : S \longrightarrow Y$. We suppose that for each $u \in L$ we have an algorithm $Alg_u : Y \times Y \longrightarrow Y$. This means that is a partial mapping such that $dom(Alg_u) \subseteq Y \times Y$ and for every pair $(x, y) \in dom(Alg_u)$ given as input for $Alg_u)$ this algorithm gives as output some element of Y.

Definition 3 *A* ***knowledge processing system*** *based on stratified graphs is a tuple*

$$KPS = (\mathcal{G}, (Y, \odot), ob, \{Alg_u\}_{u \in L})$$

where

- $G = (G_0, L, T, u, f)$ *is a stratified graph over* $G_0 = (S, L_0, T_0, f_0)$*;*
- $(Y, \odot)$ *is a binary partial algebra;*
- $ob : S \longrightarrow Y$ *is an injective mapping;*
- *For each* $u \in L$ *the entity* Alg_u *is an algorithm that defines a mapping* $Alg_u : dom(ALg_u) \longrightarrow Y$, *where* $dom(Alg_u) \subseteq Y \times Y$.

We agree to say that such a structure is a **knowledge processing system over $\mathcal{G}$ with Y as output space** an we denote this property by $KPS(\mathcal{G}, Y)$.

For each $d = ([x_1, \ldots, x_{n+1}], \sigma(u, v)) \in ASP(\mathcal{G})$ we consider the image $d_{ob} = ([ob(x_1), \ldots, ob(x_{n+1})], \sigma(u, v))$ of the path d. We denote

$$ASP_{ob}(\mathcal{G}) = \{d_{ob} \mid d \in ASP(\mathcal{G})\}$$

We remark that we can consider the operation $\otimes$ for the case of images of accepted paths, as in the case of structured paths:

$$\otimes : ASP_{ob} \times ASP_{ob} \longrightarrow ASP_{ob}$$

as follows:

- $dom(\otimes) = \{((\alpha_1, u_1), (\alpha_2, u_2)) \mid (\alpha_1, u_1) \in ASP_{ob}, (\alpha_2, u_2) \in ASP_{ob}, \\ last(\alpha_1) = first(\alpha_2)\}$
- If $([x_1, \ldots, x_k], u) \in ASP_{ob}$ and $([x_k, \ldots, x_n], v) \in ASP_{ob}$ then

$$([x_1, \ldots, x_k], u) \otimes ([x_k, \ldots, x_n], v) = ([x_1, \ldots, x_n], [u, v])$$

For a knowledge processing system based on stratified graphs we can define the inference process as in the next definition.

Definition 4 *The **inference process** $IP_{\mathcal{G},Y}$ generated by the stratified graph $\mathcal{G}$ and the output space Y is the mapping*

$$IP_{\mathcal{G},Y} : ASP_{ob}(\mathcal{G}) \longrightarrow Y$$

defined as follows:

$$IP_{\mathcal{G},Y}(d_{ob}) = \begin{cases} Alg_a(x, y) \text{ if } ([x, y], a) \in ASP_{ob} \\ \\ IP_{\mathcal{G},Y}(d_1) \odot IP_{\mathcal{G},Y}(d_2) \text{ if } d_{ob} = d_1 \otimes d_2 \end{cases}$$

Based on previous concepts and results we can propose the following algorithm of the inference process.

Input:

$KPS = (\mathcal{G}, (Y, \odot), ob, \{Alg_u\}_{u \in L})$; $((x, y) \in ob(S) \times ob(S)$

Method:

Compute $C = \{d_{ob} = (X, u) \mid first(X) = x, last(X) = y\}$;

Output:

$IP_{\mathcal{G},Y}(C)$

End

6 Conclusions

In this paper we treat from the mathematical point of view the concept of inference based on stratified graphs. We define the concept of knowledge processing system with stratified graphs and the concept of inference of such systems.

References

[1] V. Boicescu, A. Filipoiu, G. Georgescu and S. Rudeanu, Ľukasiewicz-Moisil Algebra, *Annals of Discrete Mathematics* **49** (North-Holland, 1991)

[2] **N. Ţăndăreanu**, Collaborations between distinguished representatives for labelled stratified graphs, *Annals of the University of Craiova, Mathematics and Computer Science Series*, **30** (2003), no.2, 184-192.

[3] **N. Ţăndăreanu**, Distinguished Representatives for Equivalent Labelled Stratified Graphs and Applications, *Discrete Applied Mathematics*, **144** (2004), no.1-2, 183-208.

[4] **N. Ţăndăreanu**, Knowledge representation by labeled stratified graphs, *Proceedings of the 8th World Multi-Conference on Systemics, Cybernetics and Informatics* (2004), Vol. 5, 345-350.

[5] **N. Ţăndăreanu**, Master-Slave Systems of Semantic Schemas and Applications, *The 10th IASTED International Conference on Intelligent Systems and Control* (ISC 2007), November 19-21, (2007), Cambridge, Massachusetts, 150-155.

[6] **N. Ţăndăreanu**, M. Ghindeanu, Hierarchical Reasoning Based on Stratified Graphs. Application in Image Synthesis, *Proceedings of The 15th International Workshop on Database and Expert Systems Applications*, (DEXA 2004), Zaragoza, Spania, (2004) IEEE Computer Society, Los Alamitos California, 498-502

[7] **N. Ţăndăreanu**, Proving the Existence of Labelled Stratified Graphs, *Annals of the University of Craiova*, Vol. XXVII (2000), 81-92

[8] **Nicolae Ţăndăreanu, Cristina Zamfir**, Slices and extensions of ω-trees, Annals of the University of Craiova, Mathematics and Computer Science Series, **38** (2011), no.1, 72-82.

Dănciulescu Daniela
University of Craiova
Department of Informatics,
Al.I. Cuza Street, No. 13, Craiova RO-200585
ROMANIA
E-mail: *ntand@rdslink.ro*

Third International Conference
Modelling and Development of Intelligent Systems
October 10 - 12, 2013
"Lucian Blaga" University
Sibiu - Romania

Computational intelligence in medical data sets

Ionela MANIU, George MANIU, Daniel HUNYADI

Abstract

In recent years, collection of data, regardless of the field, has become a normal phenomenon. In the activity and evolution of an organization is imperative to take into account the data collected in order to achieve decision process. As the volume and complexity of data are in constant growth is necessary to use intelligent methods and fundamental tools for storing, processing, filtering and obtaining information from these data.

1. Introduction

Extraction of information/learning from data/knowledge discovery from data is the primary goal of intelligent computational methods [4]. Learning from data can be done supervised or unsupervised. The objective of supervised learning is to predict the amount of output data based on the input data, and in unsupervised learning the objective is to describe associations and characteristics/structures of the input data.

2 Strategies to achieve the knowledge discovery

2.1 Descriptive and exploratory phase

First step in knowledge discovery consists in data exploration [1][5]. This first phase is descriptive and exploratory and analyse elements such as distribution, identification of atypical values, data transformations required by the distribution form or data standardization, means, cluster variance, correlation, classification, etc. This approach has as result an achieving of data descriptions that establish relationships among variables providing a first general idea of the data.

In this phase can be considered two main objectives. The first objective is to explore one-dimensional and multidimensional or reduce the data dimension and the methods and the instruments used are: factor analysis, principal component analysis, analysis of simple correspondences. These methods consist in analysing a weighted point cloud into a space with a special metric, the cloud forms characterizing the nature and intensity of the relationships between variables and revealing information contained in data structures. The second objective is the classification or segmentation and

it can be achieved by the following methods: hierarchical ascending classification (cluster progressive elements), the k-means (iterative aggregation elements around mobile canters) or mixed methods[3]. In this case we want the division and distribution into classes or categories by optimizing a criterion, each class having property that is as homogeneous in its entirety and report more distinctive compared to other classes.

The k-means clustering algorithm [6] is a straightforward and effective algorithm for finding clusters in data. The algorithm proceeds as follows.

- Step 1: Ask the user how many clusters k the data set should be partitioned into.
- Step 2: Randomly assign *k* records to be the initial cluster center locations.
- Step 3: For each record, find the nearest cluster center. Thus, in a sense, each cluster center "owns" a subset of the records, thereby representing a partition of the data set. We therefore have *k* clusters, C_1, C_2, . . . , C_k.
- Step 4: For each of the *k* clusters, find the cluster centroid, and update the location of each cluster center to the new value of the centroid.
- Step 5: Repeat steps 3 to 5 until convergence or termination.

The "nearest" criterion in step 3 is centroid distance (distance between the centroids of each cluster), although other criteria may be applied as well.

Technically speaking, the algorithm steps are:

- Assume the existence of N vectors $x^l = (x_1, x_2, .., x_n)$;
- Identify a representative set of *k* vectors c_j, where $j = 1, 2, ..., k$;
- Partition data in *k* disjoint subsets *Sj* containing N_j points, so to minimize the clustering function given by:

$$J = \sum_{j=1}^{k} \sum_{l \in S_j} \| x^l - c_j \|^2 \qquad (1)$$

where c_j is the average centroid data from the set *Sj*, given by:

$$c_j = \frac{\sum_{l \in S_j} x^l}{N_j} \qquad (2)$$

One attractive classification method involves the construction of a decision tree, a collection of decision nodes, connected by branches, extending downward from the root node until terminating in leaf nodes. Beginning at the root node, which by convention is placed at the top of the decision tree diagram, attributes are tested at the decision nodes, with each possible outcome resulting in a branch. Each branch then leads either to another decision node or to a terminating leaf node.

Classification and decision trees are used to forecast membership of objects / instances in different categories, based on their measures in relation to one or more predictor variables. Classification tree analysis is a major data mining techniques. The flexibility of this technique makes it especially attractive, particularly because the benefit of present and suggestive views (tree which summarizes the classification obtained).

Conceptually, the construction algorithm and decision tree classification is as follows:

- Let D_t training set which is at node *t*;
- If D_t is the empty set, then *t* is a leaf labeled default C_Φ;
- If D_t contains instances belonging to the same class C_t, where *t* is a leaf labeled C_t;
- If D_t contains several instances belonging to one class, then use an attribute node test to divide D_t in smaller subsets. The procedure is applied recursively for each node.

The strategy underlying the optimal partitioning of a node type is a greedy method, a recursive construction "top down" *divide et impera* type.

In principle, the methodology for classification and decision tree induction consists of two phases:

- Construction of the original tree, using the available training set until each leaf is "pure" or almost "pure".
- "Forming" tree as "increased" to improve the accuracy obtained by the test set.

Briefly, the algorithm behind the building and decision tree classification is as follows:

```
Build tree (training data T)
{
        Partition (T)
}
Partition (S data)
{
        if (all points of S are in the same class) then
                returns
        for each attribute A do
                evaluates the split on attribute A;
        using the best split found for partitioning S in S1 and S2
        Partition(S1)
        Partition(S2)
}
```

2.2 Inferential and confirmatory phase

The first phase is preceded by a second inferential step. This step uses the results obtained in the first stage as assumptions in statistical tests or probabilistic models that explain that to predict a certain variable with one or more explanatory variables.

The main objective at this phase is modelling and deduction of a predictive model. To achieve this you can use methods such as linear regression, ANOVA, ANCOVA, neural networks, classification and regression trees, SVM, models based on statistical observations series: spectral analysis, seasonality analysis.

3 Case study

In the database used in this paper are analysed indicators such as blood sugar, cholesterol, systolic and diastolic blood pressure, indicators that we see that are closely correlated with body mass index (IMC) based indicator which are established cases of overweight and obesity. The correlation between IMC and age was also studied [2]. A first set of data from the database is represented by the values of these indicators observed in the case of a factory employee (N=433), the second set are the values seen from a hospital employee (N=300) and a third for employees in the administrative field (N=70).

During the first phase of knowledge discovery, an descriptive and exploratory phase, the objective is to analyze the shape of distribution indicators above, identify outliers or data entry errors, missing values identification, variables transformation.

Cases studied distribution shapes and clinical characteristics are represented in figure 1 and table 1. As can been seen from the graph and histogram respectively after applying the test “Kolmogorov-Smirnov”, in the case of 6 variables analyzed distribution differs significantly from a normal distribution ($p < 0.05$). In this situation, if you want to compare values of variables is indicated using operations transformation of values or using nonparametric tests. In the analysis were compared and values of these variables on lots has been rechecked their distribution form for each lot. After the checks were encountered situations of normality and abnormality of the distribution.

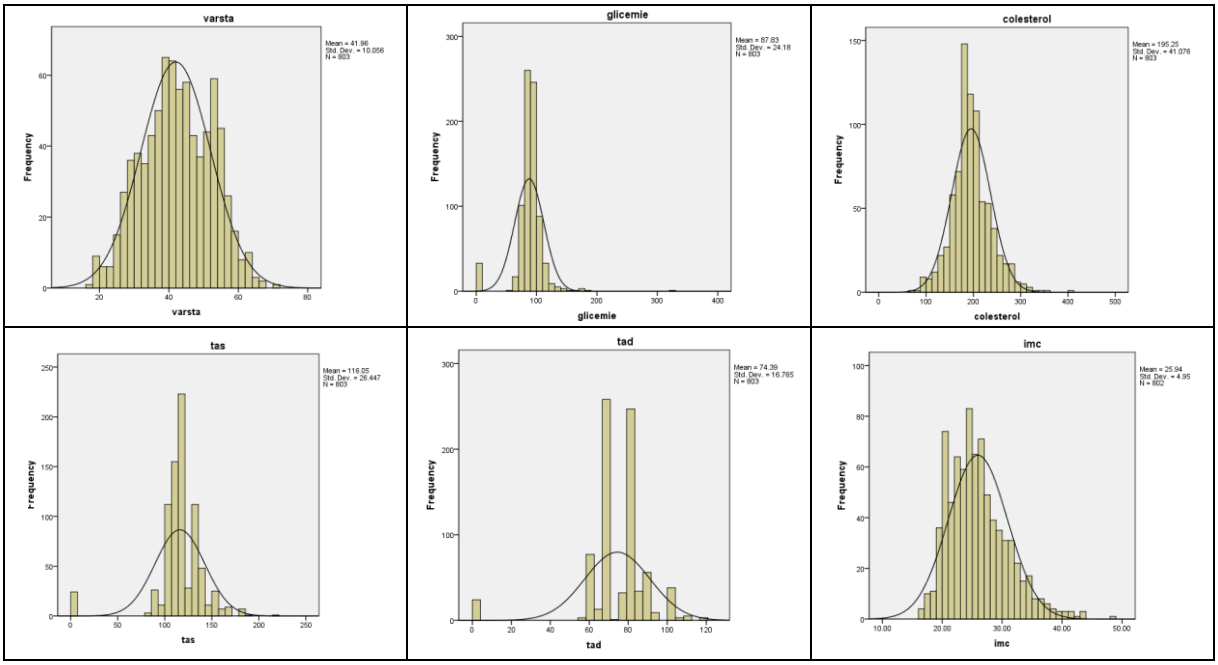


Figure 1 - Cases studied distribution shapes

Characteristics	Mean ±SD	Range	Skewness / Kurtosis
Age (varsta)	41.96 ± 10.05	17 -70	-0.04 / -0.58
Glucose (glicemie)	87.83 ± 16.25	58 - 321	4.82 / 5.25
Cholesterol (colesterol)	195.25 ± 41.07	68 - 409	0.49 / 1.58
TAS	119.62 ± 17.11	80 - 220	1.07 / 2.69
TAD	76.68 ± 10.66	55 - 120	0.75 / 1.01
IMC	25.93 ± 4.95	16.32 – 48.27	0.87 / 0.94

Table 1. Cases studied clinical characteristics

Obesity groups are defined by: normal IMC<25, overweight (suprapondere) 25<=IMC<30, obese (obezitate) IMC >=30. Data are expressed as Mean ± SD. Comparation of normal vs. overweight or normal vs. obese is done by either the parametric "One way ANOVA" with "Post Hoc Tests (Bonferroni)" or the nonparametric Kruskal-Wallis test. The significance is indicated by (♦) p<0.05, (♦♦) p<0.01, (♦♦♦) p<0.001 for normal vs. overweight and by (*) p<0.05, (**) p<0.01, (***) p<0.001 for normal vs. obese.

Clinical characteristics of the different adiposity groups defined by IMC are represented in the table 2.

Characteristics	Normal N=387	Overweight N=259	Obese N=156
Age (varsta)	37.96 ± 9.42	44.85 ± 9.06 ♦♦♦	47.10 ± 9.20 ***
Glucose (glicemie)	88.55 ± 11.78	93.06 ± 19.95 ♦♦	96.80 ± 11.66 ***
Cholesterol (colesterol)	186.20 ± 35.84	200.99 ± 41.74 ♦♦♦	208.27 ± 46.89 ***
TAS	113.11 ± 13.47	123.75 ± 17.09 ♦♦♦	129.19 ± 18.77 ***
TAD	72.21 ± 8.31	79.09 ± 10.02 ♦♦♦	83.96 ± 11.73 ***

Table 2. Clinical characteristics of the different adiposity groups

As can be seen from the table above, glucose, cholesterol, SBP, DBP were significantly higher in overweight or obese persons compared to persons with normal BMI. This trend was the same in case

of group of people who work in the factory and the hospital, while in case of the people from administrative group is found glucose values, cholesterol, SBP, DBP slightly higher in overweight to obese.

Correlation between IMC and other clinical characteristics is then analysed and the results are represented by scatterplots. Data are expressed as "Pearson correlation coefficient" and the significance is the same as upstairs. Results are presented in table 3 and figure 2.

Pearson correlation	Normal N=387	Overweight N=259	Obese N=156
Age (varsta) vs. Glucose (glicemie)	0.169 **	0.178**	0.152
Age (varsta) vs. Cholesterol (colesterol)	0.124 *	0.090	0.069
Age (varsta) vs. TAS	0.332 **	0.235**	0.233**
Age (varsta) vs. TAD	0.266 **	0.216**	0.155
Age (varsta) vs. IMC	0.204 **	-0.035	-0.062
Glucose (glicemie) vs. Cholesterol (colesterol)	0.041	0.130*	0.128
Glucose (glicemie) vs. TAS	0.090	0.243**	0.190*
Glucose (glicemie) vs. TAD	0.094	0.160*	0.160
Glucose (glicemie) vs. IMC	0.070	0.023	-0.081
Cholesterol (colesterol)vs. TAS	0.077	0.137*	0.247**
Cholesterol (colesterol)vs. TAD	0.136 **	0.188**	0.166**
Cholesterol (colesterol)vs. IMC	0.095	0.061	-0.036
TAS vs. TAD	0.731 **	0.753**	0.796**
TAS vs. IMC	0.187**	0.230**	0.155*
TAD vs. IMC	0.192**	0.185**	0.247**

Table 3. Correlation coefficient between IMC and other clinical characteristics

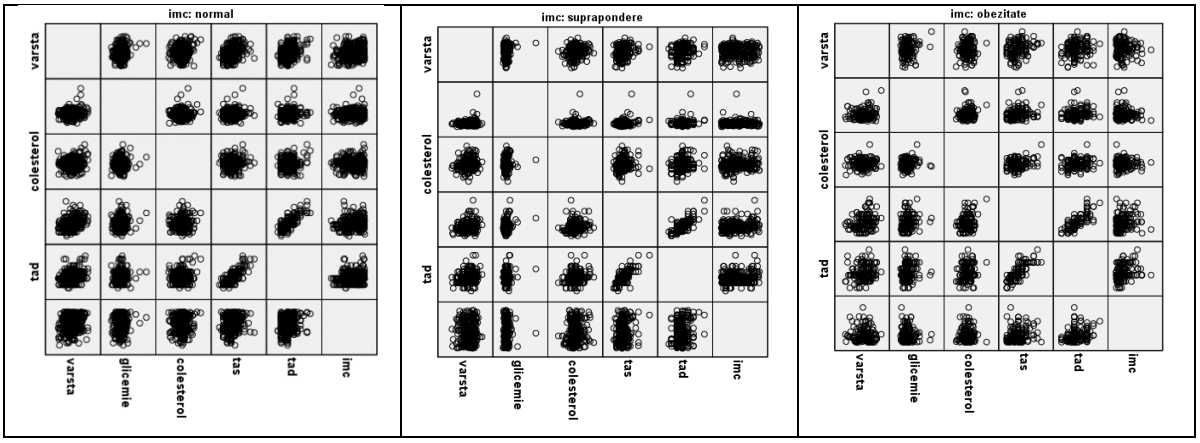


Figure 2. Correlation scatterplot

If normal BMI, it is significantly positively associated with age while if overweight and obesity have a negative association, but insignificant.

There was no significant correlation between BMI and blood sugar, cholesterol, except that in both cases the association was positive at people with normal BMI and overweight and the obese identified a negative association. A positive association was observed between BMI and blood.

Dendrogram obtained by cluster analysis, using the *Between-groups linkage* and the metrics *Squared Euclidian distance* shows that the variables are clustered in two groups, in fist group age (varsta) is clustered with IMC, and in the second one glucose (glicemie) is clustered with TAS, TAD, and cholesterol is not a part of any of the clusters (figure 3).

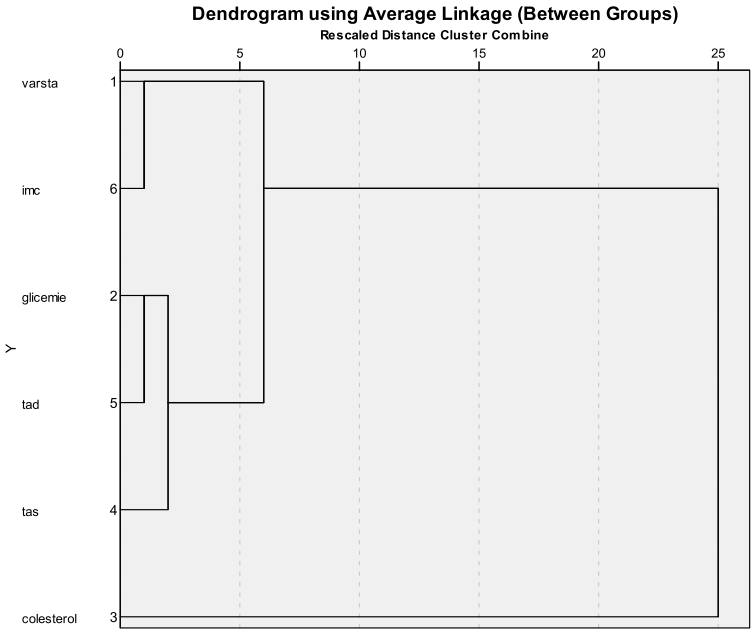


Figure 3. Dendrogram obtained by cluster analysis

4 Conclusions

Exploratory and explanatory methods are basic tools for data exploration. There is no best method, but experience in the choice of the method has an important role that is adapted to the types of database variables and having very good experience clarified the objectives of the study.

5 References

[1] Vapnik, V., The nature of statistical learning theory, Springer-Verlag, 1995
[2] Bhargava, S.K., Sachdev, H.S., Fall, C., Osmond, C., Lakshmy, R., Barker, D.J.P., Biswas, S.K.D., Ramji, S., Prabhakaran, D. & Reddy, K.S., Relation of serial changes in childhood body-mass index to impaired glucose tolerance in young adulthood. New England Journal of Medicine, 2004
[3] Han, J. & Kamber, M., Data Mining: Concepts and Techniques (2nd ed.). Ed. Morgan Kaufmann, 2006
[4] Baccini, A., Besse, P., Data mining / Exploration Statistique. Toulouse INSA, 2010
[5] Lepadatu, C., Explorarea datelor si descoperirea cunostiintelor – probleme, obiective si strategii, RRIA, vol. 4, nr. 4, 2012
[6] Daniel T. Larose (2005), Discovering Knowledge in data. An Introduction to Data Mining, John Wiley & Sons, Inc

Ionela MANIU
"Lucian Blaga" University of Sibiu
Faculty of Sciences
Sibiu, Dr. Ioan Rațiu St. No. 5 - 7
ROMÂNIA
E-mail: ionela.maniu@yahoo.ro

George MANIU
"Lucian Blaga" University of Sibiu
Faculty of Sciences
Sibiu, Dr. Ioan Rațiu St. No. 5 - 7
ROMÂNIA
E-mail: costelmaniu@yahoo.com

Daniel HUNYADI
"Lucian Blaga" University of Sibiu
Faculty of Sciences
Sibiu, Dr. Ioan Rațiu St. No. 5 - 7
ROMÂNIA
E-mail: daniel.hunyadi@ulbsibiu.ro

Third International Conference
Modelling and Development of Intelligent Systems
October 10 - 12, 2013
"Lucian Blaga" University
Sibiu - Romania

A Second Order-Cone Programming Formulation for Simple Assembly Line Balancing Problem

Vasile Moraru, Sergiu Zaporojan

Abstract

Decision support in order to assure an optimal business in the framework of an industrial company is based on some mathematical models, including optimization. The paper discusses the numerical solving of a task known as the Simple Assembly Line Balancing Problem (SALBP-I). In this context, a model based on the Second-Order Cone Programming (SOCP) it is proposed.

1 Introduction

The success at the level of an enterprise requires optimal organization of the production processes and related activities. These activities include organizational processes, economic and financial, production, trade, and, not least, information processes. What it is important in the organization of the business process it is the optimization of the enterprise activity, taking into account the market demands and current technology power. The activities listed are made by human agents and/or machines to help achieve business objectives across the enterprise. So, business process optimization is related to activities (or tasks), participants (human agents and/or machines) and targets (performance indicators).

In general, the requirements for optimal organization of the enterprise activity are actual and fit into the idea of reengineering. This idea is centered on all processes in the modern enterprise. Reengineering is a radical redesign of a business process to achieve a considerable improvement in performance indicators (cost, quality, productivity, etc.). The idea of reengineering is actual one as information technology is constantly changing. Hence the ability to adapt quickly to market demands.

Decision support for an optimal business in the framework of an industrial enterprise is not always possible without software products that are based on mathematical models of combinatorial optimization. It is the case of the simple assembly (manufacturing) line balancing problem (SALBP).

The assembly line consists of a finite number of workstations that are running individual operations (tasks) to manufacture a single product. The problem now is to combine operations and workstations in order to obtain an optimal distribution of the workload to a minimum number of stations. At the same time, it is necessary to ensure conditions of precedence for the execution of operations.

Assume the following conditions ([1]):

- the assembly line is designed for a single product and supports only one mode of functioning;
- the stations are serial arranged;

- the execution time for operations is deterministic;
- the partition of operations is prohibited;
- all operations must be performed;
- there are precedence constraints;
- the execution time of an operation does not depend on the station on which is running;
- the cycle time is fixed.

In the literature, several models of the SALBP-I problem have been presented ([2]). Most of them are formulated in terms of linear programming with binary variables and fit into the class of NP-hard problems ([3]). In this paper, we propose modeling the problem as a second-order cone programming (SOCP).

2 Mathematical programming model for SALBP-1

It is assumed an assembly line of *n* stations. At each station runs one operation by a person or an automatic device (robot), which is needed to manufacture a product. For product assembly all operations must be done in a strict order.

We denote by *P* the immediate precedence matrix of dimension $n \times n$:

$$P(i,j) = \begin{cases} 1, \textit{if task } j \textit{ is an immediate successor of task } i, \\ 0, \textit{otherwise.} \end{cases}$$

It is consider to be known the production rate *R* (the number of items collected per unit time) and the processing time t_i of the operation i. The number of stations can not be lower than T/C, where $T = \sum_{i=1}^{n} t_i$ is the time of all needed operations, and $C = 1/R$ is the total cycle time of the assembly line.

It is required to perform the distribution of operations to workstations so that the number of jobs to be minimal, i.e., the maximizing of the number of "empty" jobs. An "empty" is a plant that does not perform any operation.

We introduce Boolean variables x_{ij} and y_i:

$$x_{ij} = \begin{cases} 1, \textit{if task i is assigned to workstation } j, \\ 0, \textit{otherwise.} \end{cases}$$

$$y_i = \begin{cases} 1, \textit{if workstation i has a task assigned to it}, \\ 0, \textit{otherwise.} \end{cases}$$

such that, $x_{ij} = 1$ if the operation i is carried out at the workstation j and $x_{ij} = 0$ when it is carried out at another station; $y_i = 1$ if one of the operations is performed at the station i, and $y_i = 0$ in the opposite case. For the *SALBP-I* problem different formulations have been proposed. We will consider the following mathematical model ([1]):

$$\sum_{i=1}^{n} y_i \to \min \tag{1}$$

subject to

$$\sum_{j=1}^{n} x_{ij} = 1, \forall i = 1,2,\text{K}\ ,n\ , \tag{2}$$

$$\sum_{i=1}^{n} t_i x_{ij} \le C, \forall j = 1,2,\text{K}\ ,n\ , \tag{3}$$

$$x_{ik} \le \sum_{j=1}^{k} x_{sj}, \forall k = 1,2,\text{K}\ ,n, \forall i,s : P(i,s) = 1\ , \tag{4}$$

$$\sum_{i=1}^{n} x_{ik} \le n(1 - y_k), \forall k = 1,2,\text{K}\ ,n\ , \tag{5}$$

$$x_{ij} \in \{0,1\}, y_i \in \{0,1\}, \forall i,j = 1,2,\text{K}\ ,n\ . \tag{6}$$

Restrictions (2) ensure that each operation will be performed only at a single workstation, and (3) - that the cycle must be greater than or equal to the length of time at all stations. Conditions (4) require the precedence relations between operations. If $x_{ik} = 0$ (operation i is not running at the station k), then $\sum_{j=1}^{k} x_{ik}$ may take any value of 0 or 1 and (4) becomes $\sum_{j=1}^{n} x_{sj} \ge 0$, it is always true and not represent a constraint. If $x_{ik} = 1$, then (4) is equivalent to (2). Restrictions (5) means the following: if $y_k = 0$, then $\sum_{i=1}^{n} x_{ik} \le n$, which relationship is always satisfied. Thus (1) gives the number of "empty" jobs, i.e. the number of stations with $\sum_{i=1}^{n} x_{ik} = 0$, stations that do not perform any operation. Conditions (6) force x_{ij} and y_i to be binary.

Various heuristic and exact methods ([1]), ([4]), ([5]), ([6]) have been proposed to solve the zero-one linear program (1)-(6).

3 A second order-cone programming reformulation

The conditions (6) are equivalent to the non-convex quadratic constraints:

$$\left.\begin{array}{l} y_i^2 - y_i = 0, \forall i \\ x_{ij}^2 - x_{ij} = 0, \forall i,j \end{array}\right\}. \tag{7}$$

It can be established that the constraints (7) are equivalent with:

$$\left.\begin{array}{l} \sum_{i=1}^{n} y_i^2 \ge \sum_{i=1}^{n} y_i, \\ 0 \le y_i \le 1, \forall i \end{array}\right\}, \tag{8}$$

and

$$\left.\begin{array}{l} \sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}^2 \ge \sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}, \\ 0 \le x_{ij} \le 1, \forall i,j \end{array}\right\}. \tag{9}$$

Applying the inequality:

$$\sum_{i=1}^{n} a_i \geq \sqrt{\sum_{i=1}^{n} a_i^2} ,$$

true for $\forall a_i \geq 0, a_i \in R$, from (8) we have:

$$\sum_{i=1}^{n} y_i^2 \geq \sum_{i=1}^{n} y_i \geq \sqrt{\sum_{i=1}^{n} y_i^2} .$$

Taking into account that the variables y_i are binary, from the last inequalities we obtain the second-order cone of dimension $(n+1)$ (Lorenz cone):

$$\mathrm{K}_1 = \left\{ \begin{bmatrix} z \\ y \end{bmatrix} : z \in R, y \in R^n \ for \sqrt{\sum_{i=1}^{n} y_i^2} \leq z \right\},$$

where $z = \sum_{i=1}^{n} y_i$ şi $y = (y_1, y_2, \mathrm{K}, y_n)^T$.

In the same way, from (9) we obtain:

$$\sqrt{\sum_{i=1}^{n} \left(\sum_{j=1}^{n} x_{ij} \right)^2} \leq \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} \leq \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij}^2 .$$

Thus (9) determines the second-order cone:

$$\mathrm{K}_2 = \left\{ \begin{bmatrix} u \\ X \end{bmatrix} : u \in R, X \in R^n \ for \sqrt{\sum_{i=1}^{n} X_i^2} \leq u \right\}.$$

Above we used the notation:

$$X_i = \sum_{j=1}^{n} x_{ij}, i = 1,2,\mathrm{K}, n , \ X = (X_1, X_2, \mathrm{K}, X_n)^T , \text{ and } u = \sum_{i=1}^{n} X_i .$$

Thus, we can reformulate the problem (1) - (6) in terms of second-order cone programming:

$$z \rightarrow minimize ,$$

subject to restrictions (2) - (5), and

$$\sqrt{\sum_{i=1}^{n} y_i^2} - z \leq 0,$$

$$\sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij}^2} - u \leq 0,$$

$$\sum_{i=1}^{n} y_i - z = 0,$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij} - u = 0,$$

$$0 \le y_i \le 1, i = 1,2,\text{K}\ ,n,$$

$$0 \le x_{ij} \le 1, i, j = 1,2,\text{K}\ ,n.$$

4 Conclusions

The paper contains a reformulation of the simple assembly line balancing problem in terms of second-order cone programming. This problem can be effectively solved by the interior point algorithm ([7]). There is specialized software for solving conic optimization problems ([8]).

References

[1] I. Baybars. A survey of exact algorithms for the simple line balancing problem. *Management Science*, *32*, 909-932, 1986. http://www.jstor.org/stable/2631657.

[2] A. School, *Balancing and Sequencing of Assembly Lines (series: Contributions to Management Science),* Physica-Verlag Heidelberg, 2-nd edition, 1999.

[3] T. K. Bhattachaje, S. Sahn. Complexity of Single Model Assembly Line Balancing Problems. *Engineering Cost and Production Economics*, *18*, 203-214, 1990.

[4] A. School, C. Becker. State of the Art Exact and Heuristic Solution Procedures for Simple Assembly Line Balancing. *European Journal of Operational Research*, *Vol. 168*, No. 3, 666-693, 2006.

[5] A. L. Arcus. COMSOAL a Computer Method of Sequencing Operations for Assembly Lines. *The International Journal of Production Research, Vol. 4,* No. 4, 259-277, 1996.

[6] E. Erel, S. C. Sarin. A Survey of the Assembly Line Balancing Procedures. *Production Planning and Control, Vol. 9,* No. 5, 414-434, 1998.

[7] Yu. Nesterov, A. Nemirovsky. Interior-point polinomial methods in convex programming. *Volume 13 of Studies in Applied Mathematics.* SIAM, Phildelphia, PA, 1994.

[8] H. D. Mittelmann. The State-of-the-Art in Conic Optimization Software. *In Handbook on Semidefinite, Conic and Polynomial Optimization, Volume 166*, 671-686, 2012.

Vasile Moraru
Technical University of Moldova
Applied Informatics Department
168, Stefan cel Mare str., Chisinau, 2004
MOLDOVA Republic of
E-mail: moraru@mail.utm.md

Sergiu Zaporojan
Technical University of Moldova
Computer Science Department
168, Stefan cel Mare str., Chisinau, 2004
MOLDOVA Republic of
E-mail: zaporojan_s@yahoo.com

Third International Conference
Modelling and Development of Intelligent Systems
October 10 - 12, 2013
"Lucian Blaga" University
Sibiu - Romania

Comparative Study in Building of Associations Rules from Commercial Transactions through Data Mining Techniques

Mircea-Adrian MUŞAN, Ionela MANIU

Abstract

In this paper we have built processes for extracting data mining association rules based on frequent sets of articles from commercial transactions. We used as a working data set a database created from online retail transactions. Based on the particularities of processes built, we performed a statistical analysis to illustrate the efficiency, precision and accuracy of data mining techniques used.

1 Introduction

The growing interest for Data Mining domain can be motivated through the pressing need, common to many areas of reference, to describe, to model and, especially, to understand large sets of data.
The process of knowledge discovery is equally old as the cerebral man. Since the discovery of fire and reaching out to the current studies on marketing, man made "data mining" without realizing it. Today, aided by tremendous computing power of computers, it can now adventure in exploring information by using the most effective means of working with existing data.

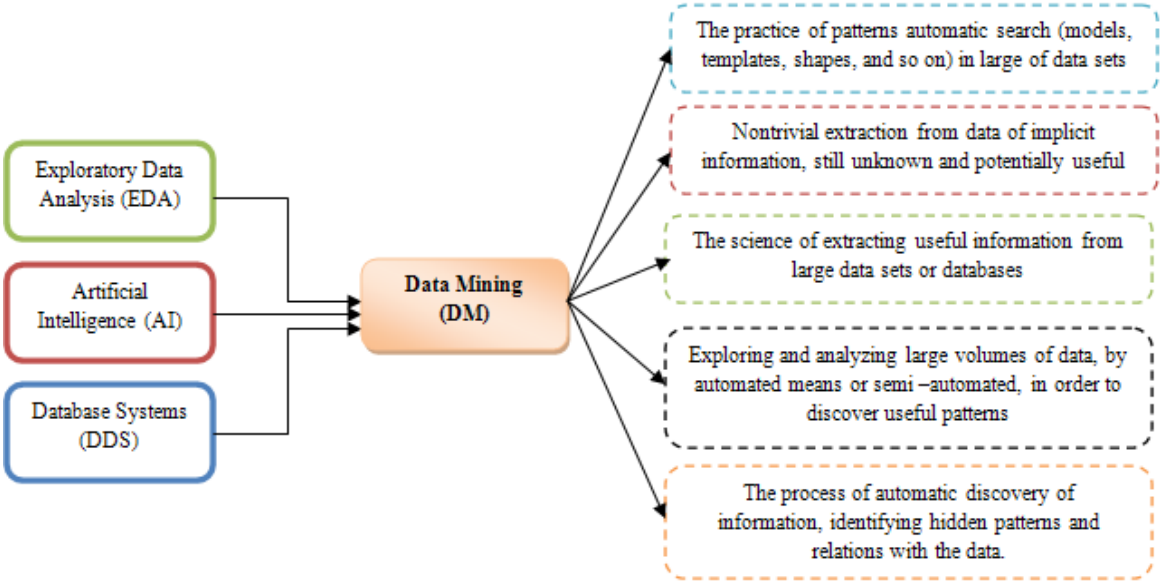


Figure 1 – Roots and significance of data mining

As can be seen in *Figure 1*, it is difficult to formulate a single definition for data mining. On base of term roots (presented in *Figure 1*), namely, exploring data analysis, artificial intelligence and database systems, the most frequently encountered significance for the concept of "data mining" is, in a few words, "knowledge-discovery in databases" (KDD), as it is named in work [1]. Another definition, in the same work, is "extraction of interesting (non-trivial, implicit, previously unknown and potentially useful) patterns or knowledge from huge amount of data".
A significant category of data mining techniques is that of mining frequent patterns, associations and correlations. Algorithms built for association rules are very useful from the perspective of marketing, because they develop methods for finding customers shopping patterns [6]. Applications of these special techniques are in basket data analysis, cross-marketing, catalogue design, sale campaign analysis, click stream of web logs analysis, and DNA sequence analysis [1]. From marketing perspective, the workings of these techniques are simple: purpose is to find correlations between articles sold. Association rules are based on two measures which quantify the support and confidence of the rule for a given data set.
The use of these techniques is to find trends and correlations in databases, which helps experts to take correctly and efficiently decisions in the future.
The second section of this paper presents an experiment based on the algorithm used; the third describes the processes constructed and the results obtained. A statistical analysis of the results was performed in the last section.

2 Experiment based on FP-Tree algorithm for our data set

The database after which we will make the processing is described in *Section 3* of this paper. To extract association rules from the products traded we chose operation by FP-Growth technique. The FP-Growth algorithm, that means Frequent Pattern Growth Algorithm, was developed by J. Han, H. Pei, and Y. Yin [8]. This efficient, fast and scalable algorithm [6] is a method for mining the complete set of frequent patterns by pattern fragment growth, using an extended prefix-tree structure for storing essential information about frequent patterns, named FP-tree [8].
Among the advantages offered by this algorithm, we can mention that it is one of the fastest in obtaining association rules, as J. Han wrote in his paper [8], that the FP-Growth algorithm has better performance than AprioriAlgorithm [3], Tree-Projection [9], RElim [10] or Eclat [11]. It also has disadvantages, namely, it is more difficult to implement that other approaches like a complex data structure and an FP-tree, it can need more memory than a list of transactions [2].

In our case study we randomly selected a set of ten transactions made. In order to not have a very wide range of products, articles of transactions we considered as being the name of category from which these products belong. Thus, if in a certain chosen transaction we will find at least two products from the same category, we keep only one item, namely, the name of that category. After all this, we obtained the following table:

Tr. ID	**List of product categories**
0	Hair, Women fragrance, Tools and brushes
1	Skin care, Women fragrance
2	Skin care, Men fragrance
3	Hair, Men fragrance, Women fragrance, Bath and body
4	Women fragrance, Bath and body
5	Skin care, Men fragrance
6	Hair, Skin care, Women fragrance
7	Men fragrance, Women fragrance
8	Skin care, Men fragrance, Bath and body
9	Hair, Skin care, Women fragrance

Table 1 – Set of transactions chosen by category name

After establishment of the list of transactions, it moves to the next level, namely, determination of frequency of individual items. In our case we have the next list: *Hair* – 4, *Skin care* – 6, *Men fragrance* – 5, *Women fragrance* – 7, *Bath and body* – 3 and *Tools and brushes* – 1.
The next step is that of sorting descending items in transactions and removing those items that are infrequent for the parameters chosen in our case, in our case *Tools and brushes*. After that, the transactions are sorted lexicographically in ascending order. This is the last step before construction of frequent patterns tree. The result is presented in *Figure 2*.

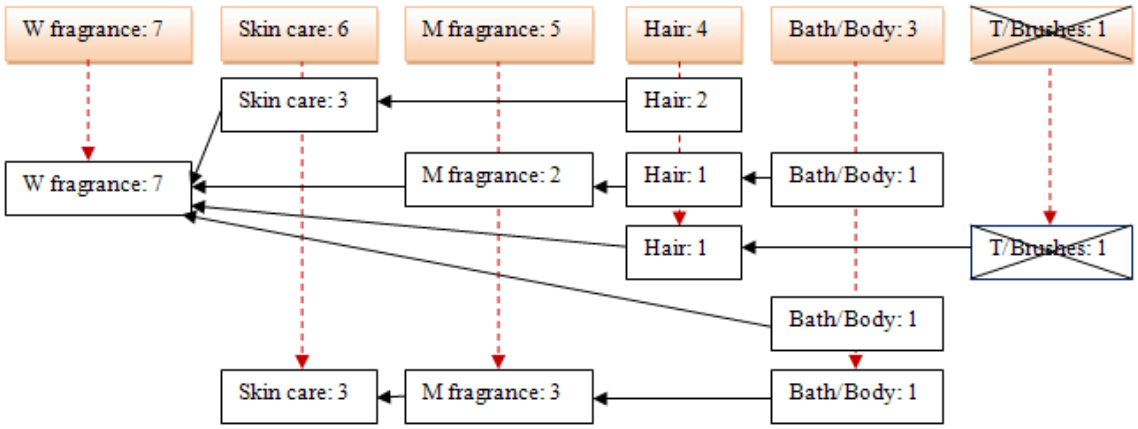


Figure 2 – FP Tree. Representation of transactions

The results obtained using FP-Tree algorithm, for our experiment, are presented in *Table 2*. For this, a value for minimum support (50%) is selected, relevant for our case. For establishing frequent sets of items we have chosen the same display mode of results like application RapidMiner.

Size	Support	Item 1	Item 2	Item 3
1	0.700	Women fragrance		
1	0.600	Skin care		
1	0.500	Men fragrance		
1	0.400	Hair		
1	0.300	Bath and body		
2	0.300	Women fragrance	Skin care	
2	0.200	Women fragrance	Men fragrance	
2	0.400	Women fragrance	Hair	
2	0.200	Women fragrance	Bath and body	
2	0.300	Skin care	Men fragrance	
2	0.200	Skin care	Hair	
2	0.200	Men fragrance	Bath and body	
3	0.200	Women fragrance	Skin care	Hair

Table 2 – Displaying of frequent sets

3 Presentation of the process built

3.1 Description of the process

In our process programming we used RapidMiner. RapidMiner assures data mining and machine learning procedures, such as: pre-processing and visualization of data, transformation, modelling, evaluation, and deployment of data. RapidMiner is written in the Java programming language and uses learning schemes and attribute evaluators from the Weka machine learning environment and statistical modelling schemes from R-Project [7].
RapidMiner contains a collection of modular operators which allow the design of complex processing for a large number of data mining problems. The most important characteristic of

RapidMiner is the ability to imbricate operators' chains and building trees of complex operators. To support this feature, data core of RapidMiner acts as a databases management system.
In order to program the desired process for analyzing associations of appearances frequent transaction sets, we used a dataset of an e-commerce company, operating in the field of perfumery and personal care products.
We chose a database of a company that operate exclusively online for several reasons, namely: the first reason is that the data set proposed for marketing is homogeneous and the second is that most transactions are composed of at least 2 – 3 products, and that from economic reasons related to saving the transport costs, promotions on the number of products, reasons helpful in analyzing associations based transactions.
The dataset used (file named Fragrance.xls) has over 1,500 lines, overall 500 transactions, and the following structure of fields:

- *Current number of record* (auto number)
- *ID of product* (a numerical value between 101 and 999)
- *Name of product* (nominal value)
- *ID of category* (a numerical value between 1 and 99)
- *Name of product's category* (nominal value)
- *ID of transaction* (a numerical value)

Based on the dataset described above, we developed a data mining process developed using Rapid Miner, which will determine sets of frequent appearances from transactions, on which are generated association rules. Process built is based on the drawings of other processes constructed through the work [3] [4], being shown in *Figure 3*.

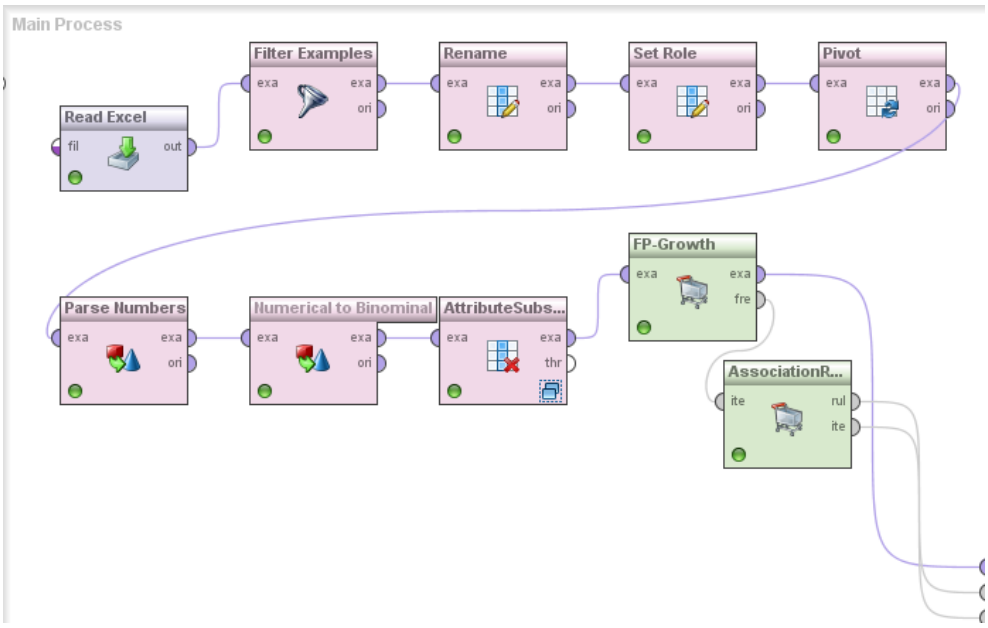


Figure 3 – The RapidMiner process for determining sets of frequent appearances and association rules generated

For this process writing, created by facility GUI of RapidMiner and refined programmatically through adequate XML code, for reasons of space, it will not be exposed in this article, we used the following operators:

- **Read Excel**, through which we took the dataset, but eliminating those elements deductible, namely the information about product categories, and that is not to force later shift of all attributes in a binomial form, which is essential in process development .
- With help of **Filter Examples** we have split the initial data set in several samples, depending on the selected product categories and units of time in which transactions were made. Thus, many results were generated, which are interpreted and analyzed in the following sections.

- **Rename** is used to rename fields, which by their nature, participate directly in the results, and as such, would hold their visibility.
- **Set Role** is used by RapidMiner to change the role of one or more attributes. In our case we put value *ID of transaction* to field ***attribute name***, ***target role*** received value *id*. Through option ***set additional roles*** we have established *Name of product* as being *regular* type.
- **Pivot** is an important operator of this process and we used it to rotate the example set by grouping multiple examples of same groups to single examples. By option ***group attribute*** we selected the field *ID of transaction*, by ***index attribute*** we have chosen the field *Name of product* and through ***weight aggregation*** we selected the option *count*.
- **Parse Number** is an operator auxiliary in this process, which we wrote for applying the next operator from our process for all data.
- **Numerical to Binomial** changes the type of the selected numeric attributes to a binominal type. It is an essential operator from this process, because the operator with name ***FP-Growth*** works only with binomial values. Because we applied before the operator ***Parse Number***, in this case we chose the option *all* for the option ***attribute filter type***.
- **Attribute Subset**, through which a subset is selected, composed of one or more attributes, from the input dataset and applies the operators in its subprocess on the selected subset.
- **FP-Growth** is a central operator of our construction. It calculates all frequent itemsets from the given dataset using the *FP-tree* data structure. The range of values within which we chose *minimum support* for establishing frequent sets of items is described in *Section 4*.
- **Association Rule Generator** (**Create Association Rule**) was written to obtain the association rules generated based on frequent occurrences of articles in transactions as they have been previous outcomes by using of operator, FP Growth. Data related to the values received by *minim confidence* attribute will be reported in *Section 4*, these constituting the support for the hypothesis of statistical analysis based on the results obtained. For this operator we have selected the output port *ite*, in order to see the frequent item sets obtained by FP-Growth operator.

3.2 The results obtained

In order to show the results, we selected a value for minimum support and one for minimum confidence. For reasons of space, we chose the maximum values from the range presented in *Section 4*, resulting the case with the fewest rules of associations obtained. So, we have chosen minimum support as 12%, and minimum confidence as 50%, as an example. The results generated by RapidMiner can be observed in the following list. It is quite obvious that if we choose smaller values for the parameters indicated, the number of generated rules will be greater, but including among themselves the rules presented below.

```
Association Rules
[Body lotions & body oils] → [Men's perfume] (confidence: 0.500)
[Shampoo, Conditioner] → [Gift sets] (confidence: 0.500)
[Shampoo] → [Conditioner] (confidence: 0.517)
[Teeth whitening, Hair color] → [Cellulite & stratch marks] (confidence:
0.524)
[Hand & foot cream] → [Men's perfume] (confidence: 0.560)
[Eye brushes] → [Women's perfume] (confidence: 0.562)
[Men's perfume, Teeth whitening] → [Night cream] (confidence: 0.571)
[Lip brushes] → [Women's perfume] (confidence: 0.605)
[Eye brushes] → [Face brushes] (confidence: 0.625)
[Conditioner] → [Shampoo] (confidence: 0.667)
[Teeth whitening, Moisturizer] → [Hair color] (confidence: 0.688)
[Men's perfume, Night cream] → [Teeth whitening] (confidence: 0.750)
[Day cream, Hair color] → [Teeth whitening] (confidence: 0.786)
[Gift sets, Shampoo] → [Conditioner] (confidence: 0.789)
[Face brushes] → [Eye brushes] (confidence: 0.833)
[Gift sets, Conditioner] → [Shampoo] (confidence: 0.938)
```

List 1 – The results obtained by RapidMiner on our data set

These results were obtained from *Text View* option, generated by RapidMiner. If it is desired a visualisation more complete of the results, one can choose *Table View* option. For example, first association rule has *Support* 0.050, *LaPlace* 0.955 and *Gain* -0.150.
At the results interpretation, the following rules can be obtained: "for the premise *Body lotions & body oils* at least 12% of the customers of this product always buy *Men's perfume*, obtained by the attribute conclusion" or "at least 12% of customers of *Shampoo* and *Conditioner* always buy *Gift sets*".

4 Statistical representations and analysis based on the results

In this paper, for the experimental part, we used Rapid Miner pre-programmed process presented in *Section 3*, executing it several times. Execution numbers has been determined by a variety of values for the process key attributes: set minimum support and guaranteed minimum confidence. For our experiment, we used a minimum support of a range of values from the set [0, 0.12], which are in arithmetic progression with the ratio 0.02, each of which is assigned to a value from the set [0, 0.5] in arithmetic progression with ratio 0.05, for minimum confidence attribute. Thus 78 runs resulted in the process, enough to carry out a statistical experiment, based on our inputs.
Then we divided the data set in three equal parts according to chronological data entry, relying on a statistical survey of consumer behaviour on basis of joint sets transactions frequent occurrences in certain units of time.
The process was repeated for each of the three cases, yielding filtering through *Filter Examples* operator, so that attribute *condition class* receive value *attribute_value_filter* and *parameter string* corresponding filter condition. Running three new executions has been done using the same value range *minim Support – minim Confidence* as in process running on the full data set.
Based on ideas presented in the article [5], we used as a premise in our work, in statistical evaluation, the maximum of items present in the transaction and the average number of items per transactions. This gave the following table:

Transaction	Trial run on entire data set	Trial running on first data set	Trial running on second data set	Trial running on third data set
Maxim	9	6	6	9
Mean	3,340796	3,306931	3,207961	3,507538

Table 3 – The evidence of items from transaction

For the entire database a negative correlation was obtained linking the parameters support (r = -0.262, 0.043), confidence (r = -0.902, p = 0.000) and number of association rule, with a stronger link between confidence and rules number.

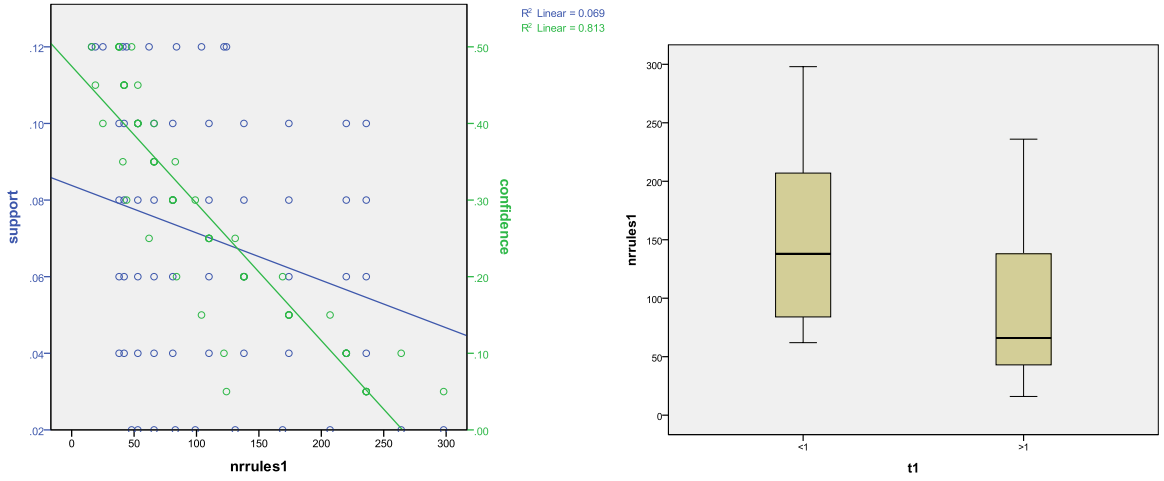


Figure 4 – Number of association rules vs. support – confidence and time, for the entire database

In case of database division, when the number of association rules is smaller, a negative correlation was kept between both parameters support (r1 = -0.505, p = 0.000), the confidence (r2 = -0.717, p = 0.000) and the number of association rules, having strong correlation between the confidence and the rules number. For a higher number of association rules, negative correlation are preserved between both parameters (r1 = -0.586, p = 0.000, r2 = -0.323, p = 0.012) and the number of association rules, but the link is strongest between the support and the association rules number.

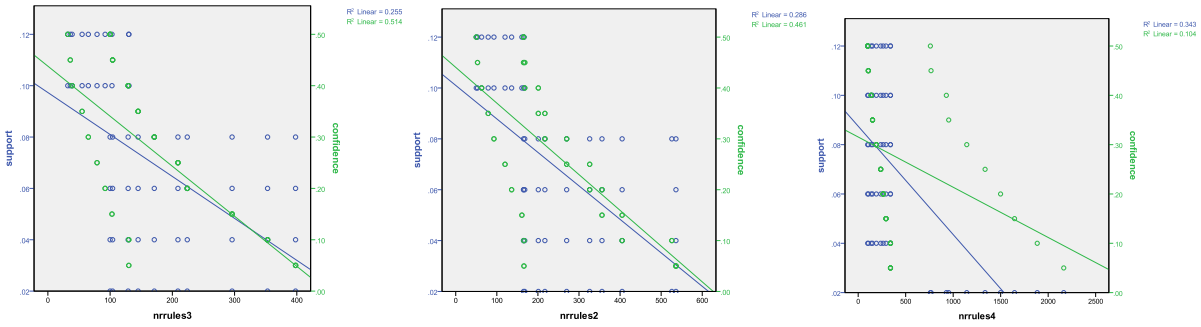


Figure 5 – Number of association rules vs. support – confidence, for database divisions

It is observed that for the same number of transactions (taken on time series) with the same products range, and the same *minim Support – minim Confidence*, a larger generated number of association rules does not involve a higher execution time, as can be seen in *Figure 6*, where on x-axis is represented the execution time and on y-axis the number of association rules generated.

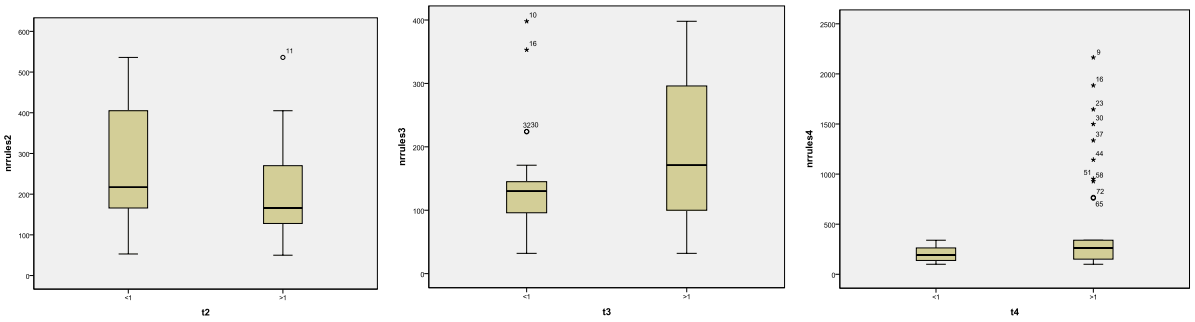


Figure 6 – Number of association rules vs. time, for database divisions

5 Conclusions

Techniques for determining the association rules are some very powerful tools in making decisions of marketing. Thus, based on them can be established some promotional packages, certain promotions, web page layout or arranging on the shelf, or simply, can track some trends of consumers. About the execution time of the association rules processes, the statistic analysis concludes that a larger generated number of rules does not involve a higher execution time.

References

[1] Jiawei Han, Micheline Kamber, *Data Mining: Concepts and Techniques, 2nd ed.*, The Morgan Kaufmann Series in Data Management Systems, Jim Gray, Series Editor Morgan Kaufmann Publishers, March 2006. ISBN 1-55860-901-6, http://www.cs.uiuc.edu/homes/hanj/bk2/slidesindex.htm

[2] Ch. Borgelt, *Frequent Pattern Mining*, Intelligent Data Analysis and Graphical Models Research Unit European Center for Soft Computing, 33600, Mieres, Spain, 2005

[3] Daniel Hunyadi, *Performance Comparison of Apriori and FP-Growth Algorithms in Generating Association Rules*, The 5th European Computing Conference (ECC' 11), Paris, France, April 28-30 2011, pp. 376-381, 978-960-474-297-4
[4] Mircea Muşan, *Versatile integration of data mining techniques of description and prediction in Web informatics systems of Business Intelligence*, Second International Conference "Modeling and Development of Intelligent Systems", Sibiu, Romania, September 29 – October 02, 2011, pp. 97-104, ISSN 2067-3965, ISBN 978-606-12-0243-0
[5] Pratiksha Shendge, Tina Gupta, *Comparative Study od Apriori & FP Growth Algotithms*, Indian Jounal of Research, PARIPEX, Volume: 2, Issue: 3, March 2013, pp. 20-22, ISSN 2250-1991
[6] Jerzy Korczak, Piotr Skrzypczak, *FP-Growth in Discovery of Customer Patterns*, Data-Driven Process Discovery and Analysis Lecture Notes in Business Information Processing Volume 116, 2012, pp 120-133, Print ISBN 978-3-642-34043-7, Online ISBN 978-3-642-34044-4
[7] "*RapidMiner*", Rapid-i. Retrieved 7 March 2011, http://rapidminer.com/products/rapidminer-studio/
[8] J. Han, H. Pei, and Y. Yin, *Mining Frequent Patterns without Candidate Generation*, In: Proc. Conf. on the Management of Data (SIGMOD'00, Dallas, TX). ACM Press, New York, NY, USA 2000
[9] Agarwal,R.C., Agarwal C.C., Prasad, V.V., *A Tree Projection Algorithm For Generation of Frequent Itemsets*. Journal on Parallel and Distributed Computing, vol. 61, 2000
[10] Christian Borgelt, *Keeping Things Simple: Finding Frequent Item Sets by Recursive, Elimination*, Workshop Open Source Data Mining Software (OSDM'05, Chicago, IL), 66-70, ACM Press, New York, USA 2005
[11] Zaki, M.J., Parthasarathy, S., Ogihara, M., Li, W., *New Algorithms for Fast Discovery of Association Rules*. ACM SIGKDD (1997)

Mircea-Adrian MUŞAN
"Lucian Blaga" University of Sibiu
Mathematics and Informatics
Sibiu, Street Ion Rațiu, No. 5
ROMANIA
E-mail: musanmircea@yahoo.com

Ionela MANIU
"Lucian Blaga" University of Sibiu
Mathematics and Informatics
Sibiu, Street Ion Rațiu, No. 5
ROMANIA
E-mail: ionela.maniu@yahoo.ro

Third International Conference
Modelling and Development of Intelligent Systems
October 10 - 12, 2013
"Lucian Blaga" University
Sibiu - Romania

The dangers of Social Media. A case study on children age 10 to 12

Alina Elena Pitic, Ioana Moisil, Călin Bucur

Abstract

Since the year 2000, The Children's Online Protection Act (COPA) tries to protect children under the age of 13. However, nowadays millions of children below 13 years old are active users of Social Media. We show some of the dangers the children are exposed and we try to propose some safe alternatives. We conducted a case study on our local community by asking primary schools children from Sibiu (Romania) questions about their involvement in social media. We present some initial results from an ongoing study, started in the late 2012.

Keywords: Social Media, social consciousness, community question answering

1 Introduction

1.1 What is Social Media

We can find one of the most comprehensive definition of Social Media in [1]: "Social media encompasses a wide range of online, word-of-mouth forums including blogs, company-sponsored discussion boards and chat rooms, consumer-to-consumer e-mail, consumer product or service ratings websites and forums, Internet discussion boards and forums, moblogs (sites containing digital audio, images, movies, or photographs), and social networking websites, to name a few." Some examples of Social Media can be found in Table 1.

Type of Social Media sites	**Example**
Social Networks	Facebook
Multimedia (video) sharing	YouTube
Photo sharing	Flickr
Business Networking	LinkedIn
Collaborative sites	Wikipedia
Virtual world simulations	Second Life
Commerce communities	eBay
Micro-blogging	Twitter

Table 1. Examples of Social Media

Many researchers focus their work on the opportunities offered by the continuous growth of the Social Media Networks ([2], [3], [4])

However, there is only little research (or even concern) on the dangers of the Social Media. A simple experiment demonstrates this. We use Google Search engine with two strings: "opportunities of social media" and "dangers of social media". In the first case we received 2.260

results, while in the second one only 82 results. Most of the research on the impact of Social Media over the children and young people is focused on the social aspects ([5]).

1.2 Some of the benefits of Social Media

There are three different cases on which we can discuss about the benefits of Social Media: the children, the teacher and the parent.

1.2.1 The benefits for the children

- Children have a way to see the influence of their social networks on school activities and the other way around
- They can connect their academic and social actions to a wider horizon of information
- Realize that their online actions leave a digital footprint and teaches them responsibility, good citizenship, safety or the importance of a good reputation
- Makes classrooms more engaging through a larger horizon of voices (ex: Youtube, Twitter, Facebook, Skype)
- Develop collaborative skills (very important asset in this century) – children work in groups, teams and comment on each other's tasks and can engage in discussions with their colleagues or teachers
- Receive the skills needed not only for highschool but in a future carrier also
- Encourage and reward the use of technology in school and in their life in general

1.2.2 The benefits for the teachers

- Brings the community of teachers together to better handle education related issues
- Friendly environment to discuss new ideas and exchange information
- Inspiration for new ideas in the classroom
- Increases professional engagement and spurs continuing education and training
- Boosts inter-cultural and cross-cultural information exchange in the field of education

1.2.3 The benefits for the parents

- Boosts dialogue between teachers and their students
- Increased presence in the classroom and greater control of the curriculum
- Ensures greater transparency for educational units
- Helps in better grasping teacher and class expectations

1.3 Some of the dangers of Social Media

The activity and behavior of children should be monitored by parents all the time. They are most vulnerable to social media dangers and they should be kept away from possible tempting illegal activities.

One of the biggest threats is represented by so called predators hiding under fake Social Media profiles. It is now easy for predators to pose as another child having the same interests as our children. If a kid accepts a predator as a friend, he can find all the information he wants to know, such as the age or the real home address.

The child identity theft, another cyber-crime, is increasing at a high rate in the last years. It is much easier to set up a phishing scheme for a child, especially if the kid has access to a credit card or if the parents use their credit card information using the child account (for instance to buy some online game items).

Cyber bullying and Social Media harassment represent dangers that appears mostly because there is no physical presence, making the children less inhibited to say inappropriate things.

2 Social Media for children

If we want a safe alternative to Facebook and other similar Social Networks we should have in mind safety and age-appropriate fun. To build a social network for children we should include all the traditional features provided by standard social networks, like chat, multimedia upload, profiles, comments, ratings. Furthermore, additional features like games, contests and virtual gifts should be considered. Maybe the most important thing is that social networks for children should include is the parental monitoring feature.
In [6] we have found a list of safe social networks for children.

Site	Age	Observation
ScuttlePad	7+	intended to create a safe online space for children
Togetherville	7+	permits connection to Facebook friends of the parents
Yoursphere	9+	it has a strict membership rules and all sign-ups are vetted against a database of registered sex offenders
Franktown Rocks	10+	creates a safe spot for kids to listen music
GiantHello	10+	similar with Facebook, but with much more attention to safety and privacy concerns for children
GirlSense	10+	closed at the moment we write this paper
Sweety High	11+	social network for girls witha very good privacy.

Table 2. Safe social networks for children

3 Case study

3.1 Social Media usage

Some of our prior case studies emphasize the idea that the children are attracted on different activities using the computers, especially games and social networking. Figure 1 shows some of our results. First figure represents the answer to the question "I'm using the computer for…", the second one "I'm using the computer … hours/day" and the last question is "What are you searching on the internet?".

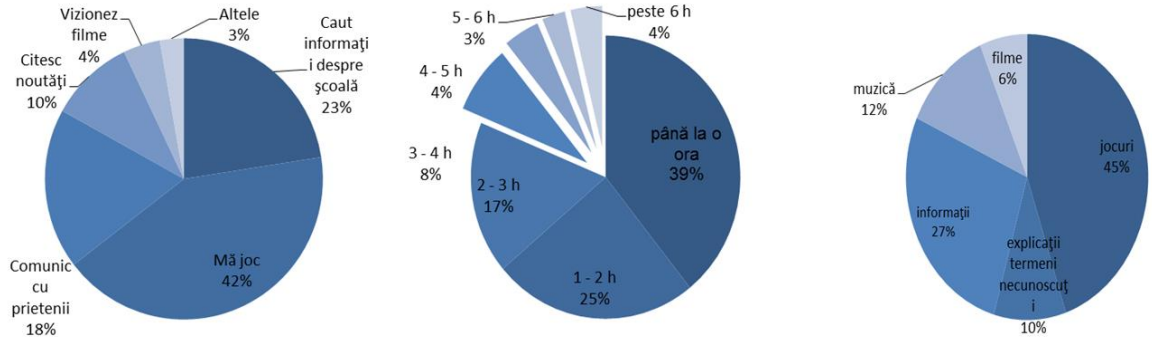


Figure 1 – Some initial results

The study was conducted in 2011 over almost 400 children ages 8 to 12. One important observation from this study was the fact that at age 10+ more and more children use computers to communicate with their friends and they exceed the 2 hours/day of using the computer. However, out study did not ask the children about their online social behavior.

An ongoing study, started in the late 2012, is about the way children aged 10 to 12 use the social networks and how it can affect their life. We obtained results from around 80 children, but more

results are yet to come. First question was a multiple choice one: "Select the Social network you've heard about" (Figure 2).

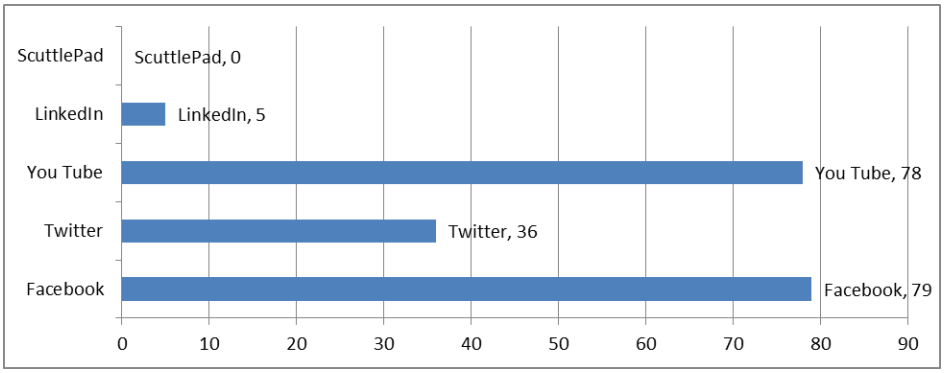


Figure 2 – Select the Social network you've heard about

The second question was "I'm using… at least once a week". The results can be observed in Figure 3.

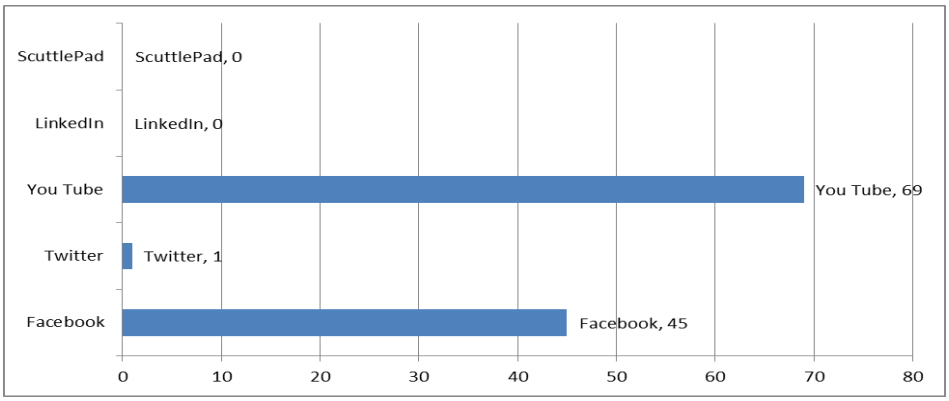


Figure 3 – Usage of the Social Networks

Even if You Tube has much better results than Facebook, we have we can assume that only a small number of the children are registered You Tube users, even if we did not specifically ask it.

We ask about the Social Networks designed for children (Table 2), but **none** of the children even heard about them.

3.2 The dangers of Social Media

Our study focused on the main three dangers of the Social Media for the kids: predators, phishing scams and cyber bullying.

We have asked the children "Did you talk on-line with strangers?" (Figure 4). The results are selected from the children that have an active FaceBook account.

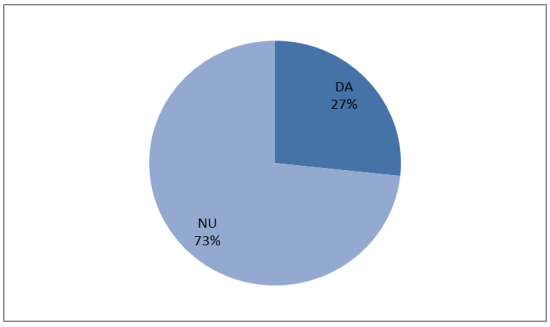


Figure 4 – "Did you talk on-line with strangers?"

However, the children consider that if an on-line profile is on friends' list means that person is not a stranger.

The next question was: "I know what a phishing scam is" (Figure 5).

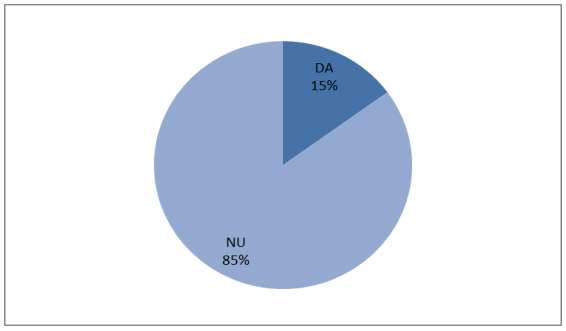


Figure 5 – "I know what a phishing scam is"

In this case it is clear that only a few children are aware of the possibility of virtual theft. Anyway, the best way of avoiding this is to never let the children to buy from the internet without adult supervision.

The last question was about cyber-bullying: "Did you sent any mean messages?" (Figure 6)

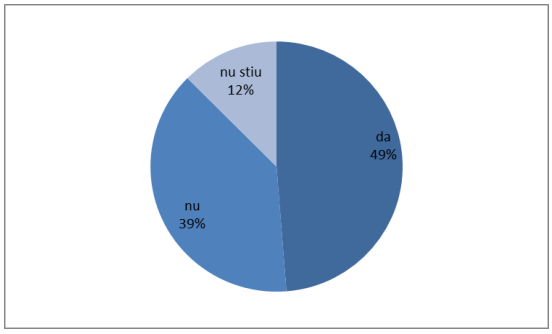


Figure 6 – Did you sent any mean messages?

The children, hiding behind computer, are feeling free to be mean with others, to provoke strangers or friends or to invent fictional things about peoples or situations.

4 Conclusions

Even if we still need to gather more results some conclusions can be made:

- The children under the age of 13 use Facebook, some of them on a daily basis
- The children are not aware of the dangers lurking beneath social networks
- The parents are not aware of the dangers of using social networks

The main question that still remains to be answered is "What do we do to change this situation?"

References

[1] W. Glynn Mangolda, David J. Fauldsb, *Social media: The new hybrid element of the promotion mix*, Business Horizons, Volume 52, Issue 4, July–August 2009, Pages 357–365.

[2] Andreas M. Kaplan, Michael Haenlein. *Users of the world, unite! The challenges and opportunities of Social Media*. Business Horizons, Volume 53, Issue 1, January–February 2010, Pages 59–68
[3] http://research.microsoft.com/en-us/projects/socialmedia/
[4] http://www.rin.ac.uk/our-work/communicating-and-disseminating-research/social-media-guide-researchers
[5] Gwenn Schurgin O'Keeffe, Kathleen Clarke-Pearson, *The Impact of Social Media on Children, Adolescents, and Families,* PEDIATRICS Vol. 127 No. 4 April 1, 2011, Pages. 800 -804
[6] http://www.npr.org/2011/07/11/137705552/ten-safe-social-networking-sites-for-kids
[7] Pitic Elena Alina, Moisil Ioana, Dzitac Simona, *Rasing Energy saving awareness through educational software* , IJCCC2013, VOL. 8, ISSUE 2, pg. 255-262, 2013

Alina Elena Pitic
Lucian Blaga University of Sibiu,
Faculty of Sciences,
Department of Mathematics and Informatics
Sibiu, 5-7, Ion Rațiu Street
ROMANIA
E-mail: alinap29@yahoo.com

Ioana Moisil
Lucian Blaga University of Sibiu,
Faculty of Sciences,
Department of Mathematics and Informatics
Sibiu, 5-7, Ion Rațiu Street
ROMANIA
E-mail: im25sibiu@gmail.com

Călin Bucur,
Lucian Blaga University of Sibiu,
Faculty of Economics,
Sibiu, 17, Calea Dumbrăvii,
ROMANIA
E-mail: im25sibiu@gmail.com

Third International Conference
Modelling and Development of Intelligent Systems
October 10 - 12, 2013
"Lucian Blaga" University
Sibiu - Romania

Methodological Framework for Creating a Workflow Model when Processing Data Research

Alexandra-Mihaela Pop, Ioan Pop

Abstract

In this article we present a methodological approach for creating workflow models. Using a workflow model for data processing research can be considered a tool for quantitative management into project. The advantage of the approach of a workflow model in the research is that the data collected by investigating a large population studied, is processed automatically, dynamically and executed with real-time machine learning methods. In practice, this approach leads to the enrichment of the "Project Management Body of Knowledge". Also the methodological framework can serve as a tool to initiate and train students interested in developing research projects. The article describes a scenario for creating a workflow model for communication management.

1 Introduction

Processing data from different research processes involves a computationally intensive, statistical analysis and interpretation techniques based on machine learning methods execution. To speed up and automate data processing it is necessary to create workflows that support the researcher or expert. Workflows can be created with different toolboxes specific data processing tasks. There are toolboxes for creating workflows for business - business workflows - but also for scientific research - scientific workflows. Scientific workflows can be composed of steps that follow the stages of a research process such as: acquisition, integration, reduction, analysis, visualization, and publication (e.g. in a shared database) of scientific data [02].

For researchers a scientific workflow is a model that can assure the automatization of data processing by executing the model in a repetitive way. Also, a scientific workflow is an additional support for monitoring the execution of algorithmic methods in real-time.

In the first part of this article we present ways to model the processes researched, a mathematical model of a workflow, and a framework for creating a workflow.

The second part is devoted to using the techniques for creating workflows with the help of the Weka toolbox.

In the last part of the article we present a scenario in which we created a working model of such a workflow. Workflow model execution from the case study helps a researcher focus on the analysis of communication in project management. Based on the results of workflow execution, the researcher can draw interpretations motivate members by communicating in the project.

2 Modeling as a tool for manipulation the applicative and constructive entities

2.1 Ways of modeling

Natural or artificial processes are the object study of people concerned with increasingly discovering reality. In order to be understood and interpreted correctly these processes are represented by different models. As abstractions of essential attributes, models are characterized by a certain fidelity of the processes but the most important thing is that they help by simulating their execution in obtaining reliable solutions obtained after processing.

Modeling is done in various forms in relation to the methods used in the field where it is produced. For example, in architecture iconic designs is used, while in economics symbolic modeling is used (through mathematical models). All modeling techniques are based on the mechanism of abstraction of things, phenomena and processes. Thus, obtaining an execution flow model for data processing is a good method for handling, automation and optimization of these processes. Methodologically speaking, designing a workflow model involves identifying the basic elements of the new workflow, assess and document key characteristics of process modeling, i.e. the following steps [04]:

- Outline the workflow;
- Detailing the various levels of work;
- Evaluation and verification of sustainability at every level;
- Review, or move to the next level of detail;
- Iteration elements;
- Review on a larger sample.

2.2 Mathematical model of the workflow

As a model of computation, a workflow can be abstracted by a mathematical representation as a graph as a set of pairs actor-connection, where the actor can be: a job, task or step work, and the connection is an arc routed [02]. Computation model on the workflow has the following notation: at $\boldsymbol{w}$ graph is associated with a set of input parameters $\boldsymbol{p}$, a set of input data $\boldsymbol{x}$ and a set of output data $\boldsymbol{y}$. Therefore, we represent the model as a workflow application form

$\boldsymbol{M: W \times P \times X \rightarrow Y}$, where $\boldsymbol{w \in W, p \in P, x \in X}$ and $\boldsymbol{y \in Y}$.

$\boldsymbol{M}$ application is defined as $\boldsymbol{y = M\,(Wp\,(x))}$, i.e. any $\boldsymbol{w}$ workflow of $\boldsymbol{W}$ for proper parameter set to $\boldsymbol{p}$ and input $\boldsymbol{x}$, workflow determines an output $\boldsymbol{y}$. This graph model is found in the Kepler system, a system for creating scientific workflows [11].

2.3 A framework for workflow model

A framework based on a workflow model can support the process management task be it social, economic or scientific. The management of the scientific processes can be assisted by framework which is modelled through a scientific workflow.

Scientific workflows have specific operations research and are planned during the design flow. Execution is conducted at runtime; the user specifies data processing operations flow while dependencies are specified by the workflow designer. Typically, workflows are represented visually by the use of block diagrams, or by specifying them by means of a specific programming language [06].

The researcher utilizes a workflow as a deliverable in his project as a recipe to automate, document, and make possible a scientific process repeatability. Thus the scientific workflow has a life cycle like any other artefact from the project.

The life cycle of scientific workflow route is feasible and has associated phases as: development, implementation and execution of scientific workflows. These phases are largely an endorsement by the workflow systems, systems that have methods and techniques of data mining.

Table 1 shows the key elements underlying the design of a workflow model to a process of analysis based on a relationship of communication.

Table 1. Phases of a design methodology for workflow model.
(adapted from A. Sharp and P. McDermott, *Workflow Modeling)*

1) Establish process context, scope and goals	2) Understand as-is process-workflow and other enables	3)Define to-be process characteristics and requirements
•***Identify related processes*** - identify and link activities - 1:1 links are in same process - draw Overall Process Map •***Clarify target process' scope*** - triggering event, ~5+/- processes, result for each stakeholder, cases/variations •***Clarify as-is process elements*** - functional areas - actors and responsibilities - systems and mechanisms •***Assess as-is process by stakeholder*** (initial) - also specify context and consequences of inaction •***Specify to-process goals*** - subjective and objective •***Specify performance metrics*** - customer-focused outcomes, not internal task efficiency	•***Organize and initiate session*** - staff and management plus external stakeholders - review scope, issues, goals - review ground rules •***Build as-is swimlane diagram*** - one case and path at a time - 1)"Who gets it next?" 2)"How does it get there?" 3)"Who *really* gets it next?" •***Check each step - 5 questions*** 1) again "How does it get there?" 2)"No mushy verbs?" 3)"All triggers shown?" 4)"All participant actors shown?" 5)"All outputs shown?" •***Model other process cases*** - create new diagram, or use original case as a starting point •***Add additional levels of detail*** - only if necessary	•***Assess as-is process by enabler*** (final assessment) - using as-is diagram as a guide - helps us take a holistic view •***Decide on approach*** (abandon, outsource, leave as-is, improve, or redesigne) •***Conduct challenge session*** - challenges hidden assumptions, generates creative ideas - helps us *"think out of the box"* •***Eliminate infeasible ideas*** (cost, legal, resources, impact, …) •***Assess improvement ideas by enabler*** - helps us *avoid unanticipated consequences* - builds requirements document •***Lay out to-be workflow*** - handoff level first, then milestone and task levels

Three categories of tasks are shown in Table 1, which underline building a workflow model: 1) Entry in the context of the research process by clarifying the scope, objectives and performance metrics, 2) Understanding the workflow as the research process the task of organizing, building work steps, creating charts and other details, 3) Defining the process and requirements for workflow modeling.

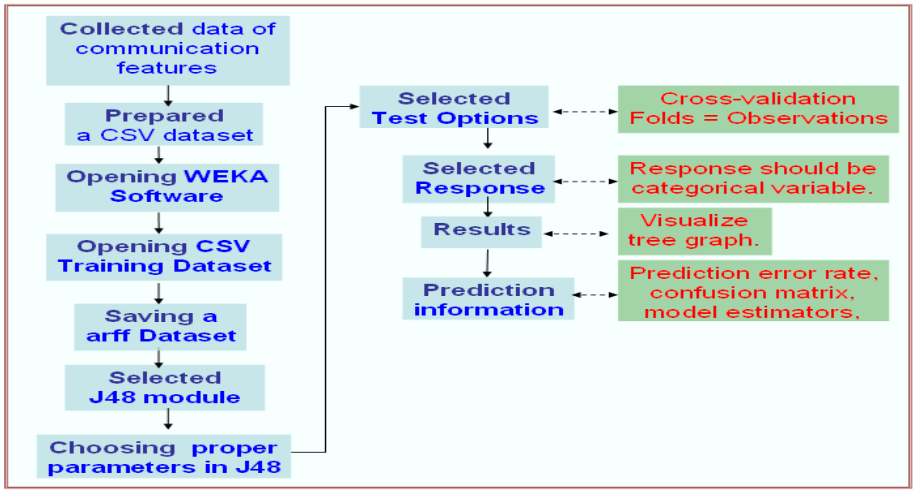


Figure 1: Workflow model for creating and processing a communication relationship.

A scientific workflow can be created with the Weka toolbox. It has even been called "Knowledge Flow" environment. This application has the possibility to create a flow of data processing to get a model to be represented as accurately as possible and help with the simulation of a scientific process. Moreover, it contributes to the implementation of the so-called quantitative management.

A flow pattern is illustrated in Figure 1. This model simulates the processing of instances of a relationship with attributes of a process of communication.

In the model in Figure1 we have followed the design of a workflow designed for processing a communication relationship. That is, we structured and framed the research process, we set semantics of tasks and next we designed the flow model (model that reproduces the process).

3 Modeling technique by Weka

There are several toolboxes that can be used for the creation of workflows as can be seen in summary of Table 2.

Tabel 2: Categories of toolboxes for the management and execution of workflows [03][08].

Category	*Toolbox*	*Synopsis*
Specialized workflow languages	XPDL	A format standardized by the Workflow Management Coalition (WfMC) to interchange business process definitions between different workflow products.
	YAWL	Graphical editor and a worklist handler, that includes an execution engine.
	SCUFL	Dataflow-centric language, defining a graph of data interactions between different services.
Graphical tools	Weka-Kflm	Workbench for creating and processing a data stream
	Weka4WS	Used in data mining systems to manage data and execution flows associated to complex app.
	Taverna	Tool for designing and xecuting workflows
	Pegasus	A set of technologies to execute workflow-based applications in a number of different environments, including desktops, clusters, Grids, and Clouds.
	Kepler	A graphical user interface and a runtime engine that can execute workflows.
	Askalon	An application development and runtime environment, developed for the execution of distributed workflow applications in service-oriented Grids.
	DVega	A scientific workflow engine that adapts itself to the changing availability of resources, minimizing the human intervention.
Textual or XML-based	PMML	A markup language for statistical and data mining models.
	BPEL	A standard executable language for specifying business processes with web services.
	DIS3GNO	A system for defining a service oriented workflow formalism and a visual software environ.
	Karajan	A workflow framework can support hierarchical workflows based on XML.

A good toolbox to achieve workflows for different areas such as the scientific, medical, social, economic is Weka. Weka Workbench has an interface for creating and processing a data stream called KnowledgeFlow. With KnowledgeFlow tool we can create frameworks planned to automate and execute a model or even a Scientific Workflow Life Cycle. The KnowledgeFlow supports Life Cycle phases including design and workflow composition, workflow resource planning, execution workflow execution analysis, visualization and dissemination.

A workflow created in Weka KnowledgeFlow has the following data processing and relationship attributes as input: a layout style of intuitive data flow, processing data in batches or incrementally, processing multiple batches or streams in parallel (each separate stream runs in its own thread) chains of data filters, prospects for models produced by classifiers for each cross-validation, visualization performance incremental classifiers during processing (classification accuracy, RMS error, predictions, etc.) facility to allow easy addition of new components to KnowledgeFlow (plug-ins).

When creating a workflow model, specifically simulate a process that complies with the majority, the next steps (with sub-steps) to build a graph with nodes and connectors: 1) the addition of nodes needed (add new node; add corresponding data source node; assignment class,

and adding CrossValidationFoldmaker node; add the node classifier, adding performance evaluation node, add visualization, etc..); 2) connecting nodes (connecting the two by two and chaining nodes) ; 3) execution of the startup process of the DataSource node "Start loading", 4) viewing the results (if the execution was done correctly, the results will be presented in a separate window).

4 Case Study: Project Communication Management

We present a detailed case study to illustrate the concepts discussed in the previous sections. More concretely we chose a workflow for processing communication data. The data is structured in a relationship with instances of communication.

Each instance contains values motivating communication characteristics. The communication relationship has seven attributes: five attributes are characteristics of communication motivating (satisfaction, trust, openness, listening, encouraging), and two are auxiliary attributes (role communicator class project and communication). Auxiliary attribute called ClasaCom is one with binary values and aims to assign each instance of communication value ("yes" above average characteristics of motivation and "no" otherwise). ClasaCom attribute is important to train classification or clustering methods in the model created with Weka.

We have created a workflow with KnowledgeFlow tool for researcher studying communication projects.

This workflow involves creating a model of analysis and interpretation of communication motivation for the a portfolio of IT projects. Here there are a series of specific problems which are not found in other types of research workflows. In our view, the workflow is a directed graph, ie a flow of execution of several data processing tasks to build a model of motivating communication. On the other hand in the model we execute the workflow in order to deliver results for the interpretation of communication motivation.

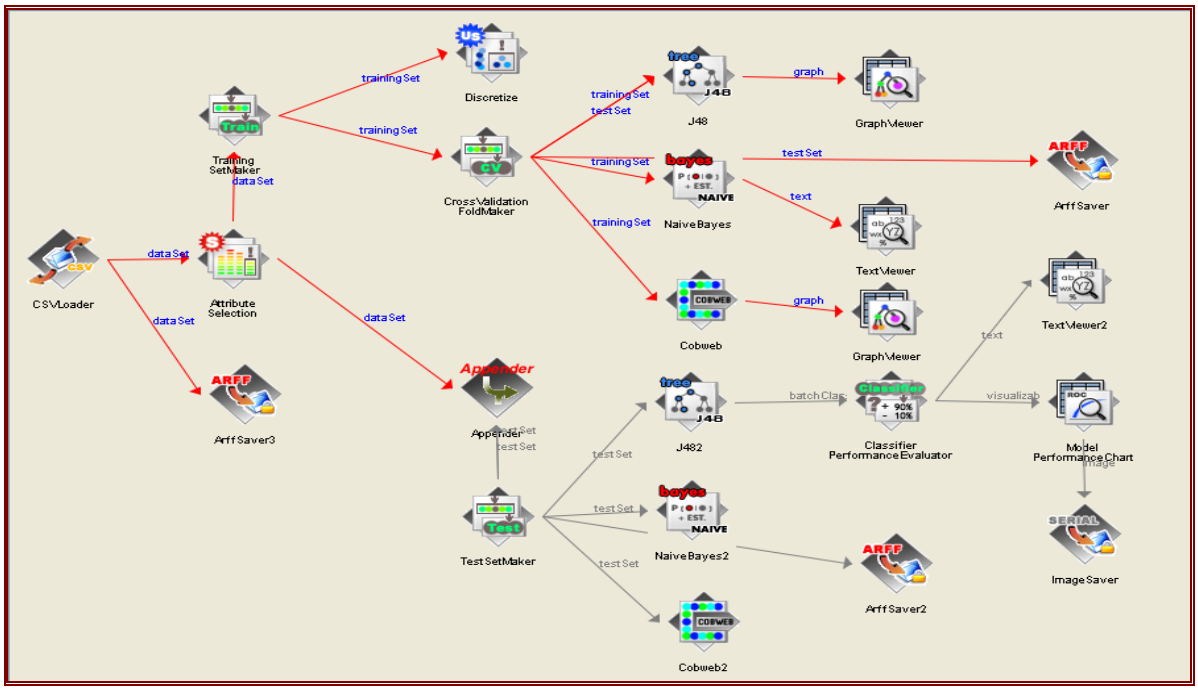


Figure 3: Workflow for data processing of motivating communication.

Resource input workflow is a communication relationship populated with values recorded by interviewing different stakeholders in software development projects. Attributes of a communication instance have characteristic values of motivation, measured on a Likert scale from 1 to 5, values that describe motivating communication management projects at different levels of communication. Task by the stream processing model does not contain complex computations, but the application of statistical techniques lead to new interpretation motivational attributes of communication quality in IT projects. Also, the flow of execution classifications / clustering and filtering leads to results that can be viewed through textual tables, graphs and diagrams necessary for the qualitative interpretations of motivation communication projects.

During flow model different backups, charts, graphs and data archiving management are made. Such a workflow that focuses on the detailed running Weka system, communication researcher is left to deal with the results of the execution model, the researcher end user workflow.

Such a worflow is illustrated in Figure 3.

Mapping Workflows to Resources – shall be made the start node of the workflow, where the input is a spreadsheet of the type CSV, which node 2 is transformed into a dataset, file type ARFF. Once the data resource has been assigned to the Workflow it is available for any type of connection allowed in Workflow processing. Resource input is a relation that contains instances of communication and motivation attributes values of communication, as described above.

Workflow Execution – can be done in two ways: automatically or bach. Workflow in Figure 3 runs bach researcher monitored. During processing flow runs: attribute selection, filtering data test drive data execution algorithms, running algorithms on test data, etc.

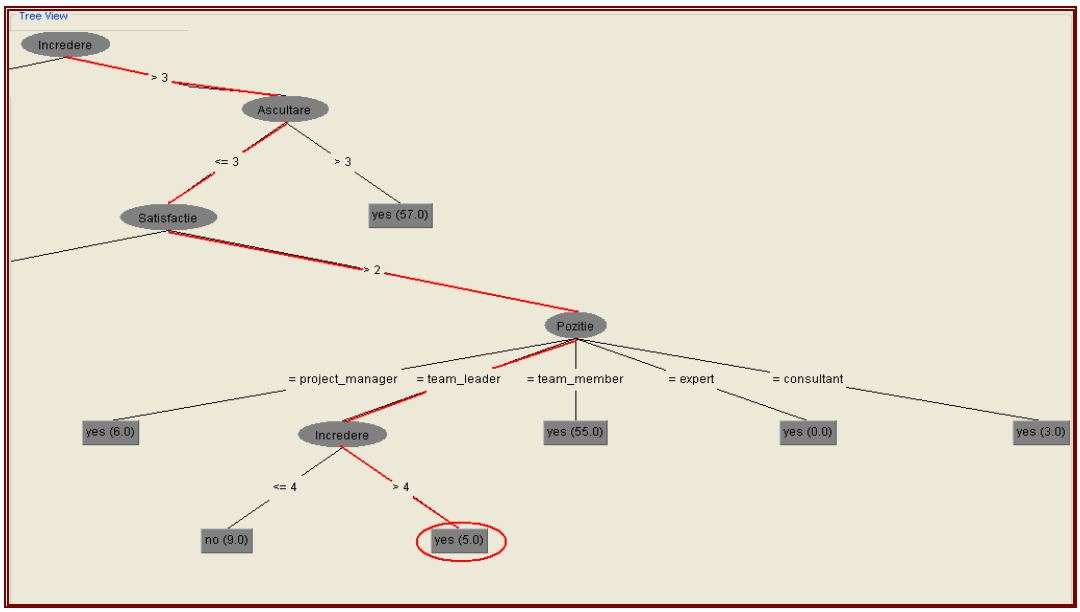


Figure 4: Part of J48 graph wiew with the branchs of the decision tree.

A result from procesing the data, after a clssifying algorythm from the clas DC45 has been applied, is a J48 graph presented in figure 4 and a 3D chart of the motivation characteristics of communication presented in Figure 5. In the graph from Figure 4 a trail is shown which suggests a decision rule for obtaining optimal motivation from communication in projects. In the chart from Figure 5, which is represented in 3D, we have the distribution and the characteristics of motivation in the communication process by satisfaction, trust, an opening.

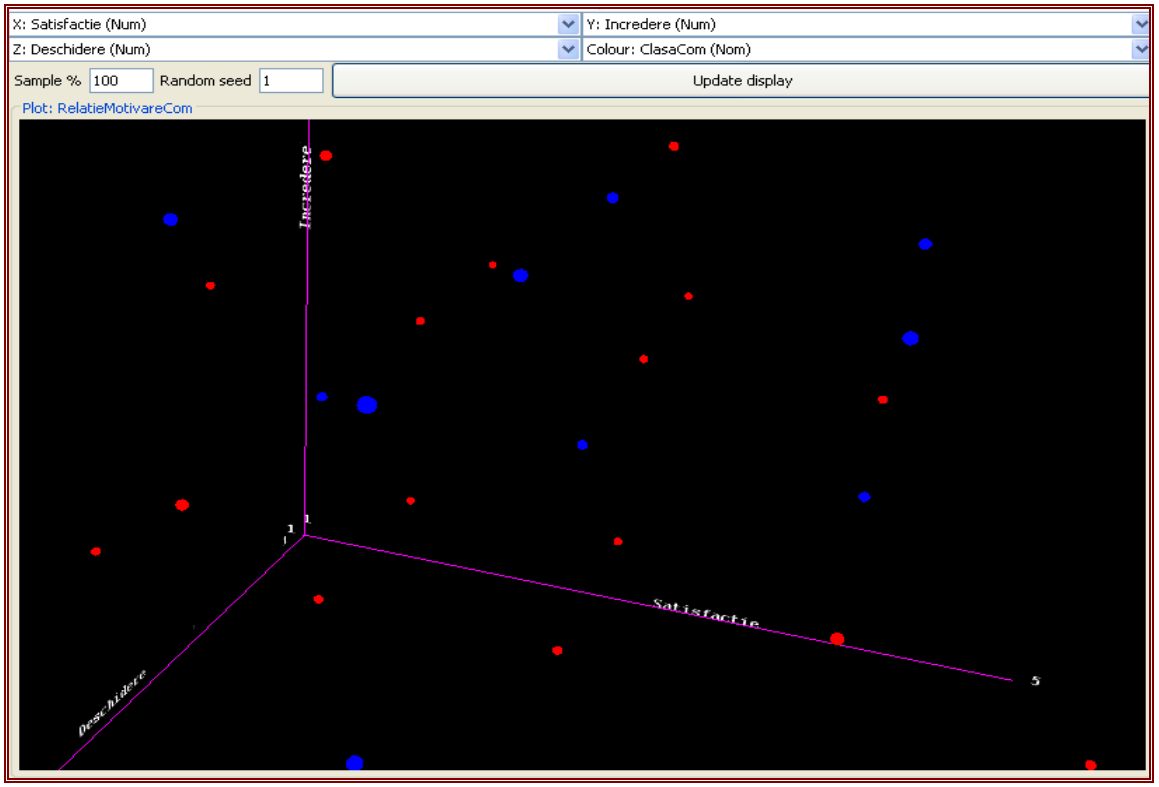


Figure 5: A 3D reprsentation of the communication characteristics of motivation.

Workflow Reuse – this workflow is saved as a .kfml file and an be used later on other data that is being collected. Also results of the workflow execution can be saved which in turn can be used for interpretations, comparisons and analysis.

5 Conclusions

Since the area of research is widening and research time spent is becoming more precious the working mechanism proposed in this article is a good design for more efficient and faster processing research data.

Methodological considerations suggested us to create efficient workflows is a framework for building workflows. Workflows automatize iterative tasks of data processing, so that researchers can better focus on the management experiment, and in this way they can more effectively manage the research process.

Workflows created with Weka are easy ways to process data using statistical techniques and methods of machine learnig. In developing a workflow there are various posiblities of creating important views of the results obtained after the data processing takes places.

Doing research work supported by workflows, encourage reuse of future results prelurare by automating a process, both within a project and a portfolio of projects.

The workflow environment presented can be extrapolated and used in distributed process research data, in grid computing and in web service orietd. Moreover, Weka has a working version Weka4WS which can design the workflow-oriented Web services (Web Service-Oriented).

References

[1] I. H. Witten, E. Frank, M. A. Hall, *DataMining: Practical Machine Learning Tools and Techniques*, Elsevier, Burlington, USA, 2011.
[2] B. Ludascher et al, Scientific Process Automation and Workflow Management, in *Scientific Data Management: Challenges, Existing Technology, and Deployment*, Computational Science Series, chapter 13. Chapman & Hall/CRC, 2009.
[3] D. Talia, Workflow Systems for Science: Concepts and Tools, *ISRN Software Engineering*, vol. 2013, Article ID 404525, 15 pages, 2013. doi:10.1155/2013/404525.
[4] A. Sharp, P. McDermott, *Workflow Modeling: Tools for Process Improvement and Applications Development*, Second Edition, Artech House, Norwood, MA 02062, 2009.
[5] B. Ludascher, et al., Scientific Workflow Management and the Kepler System, in *Concurrency and Computation: Practice & Experience*, 18(10):1039–1065, 2006.
[6] R. Prodan, T. Fahringer, *Grid Computing: Experiment Management, Tool Integration, and ScientificWorkflows*, Springer-Verlag, Berlin Heidelberg, 2007.
[7] P. Sharma, A. Rajavat, Analysis and Design of Service-Oriented Framework for Executing Data Mining Services on Grids, *International Journal of Advanced Research in Computer Science and Software Engineering* 3(3), March - 2013, pp.398-402.
[8] http://en.wikipedia.org/wiki/, *workflow system webpages* accessed at jan. 2013.
[9] WekaDoc, http://weka.sourceforge.net/*wekadoc*/ , accessed at mar. 2013.
[10] http://weka.wikispaces.com/*Exporting+Charts+from+the+Knowledge+Flow*, accessed at jun. 2013.
[11] http://www.cs.waikato.ac.nz/~fracpete/downloads/#kepler, accessed at aug. 2013.
[12] W. Tan, M. Zhou, *Business and Scientific Workflows: A Web Service-Oriented Approach*, published by JohnWiley & Sons, Inc., Hoboken, New Jersey, 2013.
[13]F. Fernandez et al., Assisting Data Mining through Automated Planning, *in MLDM 2009, LNAI 5632*, pp. 760–774, Springer-Verlag Berlin Heidelberg, 2009.

Alexandra-Mihaela Pop
University "Lucian Blaga" of Sibiu
Faculty of Engineering
Department of Industrial Engineering and Management
Str. Emil Cioran, Nr.4, Sibiu, 550025
ROMANIA
E-mail: alexandrapop_6@yahoo.com

Ioan POP
University "Lucian Blaga" of Sibiu
Faculty of Sciences
Department of Mathematics and Informatics
Str.Dr.I.Ratiu, Nr.5-7, Sibiu, 550012, ROMANIA
E-mail: ioan.pop@ulbsibiu.ro

Third International Conference
Modelling and Development of Intelligent Systems
October 10 - 12, 2013
"Lucian Blaga" University
Sibiu - Romania

Towards a Unified Similarity Measure for Node Profiles in a Social Network

Ahmad Rawashdeh and Anca Ralescu

Abstract

Assessing the similarity between node profiles in a social network is an important tool in its analysis. Several approaches exist to study profile similarity, including semantic approaches and natural language processing. However, to date there is no research combining these aspects into a unified measure of profile similarity. Traditionally, semantic similarity is assessed using keywords, that is, formatted text information, with no natural language processing component. This study proposes an alternative approach, whereby the similarity assessment based on keywords is applied to the output of natural language processing of profiles. A *unified similarity measure* results from this approach. The approach is illustrated on a real data set extracted from Facebook.

1 Introduction

Social networks allow people to connect and share their personal details. Many social networking websites have been created and they vary in the services which they provide. Mainly, they allow users to comment and post pictures or video and share.

Facebook is a social networking website that has over one billion users. It allows the user to connect to friends, create personal profiles by specifying their interest –TV, movies, sports, and books – and by posting images and videos of their activities. The website also allows anyone to create pages for their business or favorite personality. Users can even create pages for special interest groups which are open on a restricted basis to group members.[6]

People tend to form relationships with people who are similar to them. Alternatively, it can be said that if a relationship is formed between two people, then there must be some similarity between them. Indeed, it has been found that 80% of social network users form relationships with the contact of their friends [3].

Analysis of similarity between Facebook profiles can be assessed from the study of keyword similarity [3]. To find the relationship between the keywords, these are arranged in a hierarchical structure to form trees of different heights. In the forest model more than one tree is generated for each profile. Related words are retrieved by search in these profile trees, implemented as heuristic search. Semantic relationships between the words can be assessed by using Wordnet. [10]

This study proposes to find the semantic relationship between attribute entries in the social network, not only between keywords. Therefore the category of the words which appear in these entries must be found. This can be accomplished by using a tagger, a program which tags a word by its semantic category. These categories are used to extract the words suitable to assess profile similarity [4]. The (semantic) distance between profiles is very important to this process, as it has been shown that the similarity between profiles deteriorates as the distance between them increases [4].

From this point on, the paper is organized as follows: Section 2 describes the proposed approach for similarity assessment. Section 3 presents the data and the results obtained from applying this approach on a Facebook data-set. The paper closes with a discussion and conclusion section.

2 Finding Similar Profiles

The measure of similarity proposed here combines Wordnet [8] and cosine similarity, which is a very common device to assess document similarity [9].

2.1 Wordnet

Wordnet is a free lexical database that organizes English words into concepts and relations, well-known for assessing semantic similarity. English nouns, verbs, adjectives, and adverbs form hierarchies of *synset* where relations exist that connect them. The relations are Synonymy, Antonymy, Hypernymy, Meronymy, Troponymy, Entailment.

Hypernym of a word

Hypernym of a word conveys its place in a hierarchy of concepts/words and can be retrieved using Wordnet. Consider for example, the two senses of word

”comedy”:

- comedy as a ”humorous drama”
- comedy as ”comic incident”

Taking the first sense, since comedy is kind of drama, drama is a hypernym of comedy. Similarly, since drama is kind of literary work, literary work is a hypernym of drama [5]. The hierarchy determined by the hypernym relationship is a *synset*. Therefore, based on the above, the synset for comedy (with respect to the first meaning) is

Synset 1: [entity] ← [abstract entity] ← [abstraction] ← [communication] ← [expressive style,style] ← [writing style,literary genre,genre] ← [drama] ← [comedy] - light and humorous drama with a happy ending (1)

while the Synset with respect to the second meaning is:

Synset 2: [entity] ← [abstract entity] ← [abstraction] ← [communication] ← [message,content,subject matter,substance] ← [wit, humor, humor, witticism, wittiness] ← [fun, play,sport] ← [drollery, clowning, comedy, funniness] - a comic incident or series of incidents (2)

2.2 Cosine Similarity

Cosine similarity [9] has been successfully used as measure of similarity between documents. A document is described by a vector of fixed dimension of word frequencies. The similarity of two documents is assessed based on the cosine of the angle made between their corresponding vectors. More precisely, given the documents D_i, $i = 1, 2$, with corresponding word vectors v_1 and v_2, the *cosine similarity* between D_1 and D_2 and d2 is defined as

$$CS(D_1, D_2) = \frac{v_1 \cdot v_2}{\|v_1\|\|v_2\|} \tag{3}$$

where $\cdot$ is the dot product between two vectors, and $\|v\|$ denotes the norm of a vector v. Table 1 shows the result of evaluating the cosine similarities between three documents with associated vectors given as

$$\begin{array}{rcl} v_1 & = & (2, 3, 5, 10) \\ v_2 & = & (6, 2, 3, \ 0) \\ v_3 & = & (0, 1, 2, \ 0) \end{array} \tag{4}$$

Table 1: Cosine similarity between the vectors v_1, v_2, and v_2 of (4).

	v_1	v_2	v_3
v_1	1.0000	0.4013	0.4949
v_2	0.4013	1.0000	0.5111
v_3	0.4949	0.5111	1.0000

As it can be seen from this table, the largest cosine similarity is between the 2nd and 3rd document, followed by that between 1st and 3rd document. This corresponds to the first two smallest distances between the vectors v_2 and v_3, and v_1 and v_3 (and will always be so, since the vectors have positive components).

2.3 A Unified Similarity

The approach for the is illustrated on the computation of the similarity between two Facebook profiles. The following steps are performed:

1. Extract the text in the feature field (movies, title) if the data-set is not formatted well.
2. Natural Language Processing: Parse the sentence to obtain its structure.
3. Get the first synset of the word using Wordnet.
4. Encode the word
   - Get all hypernym of the synset of the word.
   - Find the distance from the word to the root of the synset.
5. Each feature field of a profile is encoded as a vector of such distances.
6. Apply cosine similarity between vectors of such distances.

The NLP component in step 2, is used to label (tag) words according to their speech category [7]. The categories used in this study are: NN (noun, proper, singular or mass), NNP (noun, proper, singular), NNS (noun, common, plural), and NNPS (noun, proper, plural) [1]. These part of speech tags are used to assess profile similarity.

The innovative aspect of the current approach is in the encoding of the text input into a vector of distances. This is done as follows: For each profile, the outcome of Step 2 is a collection of word-tag pairs (w, t_w). Given a word-tag pair, (w, t_w), w is considered for inclusion in the similarity evaluation if and only if $t_w \in Tags$, where $Tags = \{NN, NNS, NNP, NNPS\}$ denotes a set of tags of interest. Next, each selected word, w is input to Wordnet which returns the list of hypernyms, in the hierarchical synset representation of w. As illustrated in the example above on the word "comedy" more than one synset can be returned by Wordnet. In this study, only the first sysnset is used for similarity assessment. The encoding of w is the distance to it from the top hypernym ('entity') in the synset. For example, the encoding of the word "comedy" based on the first synset 1 is equal to 7.

If a word has no hypernym (e.g., it is not in Wordnet) then its encoding is 0. This process is summarized as follows. Represent a profile p as a vector of words. That is, $p = [w_1, \ldots, w_k]$ where w_i, $i = 1, \ldots, k$ is a word extracted from the profile by the tagger and k is the number of words extracted.

For each word w_i, use Wordnet to extract its first synset. Define $d_i = d(w_i)$ where, for a given word w,

$$d(w) = \begin{cases} dist(w, \text{[entity]}) & \text{if } w \text{ is in Wordnet} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $dist$ is the distance to [entity], the top hypernym of w in its first synset, output by Wordnet. The encoding of the profile p is a mapping $e : p \mapsto \Re_+^k$ such that

$$e(p) = (d_1, \ldots, d_k)$$

Given two profiles, p, and p' and their corresponding encoding $e(p) = (d_1, \ldots, d_k)$ and $e(p') = (d'_1, \ldots, d'_k)$ the similarity between p and p' is defined as the cosine similarity of $e(p)$ and $e(p')$, as shown in equation (6)

$$Sim(p, p') = CS(e(p), e(p')) \tag{6}$$

where CS is defined as in equation (3).

The process described above converts the problem of similarity assessment between unstructured data into a more rigorously defined problem of similarity between real valued vectors. In principle, it is possible, for a given word w (and hence for a profile), to obtain more than one encoding, by using all the synsets to encode a line of text using several synsets. However, this case is beyond the scope of the current study. Figure 1 illustrates the approach proposed in this study and described above.

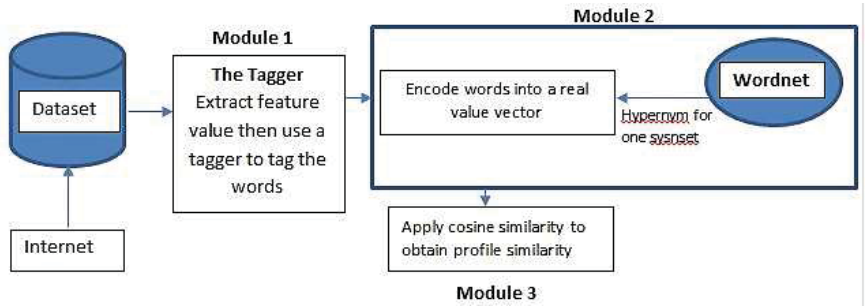


Figure 1: Diagram for computing the unified similarity measure.

The Occurrence Frequency Similarity (OF) of Node Profiles

Let u and x denote two profiles, each having the multiple valued attribute i. The *occurrence frequency* similarity measure, OF, between u and x is defined by equation (7) following the work in [2]. This measure of similarity will be used for comparisons with the measure proposed in this study.

$$OF(i_u, i_x) = \frac{1}{B}\sum_{k=1}^{B}\begin{cases} 1 & \text{if} \quad i_u.n = i_x.n \\ (1 + A \times B)^{-1} & \text{if} \quad i_u.n \neq i_x.n \end{cases} \tag{7}$$

where B is the number of attributes, i_u and i_x are the values of attribute i in the profiles u and x respectively, $i_u.n$ and $i_x.n$ denote the value of the nth subfield for i_u and i_x respectively, N is the total number of item values, and $f(\cdot)$ is the number of records; $A = \log(\frac{N}{1+f(i_u.n)})$, and $B = \log(\frac{N}{f(i_x.k)})$.

3 Experimental Results

The approach described in the preceding section is applied to a Facebook data set as shown next.

3.1 Facebook Profiles Data-set

The Facebook data-set considered in experiments contains 2013 profile pages from Facebook (raw data before the introduction of the Facebook time-line). Skull security has a list of publicly available Facebook URLs which is used to download this data-set that consists of 2013 profiles [2]. More specifically, *Data-set.txt* (Facebook Data-set) contains all the movies interest for different Facebook profile numbers. The format of the data-set is as follows: *Profile_id* followed by the Movies interest entered by the user identified by the *Profile_id*. Furthermore, various characteristics are extracted from the Facebook Data-set, as shown in Table 2. Figure 2 shows the frequency of the top 20 movies in the

Table 2: Characteristics of the Facebook profile data.

Number of Facebook profiles	2013
Average movies entries per profile	2.9
Number of movies entries for all profiles	1744
Maximum movies entries	8
Most Common Genre type [1]	which is the genre type ”unknown”
Minimum movies entries	0
Different movies count	1089

Facebook data-set.

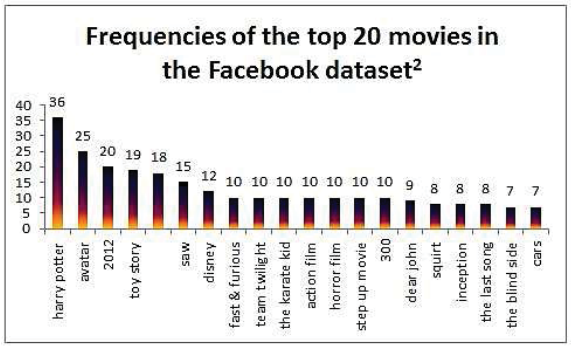


Figure 2: Frequency of the top 20 movies from the Facebook data-set.

Table 3 illustrates the encoding the Movie Attribute for three Facebook pro-

files.

Table 3: Illustration of Movie Attribute of Facebook profiles: their tags and Hypernyms.

Profile 1: Movie Attribute	Harry Potter, Transformers, Mr. & Mrs. Smith						
Words	Harry	Potter	Transformers	Mr.	&	Mrs.	Smith
Tags	NNP	NNP	NNPS	NNP	CC	NNP	NNP
dist to root in synset	0	7	8	8	ignored	8	0
Profile 2: Movie Attribute	Sherina's Adventure						
Words	Sherina	's	Adventure				
Tags	NNP	POS	NNP				
dist to root in synset	0	ignored	8				
Profile 3: Movie Attribute	Love mein Gum, Maqsood Jutt Dog Fighter						
Words	Love	mein	Gum	Maqsood	Jutt	Fog	Fighter
Tags	NNP	NNP	NNP	NNP	NNP	NNP	NNP
dist to root in synset	7	0	7	0	0	6	4

3.2 Results

The algorithm of [4] and the approach described here were implemented in Java. The similarity was calculated between each adjacent nodes' line in the data-set using both the OF measure and Wordnet approach. As we can see from the results, since the Occurence Frequency (OF) depends on whether or not there are redundant data in the data-set. Table 4 illustrates these similarity results for two profiles using OF and Wordnet approaches.

Table 4: OF and Wordnet Similarity of two Facebook profiles along their Movie Attribute.

Data Set	Facebook
Profile-1 ID	100000060663828.html
Movies Interests	Captain Jack Sparrow, Meet The Spartans, Ice Age Movie, Spider-Man
Profile-2 ID	100000067167795.html
Movies Interests	Clash of the Titans, Ratatouille, Independence Day, Mr. Nice Guy, The Lord of the Rings Trilogy (Official Page)
OF Similarity	0.9472
Wordnet based similarity	0.1892

Figure 3 shows the result of applying the OF algorithm to find the similarity and the semantic Wordnet based method for all the node pairs connected by an edge in the data set. Using OF, most of the data are similar, with similarity

value equal to 1. But using Wordnet, the similarity values are distributed over all the data having a peak value at 0.2.

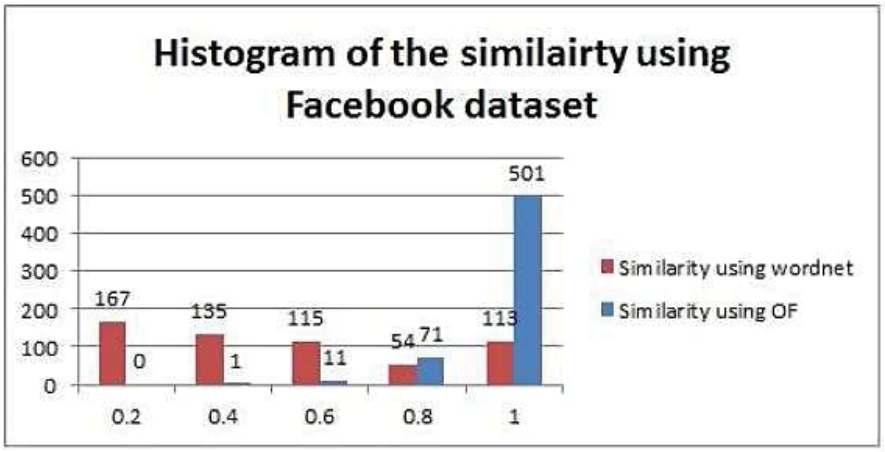


Figure 3: OF and Wordnet similarity results for the Facebook data-set.

4 Conclusions

This study introduces a new approach towards a unified measure of similarity between node profiles, and in general, between pieces of unstructured text. Natural language processing is used to extract speech parts from the texts of interest, and to encode them into vectors with positive components using the distance between the words extracted to the root of a hierarchy of concepts. Similarity is then evaluated between the resultant encoding vectors. While the results seem promising, several issues remain to be discussed and developed in subsequent studies.

References

[1] The university of pennsylvania (penn) treebank tag-set. `http://www.comp.leeds.ac.uk/ccalas/tagsets/upenn.html`. [Online; accessed 1-October-2013].

[2] Cuneyt Gurcan Akcora, Barbara Carminati, and Elena Ferrari. Network and profile based measures for user similarities on social networks. In *Information Reuse and Integration (IRI), 2011 IEEE International Conference on*, pages 292–298. IEEE, 2011.

[3] Prantik Bhattacharyya, Ankush Garg, and Shyhtsun Felix Wu. Analysis of user keyword similarity in online social networks. *Social network analysis and mining*, 1(3):143–158, 2011.

[4] Shyam Boriah, Varun Chandola, and Vipin Kumar. Similarity measures for categorical data: A comparative evaluation. *red*, 30(2):3, 2008.

[5] Ronald Bowes. Return of the Facebook Snatchers. `http://www.skullsecurity.org/blog/2010/return-of-the-facebook-snatchers`, 2010. [Online; accessed 19-July-2012].

[6] Nicole B Ellison et al. Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13(1):210–230, 2007.

[7] The Stanford Natural Language Processing Group. Pos Tagger FAQ. `http://nlp.stanford.edu/software/pos-tagger-faq.shtml`. [Online; accessed 19-July-2012].

[8] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[9] Helen J Peat and Peter Willett. The limitations of term co-occurrence data for query expansion in document retrieval systems. *JASIS*, 42(5):378–383, 1991.

[10] Matt Spear, Xiaoming Lu, Norman S Matloff, and S Felix Wu. Inter-profile similarity (ips): a method for semantic analysis of online social networks. In *Complex Sciences*, pages 320–333. Springer, 2009.

Ahmad Rawashdeh
Machine Learning and Computational Intelligence Lab
Department of Electrical Engineering and Computing Systems
University of Cincinnati, ML 0008
Cincinnati, OH 45221, USA
E-mail: *rawashmy@email.uc.edu*

Anca Ralescu
Machine Learning and Computational Intelligence Lab
Department of Electrical Engineering and Computing Systems
University of Cincinnati, ML 0008
Cincinnati, OH 45221, USA
Anca.Ralescu@uc.edu

Third International Conference
Modelling and Development of Intelligent Systems
October 10 - 12, 2013
"Lucian Blaga" University
Sibiu - Romania

A new approach in E-Commerce applications by using modern technologies for WEB

Livia Sangeorzan, Emanuela Petreanu, Claudia Carstea, Nicoleta Enache David

Abstract

It is a fact that the field of informatics is extremely dynamic. Producers and users of software products are looking solutions to the multitude of problems they are confronted with. Everybody who wants to solve their shopping needs must access an online application and if possible, by just giving a mouse click. This paper presents some new approach, using modern technologies for the web, to resolve their needs as quickly as possible and to anticipate future wish lists for the purchaser. Services Oriented Architecture (SOA) could be such a solution and also a multi agent system.

1. Introduction

The e-commerce sector, in the century of speed and information, must find an answer for everybody who wants to solve their shopping needs as quickly as possible, if possible by just giving a mouse click [4, 5, 8, 10].
The Internet is providing a range of solutions and useful information in all fields, including a large variety of online shops. The internet became an important environment for presenting and marketing of the company products in real time that is available to everyone. The web applications offer people the possibility to buy products for every activity in their life, work or spending their vacations. When you leave in a trip in nature, you might want to buy sport clothes and accessories, and if you are in a hurry and don't have time, a web shop is the best solution. Anyone can access the web application Mountain Expert in order to buy sport products. The application is an e-shop, and has a feature to register and to create a user for connecting to his account, to navigate to a list products, to save products in the shopping card in order to buy them, and finally to send an order.

This site has a simple design and can be accessed by users with a poor internet experience or even people with disabilities.
The web application Mountain Expert has applied rules and techniques of accessibility from (Web Accessibility Initiative) [5, 8].
The server application is implemented using Java JSF and Prime-faces libraries.
Java web technologies has a great flexibility allowing to be used with other technologies and also using these languages /technologies makes possible the obtaining of remarkable results. The written application in Java will be posted on a Web server and will be accessible to any user and

only after authentication you can order an item. Usability is especially important in the case of e-commerce websites. While most usability principles of regular websites still apply for e-commerce sites as well, the different specific pages such as shopping carts, shipping methods, shipping and billing addresses, order reviews, payment options, etc. all add another layer of complexity to creating usable online shops.

2. Theoretical aspects

One of the most frequently used java technology in web applications is the Java Server Faces technology (JSF) [3]. JSF made the development of web application much easier and very interesting. The base elements of this framework are the components, which are small parts from a project that has their own attributes and behaviours. JSF is a relative new technology.

In the process of creating the application has been used for design blueprints: Mocking-Bird, for implementation: Net-Beans framework, for server side: Glassfish with Derby database, UML designs and for accessibility testing: WAVE toolbar [9].

2.1 Technologies

2.1.1 Java Server Faces (JSF)

JSF is the fastest and easiest way of creation the dynamic web applications that are server- and platform –independent [10].

JSF is based on Model View controller architecture, the View is represented by the user interface in our case the facelets.html files, the Model is the data in the application and it is represented by the Entity Classes, and the Controller is represented by the Java Managed Beans and controls the retrieving of data and also resolves user request

2.1.2. Prime-faces

Prime-faces components/modules are similar to JSF components. Prime-faces provide some extra advantages, like better appearance, dynamic actions with built in Java-script or complex actions that don't require any work for the developer aside to adapt component to the application [12].

Prime-faces modules are easy to implement and use and they bring dynamic content and appearance. Some of the Prime-faces modules used in Mountain Expert are: Accordion Panel, Capcha Module, Wizard, Mega Menu, Carousel, Password Input Box, Login module, Data table and Upload Image module.

Prime-faces offer examples for these modules and more on the demo page [12].

2.1.3 Netbeans Framework

Net-beans IDE offers support for developers for Java EE (Enterprise Edition) applications, „which typically run on "big iron" servers and can support thousands of concurrent users" [2]

Net Beans is a free open source Integrated Development Environment (IDE) and platform that has been used to develop Mountain Expert. With the help of this powerful environment, using his tools was much easier to develop the Mountain Expert website.

Net Beans framework has everything one needs to start developing JSF applications.

The application is a Java EE 6 Web type, using Glassfish server.

2.1.4. Database

A database is a collection of data arranged for ease and speed the search and retrieval (American Heritage Dictionary of the English Language).

It is a difference between a database and a database management system (DBMS). A DBMS is a special program for storing and retrieving data, such as Microsoft Access, witch requires more training than using a spreadsheet or word processor.

SQL is a database computer language designed for the retrieval and management of data in relational database management systems (RDBMS), database schema creation and modification, and database object access control management. Many database products support SQL with proprietary extensions to the standard language. The core of SQL is formed by a command language that allows the retrieval, insertion, updating, and deletion of data, and performing management and administrative functions. SQL also includes a Call Level Interface (SQL/CLI) for accessing and managing data and databases remotely. My-SQL is a relational database management system (RDBMS) which has more than 11 million installations. The program runs as a server providing multi-user's access to a number of databases [6]. For the application Mountain Expert it was used Apache Derby database on port 1527. Apache Derby is a relational database that is used in Java projects because it is implemented in Java. His advantages are: small size, is respecting SQL and Java standards and is very simple to use.

2.2. Design and Implementation of *Mountain Expert* site

The website *Mountain Expert* can be accessed by different people for the same reason: to acquire mountains clothes and other accessories to use for their trips in the nature. Figure 1 presents the structure of the website and the relationships between modules.

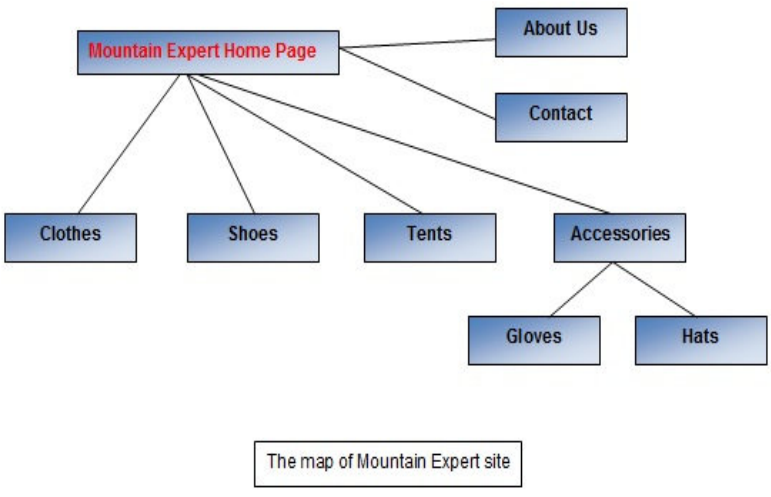


Fig.1 The map of *Mountain Expert* site

2.2.1 Design rules

For a good structure and an easy navigation, the application is respecting a part of usability standard rules of Jakob Nielsen [11]:

- Aesthetic and minimalist design;
- Help and documentation;
- Flexibility and efficiency of use;
- In order to be used by impaired people the applications is respecting the accessibility standard created by W3C called WAI - web accessibility initiative.

The colours and simple design of web pages are respecting the ergonomic standards that help users with eye problems to navigate in the application.

Using templates for all pages
For the web interface and styling in this application, created in JSF, we use Facelets tags to provide a standard web interface layout. For the design of the pages *Facelets templates* and *Faceletes template client* were used. The template is declaring parts of composition items by using tags like <ui:insert>, and the client is defining the composition using tags like <ui:composition> and <ui:define>. This way the modification of the appearance of the application is less time consuming and easier to do. It also eliminates redundant code and the flux of actions become more fluid [1].

Choosing the colours
One have chose simple colours for the design of the website because when using a e-commerce website is important that the attention is focused on the content rather on design elements that pop or move around. To make of comparison between an e-shop and real life, when one walk into a store with very loud music, it's very hard to focus on what you had in mind to shop from there, because is hard to concentrate and take any actions, if there is a disturbing element in the environment.

Consistency and placement of the content
The design of the content is kept consistent in every page, the user is being able to quickly learn how to use the website and this results in a more usable application. People are expecting to find a certain item in certain places and the site *Mountain Expert* is build to respect that rule placing:

- The logo in the left upper corner;
- The navigation bars in the middle of the page;
- Shopping cart and login module in the upper right corner.

Pagination vs. scrolling navigation
"It's the structure of a site that determines its success, a well organized site will lead users effortlessly toward their goals" [1]

Main menu is simple and we have avoided using a drop down menu because it's covering the content of the page.

The local menu is available on every page of the products and it is called breadcrumb menu that it's a list of links in order to visually see the level hierarchy from first page.

This website **is structured by category of products** like clothes, shoes, tents and accessories. Category based websites are the most common type of sites especially if is about an e-commerce application divided into the category of the products they are selling.

2.2.2 UML Diagrams

Use case diagram (Figure 2) - shows a scenario for two types of users, the administrator and a normal user. It describes the main activities performed by the users on the website.

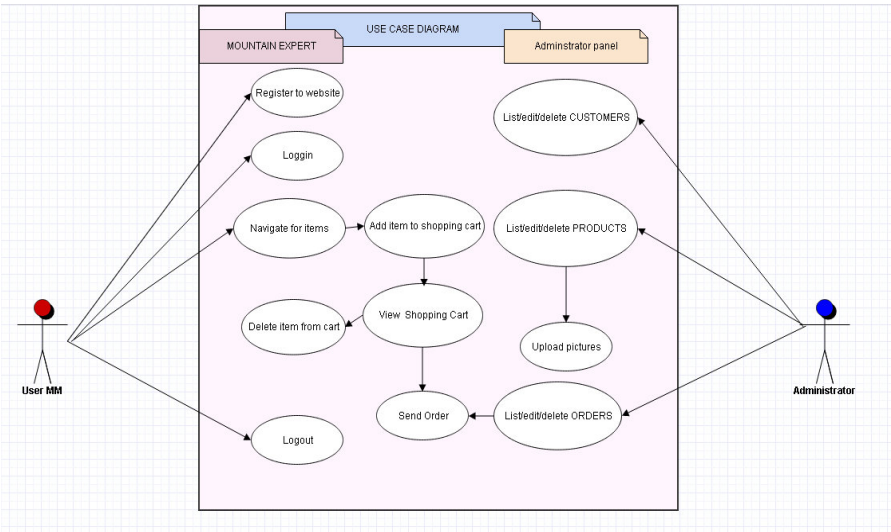


Fig.2. The Use Case Diagram

In Figure 3 one can see the home page of Mountain Expert website build with JavaServer FacesTechnology (JSF) taking into consideration the issues outlined above.

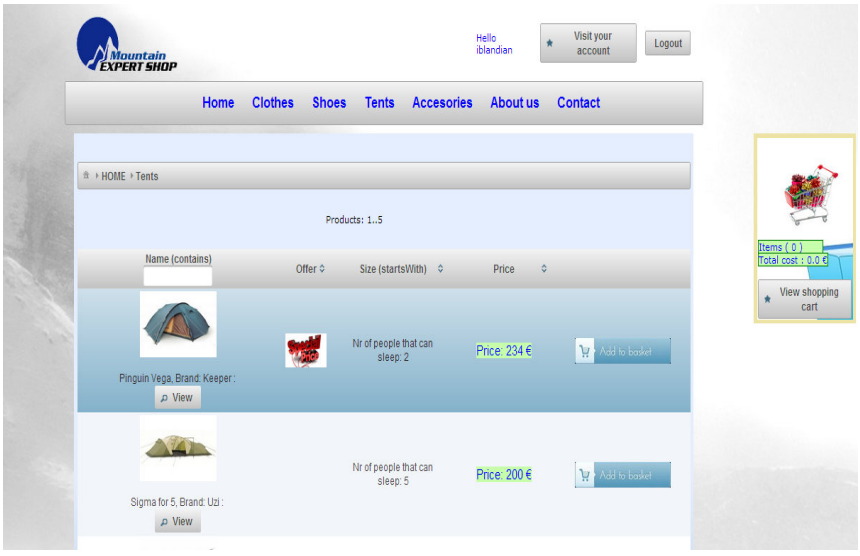


Fig.3 Home page of Mountain Expert website build with JavaServer FacesTechnology

3. Conclusion

An online shop is the physical analogy of buying your products or services from the real shops. Nowadays people don't have enough time to spend going shopping or want to have the power of searching the best product with little effort, so that is why everything went online now.

On line application can be developed using PHP, HTML and CSS3 technologies and also with Java

technologies.

Java technologies are changing rapidly from JSP to JSF .The complexity of JSF is outstanding.

But even if Java is more complex, has more power in more aspects than PHP.

If you're building a very large application that is complex and need to be scalable then Java is the best choice for development but for small applications PHP might be a better choice, being simple and easy to learn and use.

An important advantage of JSF is the component libraries like ICE, Richfaces or Primefaces. JSF can be use rather for enterprise applications, very large scale and complex ones.

An important advantage of PHP is that PHP has a big community and one can find a lot of resources about it. It is easier to use and better documented. Web hosting your website: solutions to host PHP applications are available on a bigger scale.

Mountain Expert is a website that has the purpose to present the products to the client and let you choose whatever you would like to buy and send your order and a new approach in creating a powerful e-commerce website. Being an e-shop, it has a feature to register and to create a user for connecting to your account, to navigate to a list products, to save products in the shopping card in order to buy them, to send an order. The equipments shown are for people who are experts in mountain climbing or surfing. But this site is intended also to those who are lovers of nature and they want to become an expert in the field and they choose performing equipment.

E-commerce and e-business has grown faster than all the predictions and will continue to grow. It is beginning to be part of business and is expected. Some applications, like bill paying over the internet have been successful beyond anyone's imagination. This increase in the online e-commerce market will generate new job and business opportunities for people having the skills to market on-line and for developers of applications on-line. Hence, the search engines PPC programs, SEO Services, social media co-ordination, secure on-line payments, handling the integration of on-line accounts are some of the services which surely be in demand to support this e-commerce trend in the coming years.

References

[1] June Cohen, *The Unusually Useful Web Book*, New Riders , ISBN-10: 0735712069, ISBN 13:978-0735712065, 2003

[2] David R. Heffelfinger, *Java EE 6 Development with NetBeans 7*, Packt Publishing, ISBN 978-1-84951-270-1, 2011

[3] Referinta JSF noua

[4] Ian Hlavats *JSF 1.2 Components*, Packt Publishing, ISBN 978-1-847197-62-7, 2009

[5] Damiano Distante, *Model Driven Development of Web Application with UWA, MVC and* JavaServer *Faces,* Proceeding ICWE'07 Proceedings of the 7th international conference on Webengineering, pages 457-472

[6] http://db.apache.org/derby/

[7] http://www.w3.org/standards/webdesign/accessibility

[8] http://wave.webaim.org/toolbar/

[9] Dana Nourie, *Java Technologies for web application,* http://www.oracle.com/ technetwork/ articles /javase/webapps-1-138794.html

[10] http://www.primefaces.org/showcase/ui/home.jsf

[11] http://www.nngroup.com/articles/ten-usability-heuristics/

SANGEORZAN

Transilvania University of Brasov

Department of Mathematics and Computer Science

B-dul Eroilor 29, 500036 Brasov

Romania

E-mail: sangeorzan@unitbv.ro

PETREANU

Transilvania University of Brasov

Department of Mathematics and Computer Science

B-dul Eroilor 29, 500036 Brasov

Romania

E-mail: emanuela.petreanu@gmail.com

CARSTEA

George Baritiu University of Brasov

Department of Mathematics and Informatics

Str.Lunii 6, 500327 Brasov

Romania

E-mail:

claudia.carstea@universitateagbaritiu.ro

ENACHE-DAVID

Transilvania University of Brasov

Department of Mathematics and Computer Science

B-dul Eroilor 29, 500036 Brasov

Romania

E-mail: nicoleta.enache@unitbv.ro

Third International Conference
Modelling and Development of Intelligent Systems
October 10 - 12, 2013
"Lucian Blaga" University
Sibiu - Romania

Knowledge about replenishable resources: the dynamics of unemployment and job creation

Klaus B. Schebesch **Dan S. Deac**

Abstract

Defined in a general way resources are meant to enable the functioning of complex systems like human societies or biological and techno-economical networks. Apart from the primary inputs to such systems like raw materials, energy, raw information and finance there are a series of embedded or derived complex resources like housing, education and jobs which provide pivotal services. The latter may be often characterized by knowledge production and, on a more fundamental level, by time delays in otherwise quasi-continuously acting dynamical environment. We propose to analyze the role of time delays in order to better understand the dynamics of unemployment and job creation. Temporary and permanent jobs are highly context and delay-sensitive replenishable resources. Misunderstanding their dynamics can cause high and long lasting societal costs.

1 Introduction

In biology and in ecology many good examples of the fundamental role of time delays in otherwise continuous processes are found. A classical example is the Mackey-Glass delay equation which dates back to 1977 (see Lichtenberg and Lieberman [7] detailed explanations) and which describes the regeneration of (blood) cells. Being a scalar equation of type $\frac{dx(t)}{dt} = \frac{ax(t-\tau)}{1+[x(t-\tau)]^c} - bx(t)$, with $a, b, c > 0$ and $x([-\tau, 0])$ given, its dynamics is "harmless" for small τ but exhibits chaotic fluctuations for large enough $\tau > 0$ and thus offers an explanation for hard-to-control malfunctioning of biological regeneration processes. Knowledge about "too large delays" is also vital in the context of other replenishable complex resources, as shown by Nikolopoulos and Tzanetis [3] in a paper on the impact on loss of shelter due to catastrophic events like earthquakes. Other intriguing examples are discussed in the models of Misra and Singh ([1], [2]), where the replenishable resource refers to unemployment (job losses) and job creation. As a rule, different models may express some plausible valid facet of a complex empirical process like job loss and job creation. In the sequel we use computational approaches (symbolic, numerical) in order to support the process of domain knowledge extraction.

2 Some dynamic equations of labor and employment

The models of Misra and Singh ([1], [2]) describe the process of labor market fluctuations by a nonlinear dynamical model in four variables related to employment to be described in the sequel. It is based on the aforementioned model of Nikolopoulos and Tzanetis [3] which treats housing replenishment based on

past information, the time delay of collecting reliable information being substantial. Note that when it comes to describe ways of measuring the labor markets, alternative classifications of variable or factors come to mind and they may pose some indeterminacies from the onset. This refers to e.g. what type of people to include into the class of the "unemployed", how to delimit different term structures of unemployment, etc.. In order to facilitate a model-based analysis of the employment process (in a developed economy) one may use a set of variables $x_i(t)$, $i = 1, ..., 4$, defined at each moment in time $t(t \geq 0)$ (this collection of variables may be further refined):

- $x_1(t) \geq 0$, the number of unemployed (including out of labor, part time ?),
- $x_2(t) \geq 0$, the number of persons with temporary employment,
- $x_3(t) \geq 0$, the number of persons with permanent employment, and
- $x_4(t)$, the vacancies or the newly created jobs (these may also be destroyed?)

During time evolution, We assume that a proportion of the unemployed may become permanently employed, and others temporarily employed. Furthermore some of the temporarily employed may become permanently employed. Finally, a part of both, the temporarily and permanently employed may lose their jobs and become unemployed. For simplicity, we also assume that all unemployed can initially cope with the tasks of any job on offer, but in time they are loosing skills owing to attrition. Also, we assume that there should be no barriers to job acceptance imposed by reduced individual mobility. Furthermore, as a crude approximation, a constant rate of growth of unemployment is assumed (owing for instance to the continuous action of labor saving technical progress). Migration rate of unemployed is assumed to be proportional to their number and the total number of vacant jobs which can be created are bounded and constant.

The time evolution of the umber of unemployed $x_1(t)$ is defined to depend on the following: (1) the rate of change of the number of the unemployed which will become permanently employed is proportional to $x_1(t)$ and to the number of permanent jobs $a_2 + x_4(t) - x_3(t)$, where $a_2 > 0$ is the total number of such permanent jobs available, and, (2) allowing for the transition from unemployment to temporary employment, their latter rate of change is proportional to $x_1(t)$ and to the number of temporary jobs available and vacant $a_4 + x_4(t) - x_2(t)$, where $a_4 > 0$ is the total number of temporary jobs available in the system. Hence, we have

$$\begin{aligned} \frac{dx_1(t)}{dt} &= a - a_1 x_1(t)(a_2 + x_4(t) - x_3(t)) - \\ &\quad -a_3 x_1(t)(a_4 + x_4(t) - x_2(t)) - a_5 x_1(t) + a_6 x_2(t) + a_7 x_3(t), \qquad (1) \\ &\quad \text{with initial condition } \; x_1(0) > 0. \end{aligned}$$

The coefficients $a_1, a_3 > 0$ are for scaling the single described effects and $a_5, a_6, a_7 > 0$ stand for migration rate of the unemployed, the transition rate of permanently and temporarily employed into the state of unemployment, respectively.

Turning now to the evolution of the temporarily employed persons, $x_3(t)$, we consider that the rate of change of unemployed into permanently employed will be proportional to $x_1(t)$ and to the number of vacant permanent jobs $a_4 + x_4(t) - x_2(t)$ (with $a_4 > 0$ being the number of vacant jobs permanently available). Again, the rate of transformation from being unemployed into temporarily employed is proportional to $x_1(t)$ and the number of temporary job vacancies $a_2 + x_4(t) - x_3(t)$, where $a_2 > 0$ is the number of total temporary job vacancies. Hence the differential equation for $x_2(t)$ reads:

$$\begin{aligned}\frac{dx_2(t)}{dt} &= a_3x_1(t)(a_4+x_4(t)-x_2(t))- \\ &\quad -a_8x_2(t)(a_2+x_4(t)-x_3(t))-a_9x_2(t)-a_6x_2(t), \\ &\quad \text{with initial condition } \; x_2(0)>0.\end{aligned} \tag{2}$$

Coefficient $a_8 > 0$ is a constant of proportionality and the constant decay rate $a_9 > 0$ describes the exit of temporarily employed persons from the system (due to death, old age, or migration, see a similar argument for $a_{10} > 0$ below).

The rate of change of the number of unemployed which will find a permanent job is proportional to $a_1x_1(t)+a_8x_2(t)$ and the number of vacant job positions for permanent employment is $a_4+x_4(t)-x_3(t)$. Hence, the rate of change of the permanently employed $x_3(t)$ is given by the following differential equation:

$$\begin{aligned}\frac{dx_3(t)}{dt} &= (a_1x_1(t)+a_8x_2(t))(a_4+x_4(t)-x_3(t))+a_{10}x_3(t)-a_7x_3(t), \\ &\quad \text{with initial condition } \; x_3(0)>0.\end{aligned} \tag{3}$$

where the constant decay rate $a_{10} > 0$ describes the exit of permanently employed persons from the system (due to death, old age, or migration). Finally, the time evolution of newly created jobs is proportional to the time-delayed information about the unemployed in existence at $t-\tau$:

$$\begin{aligned}\frac{dx_4(t)}{dt} &= a_{12}x_1(t-\tau)-a_{13}x_4(t), \\ &\quad \text{with } \; x_1(\theta)=V(\theta), \;\; -\tau\le\theta\le 0, \quad \text{and with initial condition } \; x_4(0)>0.\end{aligned} \tag{4}$$

Note that $V : R \to R$ is a differentiable function which has to be supplied by the modeler. The coefficients $a_{12}, a_{13} > 0$ are the rate of new job creation and the decay rate of permanent unemployment, respectively. The latter may be due to insufficient state funding or labor saving technical progress.

3 Equilibrium and bifurcation analysis

In order to analyze the dynamics of a higher dimensional system which escapes intuition some symbolic term reduction and manipulation are in order. The less consuming part (subsection 4) relates to determining the stationary points $0 = f_i(x_{10}, x_{20}, x_{30}, x_{40})$, $i = 1, ..., 4$, which allow for linearizing and simplifying around these points and upon which the qualitative dynamics of the system (stable orbits, oscillations) may be characterized. Determining the fate of the dynamics as a function of the time delay $\tau > 0$ which changes the eigenvalue spectrum of the linearized system is a more complicated symbolic procedure just touched upon in subsection 3.2 and which is based on Normal Form Theory of bifurcation analysis (for a recent account on theory and computational approaches consult Han and Yu [6]).

3.1 A unique equilibrium point in feasible region of the state space

Equilibrium or stationary points of the dynamic system 1–4 are solutions of the following system of algebraic (low order multinomial) equations:

$$\begin{aligned} a - a_1x_1(a_2 + x_4 - x_3) - a_3x_1(a_4 + x_4 - x_2) - a_5x_1 + a_6x_2 + a_7x_3 &= 0, \\ a_3x_1(a_4 + x_4 - x_2) - a_8x_2(a_2 + x_4 - x_3) - a_9x_2 - a_6x_2 &= 0, \\ (a_1x_1 + a_8x_2)(a_4 + x_4 - x_3) + a_{10}x_3 - a_7x_3 &= 0, \\ a_{12}x_1 - a_{13}x_4 &= 0. \end{aligned} \tag{5}$$

Upon adding the equations of 5 we arrive at

$$a - a_5x_1 - a_9x_2 - a_{10}x_3 = 0. \tag{6}$$

From the 4th equation of 5 results

$$x_4 = \frac{a_{12}x_1}{a_{13}}, \tag{7}$$

and from equation 6 results

$$x_3 = \frac{a - a_5x_1 - a_9x_2}{a_{10}}. \tag{8}$$

By replacing x_3 and x_4 in the second and third equations of system 5 leads finally to a reduced system of two equations in two variables:

$$\begin{aligned} f_3(x_1, x_2) &= (a_1x_1 + a_8x_2)(a_2a_{13}a_{10} + a_{12}a_{10}x_1 - a_{13}(a - a_5x_1 - a_9x_2)) - \\ &\quad a_{13}(a_{10} + a_7)(a - a_5x_1 - a_9x_2) = 0, \\ f_4(x_1, x_2) &= a_{10}a_3x_1(a_4a_{13} + a_{12}x_1 - a_{13}x_2) - \\ &\quad a_8x_2(a_2a_{13}a_{10} + a_{12}a_{10}x_1 - a_{13}(a - a_5x_1 - a_9x_2)) - \\ &\quad a_{10}a_{13}(a_6 + a_9)x_2 = 0. \end{aligned} \tag{9}$$

The two-dimensional system of equations 9 admits a positive solution x_{10}, x_{20}, which is depicted in figure 1 as the intersection of the graphs of $f_3(x_1, x_2) = 0$ and $f_4(x_1, x_2) = 0$. For a given set of parameter values

$$\begin{array}{llllll} a = 65000 & a_1 = 0.00004 & a_2 = 25000 & a_3 = 0.0003 & a_4 = 30000 & a_5 = 2 \\ a_6 = 0.0009 & a_7 = 0.0008 & a_8 = 0.0001 & a_9 = 0.9 & a_{10} = 0.9 & a_{12} = 0.9 \quad a_{13} = 0.2 \end{array}$$

we obtain the equilibrium point $(x_{10}, x_{20}) = (5520, 23370)$ and by using these values in 8 and 7 we obtain the third and forth coordinates of the equilibrium point, namely $(x_{30}, x_{40}) = (36585.55, 24840)$. It can be proven that this equilibrium point is unique for positive values of x_1 and x_2. However, this should be regarded as a simple case without claiming genericity for this to happen in most models of labor dynamics (indeed, it is not very difficult to state much simpler, empirically relevant nonlinear models with multiple equilibria, see e.g. Guckenheimer & Holmes [4]).

For obtaining information concerning the nature of the equilibrium point we compute the characteristic equation (eigenvalue equation) of the dynamical system 1–4 linearized at the equilibrium point. Using the (numerically) instantiated parameter values from above this finally results for the special case of $\tau = 0$ in the equation

$$((\lambda+11.97)(\lambda+2.98)(\lambda+3.45)-96.36-18.85\lambda)(\lambda+.2)-11.97+1.68(\lambda+2.98)(\lambda+3.46)+.5\lambda = 0$$

The solutions of this equation or the eigenvalues are (numerical values are rounded for convenience)

$$\lambda_1 ;= ; -13.37, \quad \lambda_2 ;= ; -4.41, \quad \lambda_3 ;= -0.41 - 0.11i, \quad \lambda_4 ;= ; -0.41 + 0.11i$$

The eigenvalues have negative real parts which, in the context of our dynamical system, implies that the equilibrium point (x_{10}, x_{20}, x_{30}, x_{40}) is **asymptotically stable**, i.e. the equilibrium point is attracting any orbit starting in its vicinity.

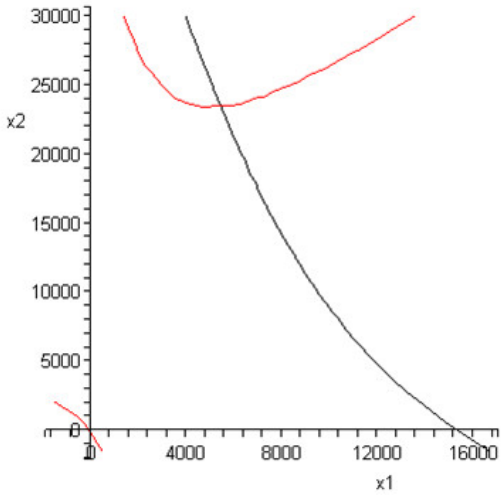


Figure 1: Graphs of the functions $f_3(x_1, x_2) = 0$ (red in colored display style) and $f_4(x_1, x_2) = 0$ (black in colored display style) intersect at a unique point which is the (x_1, x_2)-coordinate of the equilibrium.

3.2 Bifurcation analysis detects change in the dynamics

If, in contradistinction to subsection 4, we allow for delay $\tau > 0$ then the structure of the eigenvalues of a linearized system along the system trajectories may change. In order to capture this by means of detecting a qualitative change in the systems dynamics, a value τ_0 will be determined for which the system undergoes a Hopf bifurcation. This may be achieved by way of a symbolic computation which applies Normal Form Theory from bifurcation analysis (Han and Yu [6]). The procedure of determining such a $\tau_0(a, a_1, ..., a_{13}, .)$ is both potentially complex and tedious often resulting in long symbolic expressions. A Maple program developed in [6] for dealing symbolically with bifurcation analysis may be applied. In doing so a series of intermediate expressions will be generated. With b_{ij} standing for $\dfrac{\partial f_i(x)}{\partial x_j}$ computed at $x = (x_{10}, x_{20}, x_{30}, x_{40})$ we obtain the Jacobian of the dynamics at the stationary point. Formally we also differentiate with regard to $x_1(t - \tau)$, resulting in c_{41}, which in our simple linear case happens to be a_{12}. Hence upon executing in Maple commands

```
> b11:=eval(diff(F1,x1),[x1=x10,x2=x20,x3=x30,x4=x40]);
> ...
> b44:=eval(diff(F4,x4),[x1=x10,x2=x20,x3=x30,x4=x40]);
> c41:=eval(diff(F4,y1),[x1=x10,x2=x20,x3=x30,x4=x40]);
```

we get the Jacobian matrix and c_{41}, where "y1" stands for extra variable $x_1(t - \tau)$, $\tau > 0$. In order to get a Hopf bifurcation point in terms of $\tau > 0$ one executes the further Maple statements:

```
> alpha3:=-(b44+b11+b22+b44);
> alpha2:= b11*(b22+b33)+b22*b33-b12*b13-b12*b21-\
              -b32*b23+b44*(b11+b22+b33);
> alpha1:=-b44*(b11*(b22+b33)+b22*b33-b31*b13-b12*b21-\
              -b32*b23-b11*b22*b33+b12*b21*b33+b32*b23*b11+b31*b13*b22);
```

```
> alpha0:=-b44*(-b11*b22*b33+b12*b21*b33+b32*b23*b11+b31*b13*b22);

> beta2:=-b14;
> beta1:= b14*(b22+b33)-b13*b34-b24*b12;
> beta0:=-b12*b23*b34-b13*b24*b32-b14*b22*b33+b14*b23*b32+\
              +b13*b34*b22+b24*b12*b33;

> gamma6:=alpha3-2*alpha2;
> gamma4:=alpha2^2+2*alpha0-2*alpha1*alpha3-c41^2*beta2^2;
> gamma2:=alpha1^2-2*alpha2*alpha0-c41^2*beta1^2+2*beta0*beta2*c41^2;
> gamma0:=alpha0^2-c41^2*beta0^2;

> eq1:=x^8+gamma6*x^6+gamma4*x^4+gamma2*x^2+gamma0;
> sols1:=solve(eq1,x);

>#Take a positive solution of sols1
> omega:=sols1[2];
```

The resulting value of τ_0 is symbolically expressed as a function of the original model parameters $a, a_1, ..., a_{10}$, and a_{12}, a_{13} as all the Jacobian matrix entries b_{ij} are functions of some of the latter. Translating into a more readable form we finally have:

$$\tau_0 \;=\; \arccos\left(\frac{(\beta_0-\beta_2\omega^2)(\alpha_2\omega^2-\omega^4-\alpha_0)+\beta_1\omega(\alpha_3\omega^3-\alpha_1\omega)}{c_{41}((\beta_0-\beta_2\omega^2)^2+\beta_1^2\omega^2)}\right) \;+\; \text{constant}\times\frac{\pi}{\omega},$$

which is a rather complicated function of the original model parameters. Note that although $c_{41} := a_{12}$ appears explicitly in this function it also affects γ_0, γ_2 and γ_4 in the above Maple expressions. In order to appreciate the potential complexity of equivalent symbolic manipulations applied to possible model extensions, bear in mind that the simplicity of equation 4 also much simplifies the Maple reduction process show here.

For the numerically instantiated parameter values of the model a delay value of $\tau_0 = 20.05768477$ results for locating a bifurcation point. If $0 \leq \tau < \tau_0$, the solution of our dynamical system is asymptotically stable. For $\tau = \tau_0$ the solution exhibits a limit cycle and for $\tau > \tau_0$ the solution becomes unstable. More numerically oriented bifurcation analyzes than the above procedure based on Normal Form Theory may also be performed by using automatic differentiation and trajectory pursuit (Gukenheimer & Meloon [5]). For more details on the computer assisted determination of the equilibrium and bifurcation points of out dynamical system, the reader may contact the authors.

4 Modes of simulation as different views on the process

Numerical simulation is performed for alternative sets of initial conditions by using the model parameter values from subsection . In all case we use the initial function $V(-\tau \leq \theta \leq 0) = x_1(0)$, i.e. we use the initial value at $t = 0$ of the retarded variable. The software used in the sequel is Matlab 12 and Scilab 5.4.1 respectively.

Figure 2 depicts a dynamical process which is asymptotically stable in all four variables, oscillating with diminishing amplitudes.

Figure 3 depicts a dynamical process which is asymptotically stable in all four variables, oscillating with slowly diminishing amplitudes.

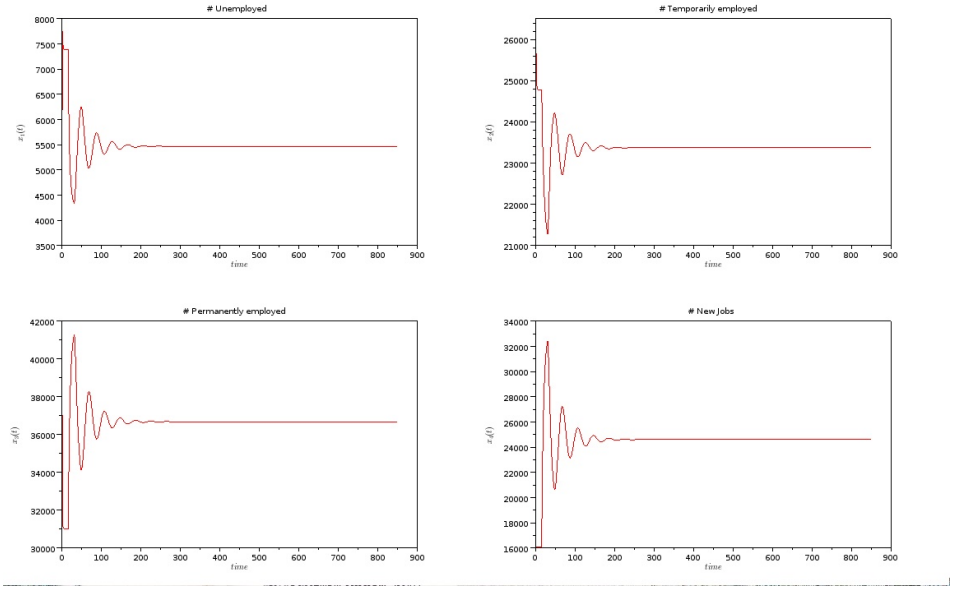


Figure 2: Time evolution of $x_i(t)$, $i = 1, ..., 4$ for the "small" delay $\tau = 15$.

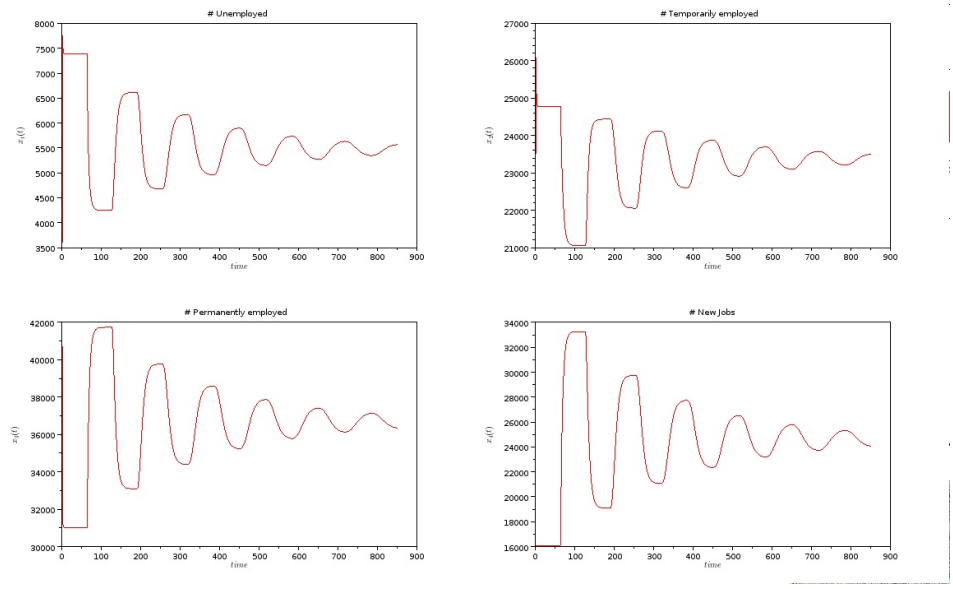


Figure 3: Time evolution of $x_i(t)$, $i = 1, ..., 4$ for the "large" delay $\tau = 63$.

We have also generated a stochastic version in the sense of Ito of the above differential equations by adding noise to the respective increments $\Delta_t x_i(t)$ independently to all $i = 1, ..., 4$ variables. Analyzing the similarity of these stochastic orbits (not shown here) to the empirical time series from labor markets of some developed economies is the aim of present research.

5 Conclusion

Our dynamical model is selected for representing certain aspects of the labor market in an advanced economy. In a preliminary phase, we consider the systemic effects of certain classes of stylized dynamic models which have been used in the literature for capturing a process which can be described as "replenishing a complex resource" in a biological or social context. This allows for viewing unemployment and job creation as being robustly described by a system of four nonlinear differential equations with time delay. As expected, the model shows that unemployment decreases if the number of newly created vacancies increases. Unemployment can be "controled" by by creating new jobs with a rate proportional to the number of unemployed (i.e. the process is not run-away; think of this as a feedback control).

A stochastic model version may be more useful for predictions. Such a model extension was tested and will be reported in future work. The existing model can be further generalized by taking into account some additional variables which describe job creation in the private and in the public sector. An interesting questions to pursue is what adaptations would be necessary if we wish to model a labor market from country with emerging economy, or those from trans-border and otherwise defined economic regions.

Acknowledgement: This paper is partially funded by the research Grant POSDRU/111/4.1/S/91816.

References

[1] A.K. Misra, A.K. Singh. A mathematical model for unemployment. *Nonlinear Analysis: Real World Applications*, 12: 128–136, 2011.

[2] A.K. Misra, A.K. Singh. A delay mathematical model for the control of unemployment. *Differential Equations and Dynamical Systems*, 21,3: 291–307, 2013.

[3] C.V. Nikolopoulos, D.E. Tzanetis. A model for housing allocation of a homeless population due to a natural disaster. *Nonlinear Analysis: Real World Applications*, 4: 561–579, 2003.

[4] J. Guckenheimer, P. Holmes. *Nonlinear Oscillations, Dynamical Systems and Bifurcation of Vector Fields*, Springer-Verlag, 1New York 1983.

[5] J. Guckenheimer, B. Meloon. Computing periodic orbits and their bifurcations with automatic differentiation. *SIAM J. Sci. Stat. Comp.* 22: 951-985, 2000.

[6] M. Han, P. Yu. *Normal Forms, Melnikov Functions and Bifurcations of Limit Cycles*, Applied Mathematical Sciences 181, DOI 10.1007/978-1-4471-2918-9_2, Springer-Verlag, London Limited 2012.

[7] A.J. Lichtenberg, M.A. Lieberman. *Regular and Chaotic Dynamics*, 2nd edition, Springer-Verlag, New York Heidelberg Berlin, 1992.

Klaus Bruno Schebesch
Vasile Goldiş Western University Arad
Department of Informatics
B-dul Revoluţiei Nr.85-86
310025 Arad, ROMANIA
E-mail: *kbschebesch@uvvg.ro*

Dan Stelian Deac
Vasile Goldiş Western University Arad
Faculty of Economics
Str. Mihai Eminescu Nr.15
310086 Arad, ROMANIA
E-mail: *dndeac@gmail.com*

Third International Conference
Modelling and Development of Intelligent Systems
October 10 - 12, 2013
"Lucian Blaga" University
Sibiu - Romania

Using ATL model checking in agent-based applications

Laura Florentina Stoica, Florin Stoica, Florian Mircea Boian

Abstract

Verification of a model executes an exhaustive search of errors in the state-space of the model, in order to verify that this model satisfies the correctness requirements. This search can be accomplished automatically, providing the answer if the verified requirement is satisfied within the model or is appearing a violation. In this paper we present an advanced technique to verify JADE software agents, using Alternating-time Temporal Logic. The proposed solution is based on our original ATL model checker.

1 Introduction

In order to build a well-functioning software system, a thorough investigation of requirements is needed and as a result the requirement specifications are obtained. After the conceptual design phase results an abstract design specification which needs to be validated and checked against the requirement specifications. More specifically, design validation involves checking whether a system design satisfies the system requirements.

Verification of a software system involves checking whether the system follow its design specifications.

Both of these tasks, system verification and design validation can be accomplished thoroughly and reliably using model-based formal methods, such as model checking [1].

Main concern of formal methods in general, and model checking in particular, is helping to design correct systems [2]. The correctness of a software system cannot be proved only by testing. The well-known statement by Dijkstra states that:

"Testing can only show the presence of errors, never their absence".

Model checking is a technology widely used for the automated system verification and represents a technique for verifying that finite state systems satisfy specifications expressed in the language of temporal logics.

Alur et al. introduced Alternating-time Temporal Logic (ATL), a more general variety of temporal logic, suitable for specifying requirements of multi-agent systems [3]. The semantics of ATL is formalized by defining games such that the satisfaction of an ATL formula corresponds to the existence of a winning strategy.

The model checking problem for ATL is to determine whether a given model satisfies a given ATL formula.

ATL defines "cooperation modalities", of the form <<A>> φ, where A is a group of agents. The intended interpretation of the ATL formula <<A>> φ is that the agents A can cooperate to ensure that φ holds (equivalently, that A have a winning strategy for φ) [4].

ATL has been implemented in several tools for the analysis of open systems. In [5] is presented a verification environment called MOCHA for the modular verification of heterogeneous systems.

The input language of MOCHA is a machine readable variant of reactive modules. Reactive modules provide a semantic glue that allows the formal embedding and interaction of components with different characteristics [5].

In [6] is described MCMAS, a symbolic model checker specifically tailored to agent-based specifications and scenarios. MCMAS has been used in a variety of scenarios including web-services, diagnosis, and security. MCMAS takes a dedicated programming language called ISPL (Interpreted Systems Programming Language) as model input language [6].

In [7] is presented a new interactive ATL model checker environment based on algebraic approach. The broad goal of our research was to develop a reliable, easy to maintain, scalable model checker tool to improve applicability of ATL model checking in design of general-purpose computer software.

Our ATL model checker tool is using oriented multi-graphs to represent concurrent game structures over which is interpreted the ATL specification language. The core of our ATL model checker is the ATL compiler which translates a formula *f* of a given ATL model to set of nodes over which formula *f* is satisfied. We found that our ATL model checker tool scale well, and can handle even very large problem sizes efficiently, mainly because it is based on a client/server architecture and take advantage of a high performance database server for implementation of the ATL model checker algorithm.

In this paper we will show how ATL model checking technology can be used for automated verification of multi-agent systems, developed with JADE.

The paper is organized as follows. In section 2 we present the definition of the concurrent game structure, the ATL syntax and the ATL semantics. In section 3 is presented the JADE FSMBehaviour. These concepts are applied in section 4 where ATL Library is used to verify the design of the JADE agents having FSM - driven behaviours. Conclusions are presented in section 5.

2 Alternating-Time Temporal Logic

The ATL logic was designed for specifying requirements of open systems. An open system interacts with its environment and its behaviour depends on the state of the system as well as the behaviour of the environment. In the following we will describe a computational model appropriate to describe compositions of open systems, called concurrent game structure (CGS).

2.1 The concurrent game structure

A *concurrent game structure* is defined as a tuple $S=\langle \Lambda, Q, \Gamma, \gamma, M, d, \delta \rangle$ with the following components:

- a nonempty finite set of all agents $\Lambda = \{1, \ldots, k\}$;
- Q denotes the finite set of *states* ;
- Γ denotes the finite set of *propositions* (or *observables*);
- γ: $Q \rightarrow 2^{\Gamma}$ is called the ***labelling*** (or *observation*) function, defined as follows: for each state $q \in Q$, $\gamma(q) \subseteq \Gamma$ is the set of propositions *true* at q;
- M represents a nonempty finite set of moves;
- the ***alternative moves*** function d: $\Lambda \times Q \rightarrow 2^{M}$ associates for each player $a \in \{1,\ldots,k\}$ and each state $q \in Q$ the set of available moves of agent a at state q. In the following, the set $d(a,q)$ will

be denoted by $d_a(q)$. For each state $q \in Q$, a tuple $\langle j_1,\ldots,j_k\rangle$ such that $j_a \in d_a(q)$ for each player $a \in \Lambda$, represents a *move vector* at q.

- the transition function $\delta(q,<j_1,\ldots,j_k>)$, associates to each state $q \in Q$ and each move vector $\langle j_1,\ldots,j_k\rangle$ at q the state that results from state q if every player $a \in \{1,\ldots,k\}$ chooses move j_a.

A *computation* of S is an infinite sequence $\lambda = q_0, q_1,\ldots$ such that q_{i+1} is the successor of q_i, $\forall i \geq 0$. A *q-computation* is a computation starting at state q. For a computation λ and a position $i \geq 0$, we denote by $\lambda[i]$, $\lambda[0,i]$, and $\lambda[i,\infty]$ the i-th state of λ, the finite prefix $q_0, q_1,\ldots,q_i$ of λ, and the infinite suffix $q_i, q_{i+1} \ldots$ of λ, respectively [3].

2.2 Syntax of ATL

The ATL operator $\langle\langle\ \rangle\rangle$ is a path quantifier, parameterized by sets of agents from Λ. The operators $\circ$ ('next'), $\Box$ ('always'), $\Diamond$ ('future') and U ('until') are temporal operators. A formula $\langle\langle \mathcal{A} \rangle\rangle \varphi$ expresses that the team $\mathcal{A}$ has a collective strategy to enforce φ.

The temporal logic ATL is defined with respect to a finite set of agents Λ and a finite set Γ of propositions. An ATL formula has one of the following forms:

(*1*) p, where $p \in \Gamma$;
(*2*) $\neg \varphi$ or $\varphi_1 \vee \varphi_2$ where φ, φ_1 and φ_2 are ATL formulas;
(*3*) $\langle\langle \mathcal{A} \rangle\rangle \circ \varphi$, $\langle\langle \mathcal{A} \rangle\rangle \Box \varphi$, $\langle\langle \mathcal{A} \rangle\rangle \Diamond \varphi$ or $\langle\langle \mathcal{A} \rangle\rangle \varphi_1 \, U \, \varphi_2$, where $\mathcal{A} \subseteq \Lambda$ is a set of players, and φ, φ_1 and φ_2 are ATL formulas.

Other boolean operators can be defined from $\neg$ and $\vee$ in the usual way. The ATL formula $\langle\langle \mathcal{A} \rangle\rangle \Diamond \varphi$ is equivalent with $\langle\langle \mathcal{A} \rangle\rangle$ *true* $U \varphi$.

2.3 Semantics of ATL

Consider a game structure $S=\langle \Lambda, Q, \Gamma, \gamma, M, d, \delta \rangle$ with $\Lambda=\{1,\ldots,k\}$ the set of players. We denote by

$$D_a = \bigcup_{q \in Q} d_a(q) \tag{1}$$

the set of available moves of agent a within the game structure S.

A *strategy* for player $a \in \Lambda$ is a function $f_a: Q^+ \rightarrow D_a$ that maps every nonempty finite state sequence $\lambda = q_0 q_1 \ldots q_n$, $n \geq 0$, to a move of agent a denoted by $f_a(\lambda) \in D_a \subseteq M$. Thus, the strategy f_a determines for every finite prefix λ of a computation a move $f_a(\lambda)$ for player a in the last state of λ.

Given a set $\mathcal{A} \subseteq \{1,\ldots,k\}$ of players, the set of all strategies of agents from $\mathcal{A}$ is denoted by $F_{\mathcal{A}} = \{ f_a \mid a \in \mathcal{A} \}$. The *outcome* of $F_{\mathcal{A}}$ is defined as $out_{F_{\mathcal{A}}} : Q \rightarrow \mathcal{P}(Q^+)$, where $out_{F_{\mathcal{A}}}(q)$ represents *q-computations* that the players from $\mathcal{A}$ are enforcing when they follow the strategies from $F_{\mathcal{A}}$. In the following, for $out_{F_{\mathcal{A}}}(q)$ we will use the notation $out(q, F_{\mathcal{A}})$. A computation $\lambda = q_0, q_1, q_2, \ldots$ is in $out(q, F_{\mathcal{A}})$ if $q_0 = q$ and for all positions $i \geq 0$, there is a move vector $\langle j_1,\ldots,j_k\rangle$ at state q_i such that [3]:

- $j_a = f_a(\lambda[0,i])$ for all players $a \in \mathcal{A}$, and
- $\delta(q_i, j_1, \ldots, j_k) = q_{i+1}$.

For a game structure S, we write $q \models \varphi$ to indicate that the formula φ is satisfied in the state q of the structure S.

For each state q of S, the satisfaction relation $\models$ is defined inductively as follows:

- for $p \in \Gamma$, $q \models p \Leftrightarrow p \in \gamma(q)$
- $q \models \neg\varphi \Leftrightarrow q \not\models \varphi$
- $q \models \varphi_1 \vee \varphi_2 \Leftrightarrow q \models \varphi_1$ or $q \models \varphi_2$
- $q \models \langle\langle \mathcal{A} \rangle\rangle \circ \varphi \Leftrightarrow$ there exists a set $F_{\mathcal{A}}$ of strategies, such that for all computations $\lambda \in out(q, F_{\mathcal{A}})$, we have $\lambda[1] \models \varphi$ (the formula φ is satisfied in the successor of q within computation λ).
- $q \models \langle\langle \mathcal{A} \rangle\rangle \Box \varphi \Leftrightarrow$ there exists a set $F_{\mathcal{A}}$ of strategies, such that for all computations $\lambda \in out(q, F_{\mathcal{A}})$, and all positions $i \geq 0$, we have $\lambda[i] \models \varphi$ (the formula φ is satisfied in all states of computation λ).
- $q \models \langle\langle \mathcal{A} \rangle\rangle \varphi_1 \, U \, \varphi_2 \Leftrightarrow$ there exists a set $F_{\mathcal{A}}$ of strategies, such that for all computations $\lambda \in out(q, F_{\mathcal{A}})$, there exists a position $i \geq 0$ such that $\lambda[i] \models \varphi_2$ and for all positions $0 \leq j < i$, we have $\lambda[j] \models \varphi_1$.
- $q \models \langle\langle \mathcal{A} \rangle\rangle \Diamond \varphi$ there exists a set $F_{\mathcal{A}}$ of strategies, such that for all computations $\lambda \in out(q, F_{\mathcal{A}})$, there exists a position $i \geq 0$ such that $\lambda[i] \models \varphi$.

3 Automated verification of an agent-based system

Our ATL model checker tool contains the following packages:

- ATL Compiler – the core of our tool, embedded into a Web Service (ATL Web Service);
- ATL Designer – the GUI client application used for interactive construction of the ATL models as directed multi-graphs;
- ATL Library – used for development of custom applications with large ATL models. Versions of this library are provided for C# and Java.

The software can be downloaded from http://use-it.ro (binaries and examples of use):

Download			Description
ATL	**ATL Designer**		Tool for an interactive design of the ATL models
	ATL Web Service		The Web service used for checking the ATL models
	C#	**ATL Library**	The C# Class Library (Visual Studio 2010)
		Example of use	Example of use (Visual Studio 2010 Project)
		Source code	C# source code for testing the ATL Library
		Zip Library [1])	Class library for compression in zip format of ATL models
	Java	**ATL Library**	The Java Class Library (JDK version 7)
		Example of use	Example of use (NetBeans 7.2 Project)
		Source code	Java source code for testing the ATL Library
		GraphStream Library [2])	Packages used for internal representation of the ATL models

Fig. 1 Download page of the site hosting our ATL model checker tool

In the following we will show how our tool can be used for applying the ATL technology in the field of agent-based applications.

Because ATL includes notions of agents, their abilities and strategies (conditional plans) explicitly in its models, ATL is appropriate for planning, especially in multi-agent systems [8].

Automated verification of a multi-agent system by ATL model checking is the formal process through which a given specification expressed by an ATL formula and representing a desired behavioural property is verified to hold for the ATL model of that system.

In the following, ATL Library will be used to detect errors in the design, specification and implementation of an agent developed in JADE.

The ATL Library (java version) will be used to validate the design of JADE agents having FSM-behaviours, in other words, to see that no incorrect scenarios arise as a consequence of a bad design.

3.1 Jade agents with FSM behaviours

The JADE platform is a middleware that facilitates the development of multi-agent systems and applications conforming to FIPA standards for intelligent agents [10].

The Agent class represents a common base class for user defined agents. Therefore, from the programmer's point of view, a JADE agent is simply an instance of a user defined Java class that extends the base Agent class.

A behaviour represents a task that an agent can carry out and is implemented as an object of a class that extends jade.core.behaviours.Behaviour. In order to make an agent execute the task implemented by a behaviour object it is sufficient to add the behaviour to the agent by means of the addBehaviour() method of the Agent class. Each class extending Behaviour must implement the action() method, that actually defines the operations to be performed when the behaviour is in execution and the done() method (returns a boolean value), that specifies whether or not a behaviour has completed and have to be removed from the pool of behaviours an agent is carrying out. Scheduling of behaviours in an agent is not pre-emptive (as for Java threads) but cooperative. This means that when a behaviour is scheduled for execution its action() method is called and runs until it returns. The termination value of a behaviour is returned by his onEnd() method [11].

Agent behaviours can be described as finite state machines, keeping their whole state in their instance variables. When dealing with complex agent behaviours, using explicit state variables can be cumbersome; so JADE also supports a compositional technique to build more complex behaviours out of simpler ones.

The FSMBehaviour is such a subclass that executes its children according to a Finite State Machine (FSM) defined by the user. More in details each child represents the activity to be performed within a state of the FSM and the user can define the transitions between the states of the FSM. When the child corresponding to state S_i completes, its termination value (as returned by the onEnd() method) is used to select the transition to fire and a new state S_j is reached. At next round the child corresponding to S_j will be executed. Some of the children of an FSMBehaviour can be registered as final states. The FSMBehaviour terminates after the completion of one of these children.

The following methods are needed in order to properly define a FSMBehaviour:

- public void **registerFirstState**(Behaviour state, java.lang.String name)
  Is used to register a single Behaviour *state* as the initial state of the FSM with the name *name*.
- public void **registerLastState**(Behaviour state, java.lang.String name)
  Is called to register one or more Behaviours as the final states of the FSM.
- public void **registerState**(Behaviour state, java.lang.String name)
  Register one or more Behaviours as the intermediate states of the FSM.
- public void **registerTransition**(java.lang.String s1, java.lang.String s2, int event)

For the state s1 of the FSM, register the transition to the state s2, fired by terminating event of the state s1 (the value of terminating event is returned by onEnd() method, called when leaving the state s1 - sub-behaviour s1 has completed).

- public void **registerDefaultTransition**(java.lang.String s1, java.lang.String s2)
  This method is useful in order to register a default transition from a state to another state independently on the termination event of the source state.

3.2 Using ATL for verification of a JADE agent

Since testing is based on observing only subset of all possible instances of system behaviour, it can never be complete.

Because testing and simulation can give us only confidence in the implementation of a software system, but cannot prove that all bugs have been found, we will use a formal method, the ATL model checking, for detecting and eliminating bugs in the design of a FSM - driven behaviour of a JADE agent.

Design validation using ATL involves checking whether a system design satisfies the system requirements expressed by ATL formulas.

For a given JADE FSMBehaviour, the ATL model checking is done in two steps:

1. In parallel with construction of the Finite State Machine of the FSMBehaviour, the corresponding ATL model is built.
2. The specification (ATL formula) representing a desired behavioural property is verified to hold for the model describing the FSMBehaviour.

By using our ATL Library [9] to perform ATL model checking, we can detect the states of the model where the ATL formula does not hold and then we can correct the given model or design by reviewing the java code for construction of the Finite State Machine.

4 Using ATL Library for verification of JADE agents

For verification of JADE agents, we need to overwrite the standard methods of JADE FSMBehaviour such that building of the ATL model is accomplished in parallel with the definition of the FSM.

```
ATLGraphModel model = null;
Dictionary dict = null;

public void registerState(Behaviour b, String name){
      if (dict.get(name) == null) {
         lastVertex++;
         model.addNode(lastVertex, name);
         dict.put(name, lastVertex);
      }
      super.registerState(b, name);
}
public void registerLastState(Behaviour b, String name){
      String newName = name + ",*FINAL*";
      lastVertex++;
      model.addNode(lastVertex, newName);
      dict.put(name, lastVertex);
      super.registerLastState(b, name);
 }

public void registerTransition(java.lang.String s1, java.lang.String s2, int event) {
```

```
public void registerTransition(java.lang.String s1, java.lang.String s2, int event) {
      lastEdge++;
      int v1 = Integer.parseInt(dict.get(s1).toString());
      int v2 = Integer.parseInt(dict.get(s2).toString());
      model.addEdge(lastEdge, v1, v2, "<" + event + ">");
      super.registerTransition(s1, s2, event) ;
}
```

Fig. 2 Overwriting the standard methods of the FSMBehaviour

Using the ***checkFSM()*** method, the new class ATL_FSMBehaviour extends the functionality of the standard JADE class FSMBehaviour by adding ATL model checking capability (the ATL formula to be verified is submitted as a parameter):

```
public boolean checkFSM(String ATLFormula){

  //model.setServer(model.SERVER_LOCALHOST);
  model.setServer(model.SERVER_USEIT);

  model.setDebug(false);
  model.setCompressed(true);

  try {
     String result = model.checkModel(xmlModel, ATLFormula, "0");
     System.out.println(result);

     int[] states = model.getStates();

     if (states == null) {
        System.out.println("There are no states in which given ATL formula is satisfied.");
        System.out.println("The FSM is not well defined!");
        return false;
     } else {
        System.out.println("The ATL formula is satisfied in the following states:");
        for (int i = 0; i < states.length; i++) {
           System.out.print(" " + Integer.toString(states[i]));
        }
        if (states.length != lastVertex + 1){
           System.out.println("\nThe FSM is not well defined!");
           return false;
        }
        else
        {
           System.out.println("\nThe FSM is well defined.");
           return true;
        }
     }
  } catch (ATLException ex) {
     System.out.println(ex.getError());
     return false;
  }
}
```

Fig. 3 A new method of FSMBehaviour for verification of an ATL formula

In the following example, the ATL formula checked is:

$$<< A >> @\, d \tag{2}$$

Thus, we verify that the state *d* is a reachable state.

In figure 4 is presented the underlying ATL model of the FSMBehaviour and loaded in ATL Designer:

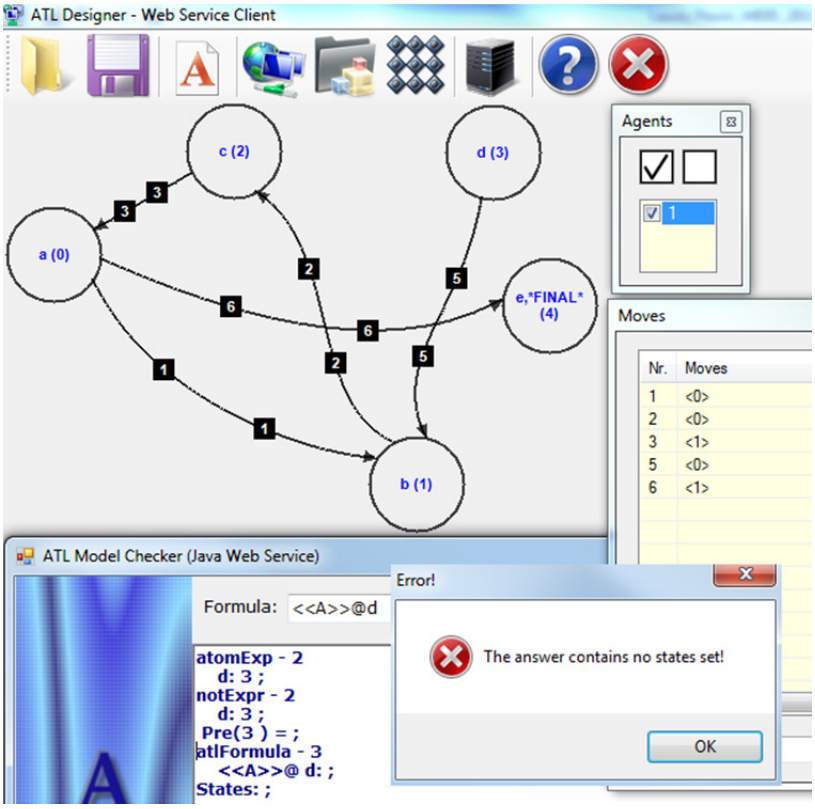


Fig. 4 Checking the ATL model in ATL Designer

As we can see from the figure 5, the desired behavioural property expressed by ATL formula does not hold and in conclusion the state *d* is not reachable.

The FSMBehaviour of the JADE agent must be clearly revised, so the new model to satisfy the specification expressed by ATL formula (2).

```
INFO: ---------------------------------------
Agent container Container-18@Server1 is ready.
---------------------------------------------
States: ;

There are no states in which given ATL formula is satisfied.
The FSM is not well defined!
```

Fig. 5 The ATL formula is not verified in the model

5 Conclusions

In this paper Alternating-time Temporal Logic was used for the automated verification of JADE agents. The proposed method can be applied to any software system (written in Java or C#) for which can be constructed an ATL model using methods of our ATL Library.

The software components (libraries, examples of code, web services, designers) of the ATL model checker tool used in this paper can be downloaded from *http://use-it.ro*.

References

[1] K.Y. Rozier, Survey: Linear Temporal Logic Symbolic Model Checking, *Computer Science Review*, Volume 5 Issue 2, 163-203, 2011
[2] J. Barnat, L. Brim, P. Ročkai, Scalable Shared Memory LTL Model Checking. *International Journal on Software Tools for Technology Transfer (STTT)*, 12(2):139-153, 2010
[3] R. Alur, T. A. Henzinger, O. Kupferman, Alternating-time temporal logic, *Journal of the ACM*, 49(5), pp. 672–713, 2002.
[4] M. Kacprzak, W. Penczek, Fully symbolic Unbounded Model Checking for Alternating-time Temporal Logic, Journal Autonomous Agents and Multi-Agent System, Volume 11 Issue 1, pp. 69 – 89, 2005
[5] R. Alur, T.A. Henzinger, F.Y.C. Mang, S. Qadeer, S.K. Rajamani, S. Tasiran, Mocha: modularity in model checking, in Proc. of CAV 98, volume 1427 of Lect. Notes in Comp. Sci., pp. 521-525. Springer-Verlag, 1998
[6] A. Lomuscio, F. Raimondi, Mcmas: A model checker for multi-agent systems, in Proc. of TACAS 06, volume 3920 of Lect. Notes in Comp. Sci., pp. 450-454, Springer-Verlag, 2006.
[7] L. F. Cacovean, F. Stoica, D. Simian, A New Model Checking Tool, *Proceedings of the 5th European Computing Conference (ECC '11)*, Paris, France, April 28-30, 2011
[8] W. van der Hoek, M. Wooldridge, Tractable multiagent planning for epistemic goals, in *Proceedings of AAMAS-02*, pp. 1167-1174, ACM Press, 2002.
[9] L. F. Cacovean, F. Stoica, WebCheck – ATL/CTL model checker tool, http://use-it.ro
[10] Java Agent Development Framework (JADE), http://jade.tilab.com/
[11] F. Bellifemine, G. Caire, T. Trucco, G. Rimassa, JADE programmer's guide, http://jade.tilab.com, 2013

Laura Florentina STOICA
Faculty of Science,
"Lucian Blaga" University
Department of Mathematics and Informatics
5-7 Dr. Ratiu Street, Sibiu
ROMANIA
E-mail: laura.cacovean@ulbsibiu.ro

Florin STOICA
Faculty of Science,
"Lucian Blaga" University
Department of Mathematics and Informatics
5-7 Dr. Ratiu Street, Sibiu
ROMANIA
E-mail: florin.stoica@ ulbsibiu.ro

Mircea Florian BOIAN
Faculty of Mathematics and Computer Science, "Babes Bolyai" University
Department of Computer Science
1 M. Kogalniceanu Street, Cluj Napoca
ROMANIA
E-mail: florin@cs.ubbcluj.ro

Third International Conference
Modelling and Development of Intelligent Systems
October 10 - 12, 2013
"Lucian Blaga" University
Sibiu - Romania

Algebraic model for the CPU arithmetic unit behaviour

Anca Vasilescu

Abstract

Modern computer systems are regarded as a sum of interconnected and communicating resources. Both the design and the operation of each of these resources, and the global behaviour and performance of the entire computer system are equally important. This approach points to a component-based analysis and development of such systems, each component being able to be specified and verified as a specific agent. Formal methods represent a reliable solution for systematically and exhaustively studying the specific agents involved in describing computer components behaviour, providing the appropriate tools for both the agents' environment modeling and the target agents' properties formal verification.

An algebraic formal framework for modelling the interconnecting processes involved in the agents' description is advanced here using the SCCS process algebra and its corresponding automatic verification benchmark, CWB-NC. In this paper we add a new component model to our formal framework by considering the CPU arithmetic unit. The original approach followed in the present paper consists in developing an SCCS based algebraic model for the arithmetic unit behaviour. The authors' contributions are both the definitions of the SCCS agents for modelling the target behaviour and the proofs for the bisimulation equivalence between those agents. Adding these results to other similar results obtained in our framework, we have important prerequisites in the future work for modelling the behaviour of the entire ALU consisting of arithmetic unit, logic unit and specific control circuits.

1 Introduction

Computer architecture provides a structured and organized view upon the computer system hardware components. With respect to the final users' demands, better solutions for designing and assembling hardware components are investigated. These solutions usually target the increasing system scalability, the components' accurate operation or reducing components' assembling costs.

Modern computer systems are regarded as a sum of interconnected and communicating resources. Both the design and the operation of each of these resources, and the global behaviour and performance of the entire computer system are equally important. This approach points to a component-based analysis and development of such systems, each component being able to be specified and verified as a specific agent.

Formal methods represent a reliable solution for systematically and exhaustively studying the specific agents involved in describing computer components behaviour, providing the appropriate tools for both the agents' environment modeling and the target agents' properties formal verification.

Considering the computer architecture description at the digital logic level, the agent-based approach is applied in this paper to cover both the digital logic circuits design and verification. An algebraic formal framework for modelling the interconnecting processes involved in the agents' description is advanced here using the SCCS process algebra [3] and its corresponding automatic verification benchmark, CWB-NC [22]. Using the operational semantics of the given SCCS algebra, we may evaluate and formal verify how the proposed implementation-based model relates to the intended specification-based definitions of the given components behaviour. As an extra mark for our model correctness, an automatic verification of the target agents' equivalence is applied using the CWB-NC tool.

This formal framework represents our research interest for obtaining an algebraic model for the entire computer operation based on the interconnected hardware components. Our main results have already aimed to a set of hardware components, as follows: counter registers [11], memory component [14], [15], logic part of the processor arithmetic logic unit [19].

In this paper we add a new component model to our formal framework by considering the other main part of the processor ALU, namely the arithmetic unit. The computer's Arithmetic-Logic Unit (ALU) is a Combinational Logic Circuit (CLC), a part of the execution unit as a core component of all Central Processing Units (CPUs) of modern computers. A concrete structure of the ALU is considered in order to achieve the most addressed arithmetic operations. The original approach followed in the present paper consists in developing an SCCS based algebraic model for the arithmetic unit (AU) behaviour. The authors' contributions are both the definitions of the SCCS agents for modelling the AU behaviour and the proofs for the bisimulation equivalence between those agents. Jointly these results and the results from [19] will be important prerequisites in our future work for modelling the behaviour of the entire ALU consisting of arithmetic unit, logic unit and specific control circuits.

2 Preliminaries

This section considerations are following our presentations of the same subjects made in [19].

2.1 Arithmetic Logic Unit

The part of the computer that performs the bulk of data-processing operations is called the central processing unit and is referred to as the CPU for central processing unit [4], [9]. The CPU is made up of three major parts, as follows: control, register set and arithmetic logic unit (ALU). The register set stores intermediate data used during the execution of the instructions. The arithmetic logic unit performs the required microoperations for executing the instructions. The control unit supervises the transfer of information among the registers and instructs the ALU as to which operation to perform.

Instead of heaving individual registers performing the microoperations directly, computer systems employ a number of storage registers connected to a common operational unit called an arithmetic logic unit, abbreviated ALU [4], [9]. To perform a microoperation, the content of specified registers are placed in the inputs of the common ALU. The ALU performs an operation and the result of the operation is then transferred to a destination register. The ALU is a combinational circuit so that the entire register transfer operation from the source registers through the ALU and into the destination register can be performed during one clock pulse period. The shift microoperations are often performed in a separate unit, but sometimes the shift unit is made part of the overall ALU.

For the target of this paper we consider a specific structure of the computer's ALU represented in Figure 1, adapted from [6].

This diagram is divided into three sections: Logic unit, Arithmetic unit and Decoder. The inputs are a, b, S_0 and S_1. The a and b inputs are used as the regular, 1-bit inputs for all operations. The S inputs operate as enable lines since for each of the four possible combinations of S values, only one of

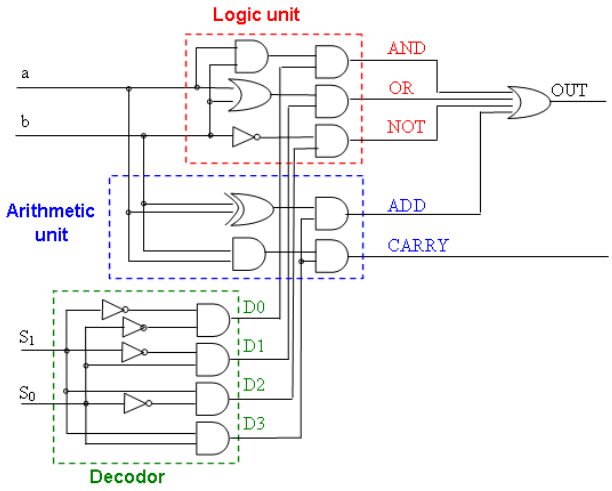


Figure 1: 1-bit UAL

the decoder outputs $D0$, $D1$, $D2$, $D3$ will be "turned on". Thus, the function of the Decoder subpart is to figure out which of the four operations will be done: AND, OR, NOT or ADD. On the right side of the circuit, all of the outputs are ORed together. However, only one of the four inputs of the OR gate could potentially be an 1 due to the enable lines.

For practical operations an 8-bit ALU is more convenient. For this, the previous diagram needs to be repeated 8 times, eventually considering also the specific lines for managing the carry bit.

In the next sections we will consider in details the arithmetic part of this structure as a collection of two boolean operations, $a\,XOR\,b$ and $a\,AND\,b$, we will define an algebraic model for this unit behaviour and we will prove its correctness.

2.2 Process algebra SCCS

The process algebra SCCS, namely *Synchronous Calculus of Communicating Systems* [1] is derived from CCS, especially for achieving the synchronous interaction in the framework of modelling the concurrent communicating processes. Both in CCS and in SCCS, processes are built from a set of atomic actions A. Denoting the set of labels for these actions by Λ, a CCS action is either (1) a *name* or an input on $a \in \Lambda$ denoted by a, (2) a *coname* or an output on $a \in \Lambda$ denoted by $\overline{a}$ or $\tilde{}a$ or (3) an internal on $a \in \Lambda$ denoted by τ. In SCCS the *names* together with the *conames* are called the *particulate actions*, while an *action* $\alpha \in \Lambda^*$ can be expressed uniquely (up to order) as a finite product $a_1^{z_1} a_2^{z_2} ...$ (with $z_i \neq 0$) of powers of names. Note the usual convention that $a^{-n} = \overline{a}^n$ and that the action **1** in SCCS is the action τ from CCS and it is identified in SCCS with the empty product. An SCCS *process* P is defined with the syntax:

P	::= nil	termination
	| α:P	prefixing
	| P+P	external choice
	| P $\times$ P	product, synchronous composition
	| P$\backslash L$	restriction, $L \subseteq A \cup \overline{A}$
	| P$[f]$	relabelling with the morphism $f : A \cup \overline{A} \to A \cup \overline{A}$

In this grammar, the restriction is inherited from CCS. There is also an SCCS specific restriction denoted by the $\restriction$ operator and structural related with the CCS operator by P$\backslash L$ = P $\restriction$ E where E = (A-L)* is the submonoid of A generated by the set difference A-L. By definition, the P $\restriction$ E agent is

forced to execute only the actions from the set E as the external actions and the agent P$\backslash L$ is forced to not execute the actions from the set L, except as the internal actions.

The operational semantics for SCCS is given via inference rules that define the transition available to SCSS processes. Combining the product and the restriction, SCCS calculus defines the synchronous interaction as a multi-way synchronization among processes.

3 The model for the arithmetic unit behaviour

As we have already mentioned in Preliminaries, we consider the arithmetic part of the arithmetic logic unit represented in the previous Figure 1. For the diagrammatic representation of this part we use in the next Figure 2 our own software LCD [13] developed for representing the digital-logic circuits and simulating their behaviour.

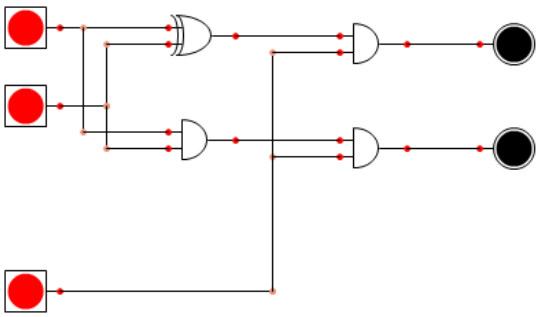


Figure 2: 1-bit UAL - Arithmetic part

3.1 The algebraic model

As the main results of this paper, we define in this section the algebraic model for the AU behaviour based on three kinds of agents: (1) the basic agents - corresponding to the main logic gates AND and XOR, (2) the enabling agents - corresponding to the connection of the AU with the decoder and (3) the arithmetic agents - corresponding to the two AND gates level.

(1) The basic agents are: ANDab and XORab. Their definitions are:

$$\text{ANDab} = \text{AND}[\Phi\text{AND}_{ab}]$$

based on the agent

$$\text{AND} = \sum_{x,y\in\{0,1\}} (in1_x in2_y \overline{out}_z : \text{nil})$$

with the Boolean evaluation $z = x$ AND y and the morphism ΦAND_{ab} defined by the relabelling pairs $in1 \mapsto upa$, $in2 \mapsto upb$ and $out \mapsto ANDabout$;

$$\text{XORab} = \text{OR}[\Phi\text{XOR}_{ab}]$$

based on the agent

$$\text{XOR} = \sum_{x,y\in\{0,1\}} (in1_x in2_y \overline{out}_z : \text{nil})$$

with z = x XOR y and the morphism ΦXOR$_{ab}$ defined by the relabelling pairs $in1 \mapsto downa$, $in2 \mapsto downb$ and $out \mapsto XORabout$.

(2) The enabling agents are: EADD and ECARRY. Their definitions are:

$$\text{EADD} = \text{AND}[\Phi\text{AND}_{EADD}]$$

with ΦAND$_{EADD}$ defined by $in1 \mapsto XORabout$, $in2 \mapsto downD3$ and $out \mapsto ADDout$;

$$\text{ECARRY} = \text{AND}[\Phi\text{AND}_{ECARRY}]$$

with ΦAND$_{ECARRY}$ defined by $in1 \mapsto ANDabout$, $in2 \mapsto upD3$ and $out \mapsto CARRYout$.

(3) The arithmetic agents are:

$$\text{ArithmADD} = (\text{XORab} \times \text{EADD})\backslash\{XORabout\}$$

and

$$\text{ArithmCARRY} = (\text{ANDab} \times \text{ECARRY})\backslash\{ANDabout\}$$

We also need some agents for modelling the distribution of the electric signal on the circuit wires. These agents depend on the number of forked lines in a circuit node. Hence, for the fork of the signal into two lines the agent is

$$\text{NODE2} = \sum_{x\in\{0,1\}} (in_x\overline{up}_x\overline{down}_x : \text{nil}).$$

We need three appropriate rellabeled agents based on the agent NODE2, as follows:

$$\text{NODE2\_a} = \text{NODE2}[\Phi 2_a] \tag{1}$$

with $\Phi 2_a$ defined by $in \mapsto a$, $up \mapsto upa$ and $down \mapsto downa$;

$$\text{NODE2\_b} = \text{NODE2}[\Phi 2_b] \tag{2}$$

with $\Phi 2_b$ defined by $in \mapsto b$, $up \mapsto upb$ and $down \mapsto downb$;

$$\text{NODE2\_D3} = \text{NODE2}[\Phi 2_{D3}] \tag{3}$$

with $\Phi 2_{D3}$ defined by $in \mapsto D3$, $up \mapsto upD3$ and $down \mapsto downD3$.

Using the above agents, we are now ready to define two agents for modelling the AU behaviour: a low-level specification agent EArithm based on the behaviour of the arithmetic unit and a high-level specification agent SpecEArithm based on the definition structure of the arithmetic unit circuit.

Hence, the implementation of the arithmetic part of the ALU based on the behaviour of the circuit is given by the agent:

$$\begin{aligned}\text{EArithm} &= \\ &= (\text{ArithmADD} \times \text{ArithmCARRY} \times \text{NODE2\_a} \times \text{NODE2\_b} \times \text{NODE2\_D3})\backslash \\ &\backslash\text{Comm\_EArithm}\end{aligned} \tag{4}$$

where the set of communicating actions is

$$\text{Comm\_EArithm} = \{upa, downa, upb, downb, upD3, downD3\}.$$

The specification of the arithmetic part of the ALU based on the definition of the circuit represented in Figure 2 is given by the agent:

$$\text{SpecEArithm} = \sum_{x,y,m\in\{0,1\}} (a_x b_y D3_m \overline{ADDout}_s \overline{CARRYout}_t : \text{nil}) \tag{5}$$

where the Boolean evaluations are:

$$s = \begin{cases} 0, \text{ if } m = 0 \\ x \text{ XOR } y, \text{ if } m = 1 \end{cases} \quad \text{and } t = \begin{cases} 0, \text{ if } m = 0 \\ x \text{ AND } y, \text{ if } m = 1 \end{cases}$$

Note that the binary number $\overline{ts}_{(2)}$ is exactly the binary sum $x +_2 y$.

3.2 The formal proof of the agents bisimilarity

In this section we will prove that the two previous specification agents for the AU are bisimulation equivalent, the appropriate equivalence in the theory of concurrent communicating processes. This result is very important for the target of this paper since it means that the behaviour of the AU modeled by the implementation agent EArithm is correct with respect to the AU definition modeled by the specification agent SpecEArithm.

Proposition 1 *The previous agents SpecEArithm and EArithm are bisimulation equivalent.*

Proof: The bisimulation relation '$\sim$' is a congruence over the class $\mathcal{P}$ of agents [3].

We consider the low-level specification for the arithmetic part of the ALU given by the previous agent EArithm (4):

$$\begin{aligned} &\text{EArithm} = \\ &\quad = (\text{ArithmADD} \times \text{ArithmCARRY} \times \text{NODE2\_a} \times \text{NODE2\_b} \times \text{NODE2\_D3}) \backslash \\ &\quad \backslash \text{Comm\_EArithm} \end{aligned}$$

where the set of communicating actions is

$$\text{Comm\_EArithm} = \{upa, downa, upb, downb, upD3, downD3\}.$$

We evaluate this agent in few steps corresponding to the inside agents.

$$\begin{aligned} &\text{ArithmADD} = (\text{XORab} \times \text{EADD}) \backslash \{XORabout\} = \\ = \ &(\sum_{x,y\in\{0,1\}} (downa_x downb_y \overline{XORabout}_z : \text{nil}) \times \sum_{z,m\in\{0,1\}} (XORabout_z downD3_m \overline{ADDout}_s : \text{nil})) \backslash \\ \backslash \ &\{XORabout\} \end{aligned}$$

where z = x XOR y and s = z AND m.

After we apply the product (SCCS synchronous composition) and the restriction on the internal communicating action $XORabout$, the agent expression is:

$$\text{ArithmADD} = \sum_{x,y,m\in\{0,1\}} (downa_x downb_y downD3_m \overline{ADDout}_s : \text{nil})$$

Following the logic expressions s = z AND m and z = x XOR y we have s = z AND m = (x XOR y) AND m, meaning $s = \begin{cases} 0, \text{ if } m = 0 \\ x \text{ XOR } y, \text{ if } m = 1 \end{cases}$.

Analogously, the expression for the ArithmCARRY agent is:

$$\begin{aligned}&\text{ArithmCARRY} = (\text{ANDab} \times \text{ECARRY})\backslash\{ANDabout\} = \\ = \;& (\sum_{x,y\in\{0,1\}} (upa_x upb_y \overline{ANDabout}_z : \text{nil}) \times \sum_{z,m\in\{0,1\}} (ANDabout_z upD3_m \overline{CARRYout}_t : \text{nil}))\backslash \\ \backslash \;& \{ANDabout\}\end{aligned}$$

where z = x AND y and t = z AND m.

After we apply the product (SCCS synchronous composition) and the restriction on the internal communicating action $ANDabout$, the agent expression is:

$$\text{ArithmCARRY} = \sum_{x,y,m\in\{0,1\}} (upa_x upb_y upD3_m \overline{CARRYout}_t : \text{nil})$$

Following the logic expressions t = z AND m and z = x AND y we have t = z AND m = (x AND y) AND m, meaning $t = \begin{cases} 0, \text{ if } m = 0 \\ x \text{ AND } y, \text{ if } m = 1 \end{cases}$.

Following the previous definitions (1), (2) and (3) of the corresponding agents NODE2_a, NODE2_b and NODE2_D3, we have

$$\text{NODE2\_a} = \sum_{x\in\{0,1\}} (a_x \overline{upa}_x \overline{downa}_x : \text{nil})$$

$$\text{NODE2\_b} = \sum_{y\in\{0,1\}} (b_y \overline{upb}_y \overline{downb}_y : \text{nil})$$

$$\text{NODE2\_D3} = \sum_{m\in\{0,1\}} (D3_m \overline{upD3}_m \overline{downD3}_m : \text{nil})$$

Considering all the previous agents expressions and the set of the internal, communicating actions Comm_EArithm = $\{upa, downa, upb, downb, upD3, downD3\}$, we conclude that:

$$\begin{aligned}&\text{EArithm} = \\ &= (\text{ArithmADD} \times \text{ArithmCARRY} \times \text{NODE2\_a} \times \text{NODE2\_b} \times \text{NODE2\_D3})\backslash\text{Comm\_EArithm} = \\ &= (\sum_{x,y,m\in\{0,1\}} (downa_x downb_y downD3_m \overline{ADDout}_s : \text{nil}) \times \\ &\times \sum_{x,y,m\in\{0,1\}} (upa_x upb_y upD3_m \overline{CARRYout}_t : \text{nil}) \times \\ &\times \sum_{x\in\{0,1\}} (a_x \overline{upa}_x \overline{downa}_x : \text{nil}) \times \sum_{y\in\{0,1\}} (b_y \overline{upb}_y \overline{downb}_y : \text{nil}) \times \\ &\times \sum_{m\in\{0,1\}} (D3_m \overline{upD3}_m \overline{downD3}_m : \text{nil}))\backslash\{upa, downa, upb, downb, upD3, downD3\} = \\ &= \sum_{x,y,m\in\{0,1\}} (a_x b_y D3_m \overline{ADDout}_s \overline{CARRYout}_t : \text{nil})\end{aligned}$$

with the logic evaluations: $s = \begin{cases} 0, \text{ if } m = 0 \\ x \text{ XOR } y, \text{ if } m = 1 \end{cases}$ and $t = \begin{cases} 0, \text{ if } m = 0 \\ x \text{ AND } y, \text{ if } m = 1 \end{cases}$.

If you compare this final expression for the low-level specification agent EArithm with the definition of the high-level specification agent SpecEArithm given in (5) it is obvious that these two agents represent the same circuit behaviour, meaning they are bisimulation equivalent, as required. □

For $m = 1$, the final expressions for s and t are validating the name of the part we are discussing about, namely arithmetic unit. This is because the final logic expressions for s and t are modelling the two specific outputs of an half adder, respectively: s represents the sum of the two input bits and t represents the carry bit, as follows:

x	y	$x +_2 y$	t	s
0	0	$\overline{00}_{(2)}$	0	0
0	1	$\overline{01}_{(2)}$	0	1
1	0	$\overline{01}_{(2)}$	0	1
1	1	$\overline{10}_{(2)}$	1	0

This result of bisimilarity shows that the behaviour of the AU follows the definition of the corresponding arithmetic circuit and, on the other hand, it is a guarantee of using these agents in other complex models.

3.3 The automatic verification of the agents bisimilarity

For the implementation-specification pair of agents EArithm-SpecEArithm, we have used the CWB-NC platform [22] for verifying the appropriate agents bisimilarity. The corresponding CWB-NC answer for this test is TRUE and the specific result is pointed in Figure 3:

```
cwb-nc> es LoadArithmUAL.cws
Executing CWB-NC script file LoadArithmUAL.cws, directing output to std_out.
September 20, 2013   17:17
Execution time (user,system,gc,real):(0.001,0.000,0.000,0.001)
cwb-nc> Execution time (user,system,gc,real):(0.036,0.000,0.000,0.036)
cwb-nc> Execution time (user,system,gc,real):(0.003,0.000,0.000,0.003)
cwb-nc> ( The output has been put into std_out )
cwb-nc> eq EArithm SpecEArithm
Building automaton...
States: 4
Transitions: 16
Done building automaton.
Transforming automaton...
Done transforming automaton.
TRUE
Execution time (user,system,gc,real):(29.276,0.000,0.027,29.276)
cwb-nc> _
```

Figure 3: Automatic verification with CWB-NC

This CWB-NC answer authenticates the theoretical result proved above using the SCCS operational semantics. It is an important benefit of our work to have the implementation-specification pair of bisimilar agents, but, unfortunately, the execution time achieved here is not convenient. It is one of our future work targets to improve this time.

Using the CWB-NC is still a reliable approach, following the research interest revealed by the consistent publications like [7] or [5] relating to the CWB-NC, even in connection with CCS, SCCS and other modelling and verification tools.

4 Conclusions

It is our general target to obtain an algebraic-based formal framework for modelling and verification the computer system behaviour. This is following a multi-agent approach, each agent individually representing a specific computer hardware component. Out of our overall interests, both the specification

and implementation modelling levels, and verification of the CPU arithmetic unit behaviour have been considered in this paper.

For the given AU structure, we have defined appropriate SCCS agents based on the definition and on the behaviour of the AU and we have proved the bisimulation equivalence between the defined agents, authenticating the correctness of the behaviour with respect to the AU definition. Based on these results, it follows that we may use these agents in the next steps for modelling other hardware components having the AU structure as internal part, for example the more complex processing units.

We also consider as future work directions the possibility of moving on from this combination based on SCCS - CWBNC to another modern opportunities based on functional programming. At this moment, an interesting and modern solution could follow the Alvis project results for modelling and/or encoding the embedded, especially rule-based systems. Following [18], [21] and [17], Alvis is developing in Krakow, Poland starting with 2009. It is based on CCS and XCCS process algebras, it is defined for the design of concurrent especially real-time systems and it also provides a possibility of a formal model verification. One of the main Alvis advantages consists in combining a flexible graphical modelling approach for interconnections among agents with a high-level programming language used for the description of agents' behaviour. Even if Alvis is based on CCS and XCCS, its internal high-level programming language is based on the Haskell syntax instead of algebraic equations. In [21], the functional programming language Haskell [8] is appreciated as the most natural way of encoding a rule-based system into an Alvis model. Moreover, Haskell features like lazy evaluation, pattern matching or high level functions make it a very attractive proposition for the Alvis interests.

From our point of view, the Alvis project means an opportunity for future work consisting of replacing the equation-based algebraic modelling approach by a Haskell-based functional approach. From the educational point of view, the Haskell opportunities for our students are already a topic of our interests [20]. From the scientific point of view, passing to the functional approach is expecting to substantially improve the CWB-NC execution time obtained here for automatic verification of the agents' bisimilarity equivalences.

If Alvis is adding the Haskell facilities over the (X)CCS process algebra characteristics, we also have the alternative of the CHP library - as a set of Haskell packages for implementing the concurrency ideas from Hoare's CSP [2]. The beginning of Communicating Haskell Processes, namely CHP research framework is in [10]. Both Alvis and CHP have gathered the research and practical results in corresponding PhD thesis [12], [16].

References

[1] R. Milner, Calculi for synchrony and asynchrony. *Theoretical Computer Science* 25, 1983, pp. 267–310.

[2] C.A.R. Hoare, *Communicating sequential processes*, Prentice-Hall, 1985.

[3] R. Milner, *Communication and concurrency*, Prentice Hall, 1989.

[4] M.M. Mano, *Computer System Architecture*, Prentice Hall Intl., 1993.

[5] D. Zhang, R. Cleaveland, E.W. Stark, The Integrated CWB-NC/PIOATool for Functional Verification and Performance Analysis of Concurrent Systems. *Tools and Algorithms for the Construction and Analysis of Systems*, Lecture Notes in Computer Science Volume 2619, 2003, pp 431-436.

[6] A.S. Tanenbaum, *Structured Computer Organization*, Pearson Prentice Hall, 2006.

[7] L. Aceto, A. Ingolfsdottir, K. Larsen, J. Srba, *Reactive Systems: Modelling, Specification and Verification*, Cambridge University Press, 2007.

[8] G. Hutton, *Programming in Haskell*, Cambridge University Press, 2007.

[9] M.M. Mano, M. Celetti, *Digital Design*, Pearson Prentice Hall, 2007.

[10] N.C.C. Brown, Communicating Haskell Processes: Composable explicit concurrency using monads. *Communicating Process Architectures* 2008, pp.67-83.

[11] A. Vasilescu, Counter register. Algebraic model and applications. *WSEAS Transactions on Computers*, Issue 10, Vol. 7, pp. 1618-1627, Oct. 2008.

[12] P. Matyasik, *Design and analysis of embedded systems with XCCS process algebra*, PhD Thesis, AGH University of Science and Technology, Faculty of Electrical Engineering, Automatics, Computer Science and Electronics, Krakow, Poland, 2009.

[13] O. Rădoi, A. Ivan, A. Vasilescu, Logic Circuit Designer. *ACTA UNIVERSITATIS APULENSIS Mathematics - Informatics*, Special Issue as Proc of ICTAMI 2009, Alba-Iulia, Romania, September 3-6, 2009, pp.979-990.

[14] A. Vasilescu, Algebraic model for the synchronous D-flip-flop behaviour. *Proc. of Intl. Conf. Modelling and Development of Intelligent Systems MDIS 2009*, Sibiu, Romania, October 22-25, pp.308-315.

[15] A. Vasilescu, Algebraic model for the behaviour of a D-flip-flops-based memory component. *book Mathematical Methods, Computational Techniques, Intelligent Systems, Proc. of the 12^{th} WSEAS Intl Conf MAMECTIS'10*, El Kantaoui, Sousse, Tunisia, May 3-6 2010, pp.42-47.

[16] N.C.C. Brown, *Communicating Haskell Processes*, PhD Thesis, The University of Kent, Computer Science subject, UK, May 2011.

[17] M. Szpyrka et al., Introduction to modelling embedded systems with Alvis. *Automatyka*, Tom 15, part 2, 2011, pp.435-442.

[18] M. Szpyrka, P. Matyasik, R. Mrowka, Alvis - modelling language for concurrent systems. *Intelligent Decision Systems in Large-Scale Distributed Environments*, ser. Studies in Computational Intelligence, Springer-Verlag, Volume 362, 2011, ch. 15, pp 315-341.

[19] A. Vasilescu, A. Băicoianu, Algebraic model for the CPU logic unit behaviour. *book Recent Researches in Computer Science, Proc of 15th WSEAS Intl Conf on Computers (Part of the 15th WSEAS CSCC Multiconference)*, Corfu Island, Greece, July 15-17, 2011, pp. 521-526.

[20] A. Vasilescu, F.-R. Drobotă, Reasons for studying Haskell in University. *Proc. of The 7^{th} Intl Conf on Virtual Learning, Virtual Learning - Virtual Reality*, Braşov, Romania, November 2-3 2012, pp. 394-400.

[21] M. Szpyrka, T. Szmuc, Design and Verification of Rule-Based Systems for Alvis Models. *Intelligent Systems Reference Library*, Volume 43, 2013, pp 539-558.

[22] CWB *** The CWB-NC homepage on http://www.cs.sunysb.edu/~cwb.

Anca Vasilescu
Transilvania University of Braşov
Department of Mathematics and Computer Science
Iuliu Maniu Street 50, 500091 Braşov
ROMANIA
E-mail: *vasilex@unitbv.ro*

Third International Conference
Modelling and Development of Intelligent Systems
October 10 - 12, 2013
"Lucian Blaga" University
Sibiu - Romania

A Method for Sampling Random Databases with Constraints

Letiţia Velcescu, Dana Simian, Marius Marin

Abstract

In this article, we propose a method for sampling the contents of random databases. This type of database is important either in modeling uncertainty or storing data whose values follow a probability distribution. Such uncertain or random data appear in a variety of scientific fields. In order to perform analysis or validate the properties of these databases it is useful to have samples of their instances. Our work introduces a formalization of relations in databases, which will provide a sound basis for the sampling algorithms that we will present. We classified the categories of atomic constraints and focused on them, thus obtaining algorithms that ensure that data satisfy each class of constraints defined in the database. For the modeling of the external constraints, our approach uses the graph theory. We illustrate the importance of the algorithm in the case of generating surface data.

1 Introduction

Generally, information is useful and valuable as it can provide support in decision making. Data behind the information may come from different sources and sometimes it is ambiguous or potentially erroneous. Even under these circumstances, databases have to provide the answers that at least converge to the real, expected information ([10]). The domain of databases that store this type of uncertain or error carrying information is closely related to the random databases field ([4], [9], [13]). The importance of this type of database is significant in a variety of research domains, like natural sciences, medicine, telecommunications and economics. It should also be noted in the domains of research that manage data provided by sensors.

When dealing with random databases, it is often necessary to obtain samples of them; such samples would allow further statistical analysis and the discovery of meaningful results on random data ([6]). A previous formal framework for database sampling in presented in [2]. Our work provides a method of sampling a database whose model might be complex. The relations in the database are considered with their corresponding attribute domains and with their specific constraints.

The database constraints are classified in distinct, atomic types according to their functionality. Each category of constraints is studied, leading to a data generation algorithm which ensures their satisfaction. The case of the external constraints is more complex and it was treated using elements of the graph theory ([11]). Overall, our method brings as benefit the possibility to obtain a database instance, given its structure, constraints and attribute domains.

In our work, we considered even the most particular types of associations that could appear in real database models, leading to constraints. As a consequence, the proposed algorithm supports even more complex database design cases, in which a foreign key is compound or the table is auto-referred.

In the second section of this paper we propose a formalization of the concepts of database theory which are relevant to our approach. The third part presents the elements of graph theory that are needed in resolving the external constraints; also, their application to our method is described. The fourth part presents the algorithm for the processing of each type of constraints category defined in the database's model. In the fifth section, we illustrate the importance of our algorithm in the case of generating surface data for the study of entropy transfer during sliding ([3]). The article ends with a conclusions section that briefly emphasizes the contributions of the presented research.

2 Relational Databases

In order to describe the method that we conceived for database sampling, we firstly provide a formalization which is close to the approach of our technique. In this respect, we start from the main concepts of database, relation, attributes and constraints.

As preliminaries from the database theory, we recall that a database is a set of relations. Each relation of the database is described by its corresponding relation schema ([1], [5]), which usually contains the attributes names, their domains and the primary key. We extend this concept, considering that a relation is described both by the schema (attributes and constraints) and the implementation (table).

Formally, consider a relational database, which is a set of relations, denoted by $DB = \left\{ R_i \middle| i \in \overline{1,n} \right\}$, where R_i are the relations and $n \in \mathbb{N}$ is the number of relations in the database. Each relation R_i has three associated sets $A(R_i)$, $C(R_i)$ and $T(R_i)$ representing the set of attributes, the set of constraints and, respectively, the relation's implementation or table.

The attributes of a relation R are represented by the set $A(R) = \left\{ A_i \middle| i \in \overline{1,n_A} \right\}$, where A_i are the attributes and $n_A \in \mathbb{N}$ is the number of attributes, denoting the relation's arity ([5]). Each attribute A_i has an associated domain of values $D(A_i)$.

As a remark, an attribute of a relation can be represented only by its associated domain.

The domain of values of the attribute A_i is the set $D(A_i) = \left\{ v_j \middle| j \in \overline{1,n_i} \right\}$, where v_j are values and $n_i \in \mathbb{N}$ is the number of values in the domain. We consider that each domain D_i contains the value *null* which denotes unknown information.

In order to classify the database's constraints in atomic types, it can be noticed that, actually, the not null constraint is a particular case of conditional (check) constraint and the primary key constraint is a combination of uniqueness and not null. Subsequently, the constraints in a database can be classified in three atomic types: conditional, unique and external. An external constraint refers to columns in different tables, so it represents the foreign key constraints.

Hence, we can consider that the constraints of a relation R are organized by categories into the tuple $C(R) = (C_C, C_U, C_E)$, where C_C, C_U and C_E are the sets of conditional, unique and respectively, external constraints.

In terms of the formalism that we are introducing for the design of our method, the three sets of constraints specified in a relation are defined below.

Definition 1. The conditional constraints of a relation R are represented by the set $C_C(R) = \left\{ P_{X_i} \middle| i \in \overline{1,n_C} \right\}$, where P_{X_i} are predicates associated to subsets X_i of attributes that belong to $A(R)$, $X_{i_1} \neq X_{i_2}$, $\left|X_{i_1}\right| \leq \left|X_{i_2}\right|$ for each $i_1, i_2 \in \overline{1,n_c}$, $i_1 < i_2$, and n_C is the number of conditional constraints, $n_C \in \mathbb{N}$, $n_C \leq 2^{|A(R)|}$.

In the definition above, we consider that the attributes of the set X_i are involved in the definition of the predicate P_{X_i}.

Further, we consider that a unique constraint is represented by the set of attributes on which the uniqueness is declared:

Definition 2. The unique constraints of a relation R are represented by the set $C_U(R)=\left\{X_i \middle| i\in\overline{1,n_U}\right\}$, where $X_i \in A(R)$ are subsets of attributes belonging to $A(R)$, $X_{i_1}\neq X_{i_2}$, $\left|X_{i_1}\right|\leq\left|X_{i_2}\right|$ for each $i_1,i_2\in\overline{1,n_U}$, $i_1<i_2$, and n_U is the number of unique constraints, $n_U\in\mathbb{N}, n_U\leq 2^{|A(R)|}$.

The external constraints in the third category involve a second relation R', two subsets of attributes belonging to $A(R)$, respectively $A(R')$ and a bijection between these subsets, like in the following:

Definition 3. The external constraints of a relation R are represented by the set $C_E(R)=\left\{\left(R',G_{X_i,X_i'}\right)\middle| i\in\overline{1,n_E}\right\}$, where R' are database relations, $X_i\in A(R)$ are subsets of attributes belonging to $A(R)$, $X_{i_1}\neq X_{i_2}$, $\left|X_{i_1}\right|\leq\left|X_{i_2}\right|$ for each $i_1,i_2\in\overline{1,n_E}$, $i_1<i_2$, $X_i'\in C_U(R')$ are unique constraints that belong to R', $G_{X_i,X_i'}$ are the graphs of bijective functions $f_i:X_i\to X_i'$ and n_E is the number of external constraints, $n_E\in\mathbb{N}, n_E\leq 2^{|A(R)|}$. For each pair $(A,A')\in G_{X_i,X_i'}$, the associated attribute domains are equal, i.e. $D(A)=D(A')$.

We emphasize that in all three definitions given before the indices of the subsets X_i are ordered ascending according to the number of elements in these subsets. This order is important in our sampling algorithm for optimization purposes.

In database theory, a relation schema can be implemented as a table. A relation schema can have multiple instances, which are the states of the relation at given moments in time ([5]). The table is the container of the current instance of the relation schema.

Formally, the table of a relation R is the set $T(R)=\left\{t_i \middle| i\in\overline{1,n_R}\right\}$, where t_i are tuples of values, $t_i\in\prod_{A_j\in A(R)} D(A_j)$ and n_R is the cardinality of the relation, $n_R\in\mathbb{N}$.

In order to describe the sampling method we conceived we need to use the projection operation from the relational algebra ([1]) and to define the extension of this operation.

The projection of a table $T(R)$ on a set X of attributes that belong to $A(R)$ is a function π_X that associates to each database's table a set of tuples:

$$\pi_X(T(R))=\left\{t_i(X)\middle| i\in\overline{1,n_T}\right\}, \tag{1}$$

where $t_i(X)=\left(v_j \middle| A_j\in X\right)$ is the restriction of the tuple $t_i=\left(v_j\middle| j\in\overline{1,|A(R)|}\right)\in T(R)$ on the set of attributes X and n_T is the number of restricted tuples, $n_T\in\mathbb{N}, n_T\leq|T(R)|$.

The table $T(R)$ corresponding to a relation R has to satisfy the constraints $C(R)$ of the same relation, i.e. the following implications hold:

$\forall P_X\in C_C(R),\ \forall t\in T(R)\Rightarrow P_X(t)=true.$

$\forall X\in C_U,\ \forall t_1\neq t_2;t_1,t_2\in T(R)\Rightarrow \pi_X(t_1)\neq\pi_X(t_2)$

$\forall\left(R',G_{X,X'}\right)\in C_E(R)\Rightarrow \pi_X(T(R))\subseteq\pi_{X'}(T(R'))$

Further, we introduce the extension of this operation. We denote by $T_X = \left\{ T_i \middle| i \in \overline{1, n_{TX}} \right\}$ a table grouped by the projection on a set of attributes $X \subseteq A(R)$, where $T_i \subseteq T(R)$ are groups of tuples that partition the table $T(R)$:

$$\bigcup_{i \in 1, n_{TX}} T_i = T(R), \ \bigcap_{i \in 1, n_{TX}} T_i = \varnothing, \tag{2}$$

such that $\pi_X(t_1) = \pi_X(t_2)$ for each $t_1, t_2 \in T_i$, $\pi_X(t_1) \neq \pi_X(t_2)$ for each $t_1 \in T_{i_1}, t_2 \in T_{i_2}$, $i_1 \neq i_2$, and $n_{TX} \in \mathbb{N}$ is the number of groups.

Definition 4. Let be T(R) a table corresponding to a relation R and let $X_1, X_2 \in A(R)$ be sets of attributes. The extension of the projection of the table T(R) is the function:

$$\pi_{X_1}^{-1}\left(\pi_{X_2}(T(R))\right) = \pi_{X_2}(T(R)) \times \prod_{A_i \in X_1 - X_2} D(A_i), \tag{3}$$

where X_1 is the set of attributes of the extension and X_2 is the set of attributes of the projection.

3 Graphs of Relations and Graphs of Attributes

3.1. Preliminaries

A multiset M_A is the graph of a function $m: A \to \mathbb{N}^*$ which assigns an order of multiplicity to each element of the support set A.

A directed graph is a pair $G = (V, M_E)$, where V are the vertices and M_E are the edges. The vertices of a graph G are represented by the set $V(G) = \left\{ v_i \mid i \in \overline{1, n_V} \right\}$, where $n_V \in \mathbb{N}$ is the number of vertices. The edges of the graph G are represented by the multiset $M_{E(G)}$, where $E(G) = \left\{ e_i = (v_1, v_2) \middle| i \in \overline{1, n_E} \right\}$ denotes a set that contains pairs of vertices, $v_1, v_2 \in V(G)$, and n_E is the number of these pairs ([7]).

The external neighbours of a vertex v of the graph G are represented by the set:

$$V_v^+ = \left\{ v' \in V(G) \middle| \exists e = (v, v') \in E \right\}. \tag{4}$$

Similarly, the internal neighbours of a vertex v are specified as:

$$V_v^- = \left\{ v' \in V(G) \middle| \exists e = (v', v) \in E \right\}. \tag{5}$$

The sets defined in formula (4) and (5) allow us to introduce the external and the internal degrees of a vertex v of the graph G. These degrees, denoted by $d_G^+(v)$, $d_G^-(v)$ specify the number of outbound, respectively inbound edges relative to the vertex v ([7], [11]). They are given by the following formulas:

$$d_G^+(v) = \sum_{v' \in V_v^+} m\left((v, v')\right), \tag{6}$$

$$d_G^-(v) = \sum_{v' \in V_v^-} m\left((v', v)\right). \tag{7}$$

We denote the set of vertices ordered descending by their internal degrees as the set $V_{d_G^-} = \left\{ v_i \middle| i \in \overline{1, |V|} \right\}$, where $d_G^-(v_{i_1}) \geq d_G^-(v_{i_2})$, $1 \leq i_1 < i_2 \leq |V|$.

The network of paths that connect two vertices $v, v' \in V$ is the set $N_{v,v'} = \left\{ P_i \middle| i \in \overline{1, n_P} \right\}$, where P_i are paths that begin at v and end at v' and $n_P \in \mathbb{N}$ is the number of possible paths.

A path that belongs to a network $N_{v,v'}$ is an ordered set $P=\left(e_i \middle| i\in\overline{1,l_P}\right)$, where $e_i \in E$ are edges of the graph, $e_i=(v_i,v_{i+1})$, $v_1=v$, $v_{n+1}=v'$, and $l_P \in \mathbb{N}$ is the length of the path.
A network of closed paths is a network of paths $N_{v,v}$.

3.2. Graphs Modelling the Relational Databases

In our approach, we consider that a relational database DB has an associated pair of graphs (G_A,G_R), where G_A the graph of is attributes and G_R is the graph of relations. We introduce these concepts in the definitions that follow in this paragraph.
Let DB be a database as defined in section 2, with $|DB|=n$, $n\in\mathbb{N}$ and $|A(R_i)|=m_i$ for each $R_i \in DB$, $i\in\overline{1,n}$.

Definition 5. The graph of attributes associated to a relational database is the graph G_A, where:

$$V(G_A)=\left\{(R,A)\middle|\forall R\in DB,\forall A\in A(R)\right\} \tag{8}$$

and

$$E(G_A)=\left\{\left((R,A),(R',A')\right)\middle|\forall R\in DB,\forall\left(R',G_{X,X'}\right)\in C_E(R),(A,A')\in G_{X,X'}\right\}. \tag{9}$$

A vertex of the graph of attributes is specified by a pair composed of the relation name and an attribute of this relation. As we can notice, the support edges of G_A are defined by pairs of attributes that belong to external constraints.

Definition 6. The graph of relations correspondent to a relational database is the graph G_R, where $V(G_R)=\left\{R\middle|\forall R\in DB\right\}$ and

$$E(G_R)=\left\{(R,R')\middle|\forall R\in DB,\forall\left(R',G_{X,X'}\right)\in C_E(R)\right\}. \tag{10}$$

The support edges of G_R are defined by pairs of relations that are involved in external constraints.
In both graphs defined above, if an edge appears more than once, then the multiplicity of that edge increases adequately.
In the following we consider that a database is correctly defined if the graph of attributes has only networks of open paths, meaning that

$$\forall v\in V(G_A)\Rightarrow N_{v,v}=\varnothing. \tag{11}$$

4 Sampling Algorithm

The algorithm we introduce refers to the sampling of data in a random database, which is defined as presented in section 2 of this paper. Briefly, the data in the generated tables need both to belong to the specified attributes domains and to satisfy the constraints defined on the relations in the database. More, we suppose that the database is correctly defined, in the sense given in (11). In order to process all categories of constraints, the algorithm involves two stages which will be presented as separate methods.
We define the algorithm ALG that constructs the tables corresponding to the relations of a random relational database as a pair of methods $ALG=(M_{CU},M_R)$, where M_{CU} represents the method that constructs the tables according to the conditional and unique constraints and M_R represents the method that manipulates the constructed tables according to the reference constraints.

4.1. Method to Sample Tables with Conditional and Unique Constraints

We describe the method M_{CU} using the following pseudo code representation.

Input: The database *DB* having the tables associated to each relation empty, i.e. $T(R)=\varnothing, \forall R \in DB$.

1. BEGIN
2. FOR $i \in \overline{1,|DB|}$
3. $T=\pi_{\varnothing}(T(R_i));$
4. FOR $j \in \overline{1,|C_c(R_i)|}$
5. $P_{X_j} \in C_c(R_i);$
6. $T=\pi^{-1}_{X_j}(T);$
7. $T=T-\{t \in T \mid \neg P_{X_j}(t)\};$
8. END FOR
9. $T=\pi^{-1}_{A(R_i)}(T);$
10. FOR $j \in \overline{1,|C_u(R_i)|}$
11. $X_j \in C_u(R_i);$
12. $T_{X_j}=T;$
13. FOR $k \in \overline{1,|T_{X_j}|}$
14. $T_k \in T_{X_j};$
15. $T_k=select(T_k);$
16. END FOR
17. $T=\bigcup_{k=\overline{1,|T_{X_j}|}} T_k;$
18. END FOR
19. $T(R_i)=T;$
20. END FOR
END

Output: The database *DB* with the tables associated to each relation having data according to the defined conditional and unique constraints.

For each relation R_i of the database, the method above considers an auxiliary table T, initially with no rows and no columns (line 3). As the method uses the extension operation of projected tables, this table is initialized using the projection on the empty set of attributes.

For each conditional constraint defined on R_i, the method takes the attributes set X_j corresponding to the constraint's predicate P_{X_j} and applies the extension with X_j of the projection on the table T (line 6). The records that do not satisfy the predicate P_{X_j} are removed from the table T (line 7). After processing all the conditional constraints, the columns that were not included in these constraints are populated by the extension of the table T, which is actually the result of a projection (line 9).

Then, for each unique constraint defined on R_i, the method takes the corresponding attributes set X_j and considers the table T_{X_j} as the table T grouped by the projection on the attributes in X_j (line 12). For each group T_k in T_{X_j}, the function *select(T_k)* chooses a single random record (line 15), so that the unique constraint is satisfied. The selection method can use the simulation of any probability distribution (uniform, exponential, normal etc.), thus giving the database a random feature ([8], [12]).

The table T becomes the union of the atomic groups T_k (line 17). After processing all the unique constraints, the table T is associated to the relation R_i (line 19).

4.2. Method to Sample Tables with External Constraints

The method M_R in the algorithm ALG can be described using the following pseudocode representation.

Input: The database DB, with $|DB| = n$, having the tables associated to each relation generated with the method M_{CU}.

1. **BEGIN**
2. $used\left[\overline{1,n}\right] = 0;$
3. **FOR** $R_i \in V_{d_{G_R}^-}$
4. **IF** $used[i] = 0$
5. $X = \{R_i\};$
6. **WHILE** $X \neq \varnothing$
7. $X_{next} = \varnothing;$
8. $X_{d_{G_R}^-} = X;$
9. **FOR** $R_{i_1} \in X_{d_{G_R}^-}$
10. $T_{del} = \varnothing;$
11. **FOR** $j \in \overline{1, C_E\left(R_{i_1}\right)}$
12. $\left(R_{i_2}, G_{X_j, X_j'}\right) \in C_E\left(R_{i_1}\right);$
13. **FOR** $k \in \overline{1, \left|T\left(R_{i_1}\right)\right|}$
14. $t_k \in T\left(R_{i_1}\right);$
15. **IF** $\pi_{X_j}\left(t_k\right) \not\subset \pi_{X_j'}\left(T\left(R_{i_2}\right)\right)$
16. $T_{del} = T_{del} \cup \{t_k\};$
17. **END IF**
18. **END FOR**
19. **END FOR**
20. **IF** $T_{del} \neq \varnothing$
21. $T\left(R_{i_1}\right) = T\left(R_{i_1}\right) - T_{del};$
22. $X_{next} = X_{next} \cup V_{R_{i_1}}^-;$
23. **END IF**
24. $used[i_1] = 1;$
25. **END FOR**
26. $X = X_{next};$
27. **END WHILE**
28. **END IF**
29. **END FOR**
30. **END**

Output: The database DB with the tables associated to each relation having data according to the defined external constraints.

The method above marks a relation as used if it was treated by the algorithm, so that it will not be processed again. Each main iteration ensures that a connected component of the relations graph has been entirely treated. The relations in the database will be processed in descending order by the number of external constraints that refer to the relation, so that the relations referred most are treated first.
For each unused relation R_i in $V_{d_{\bar{G}_R}}$, an auxiliary set X is initialized with the element R_i (line 5). Again, we consider the relations set X ordered by the internal degree $X_{d_{\bar{G}_R}}$. For each relation R_{i_1} in this set, the algorithm takes all the relations R_{i_2} (line 9) that are referred by R_{i_1}; for each tuple t_k in the associated table $T(R_{i_1})$, if the projection on the set of attributes X_j defining the constraint is not found in the projection of the table $T(R_{i_2})$ on the corresponding attributes set X_j', then the tuple t_k is included in the set T_{del} for deletion (line 16).
After processing all the constraints, the tuples in T_{del} are eliminated from $T(R_{i_1})$ and the set X will contain the relations that depend through an external constraint on R_{i_1}. If no tuples have been eliminated, then the relations that depend on R_{i_1} will not be influenced by the current relation, so they are not included in the set X for further processing.
The algorithm can be used to generate random data for given models for the purpose of testing out various theories and their results.

5 Case Study: Generating Surface Data for the Study of Entropy Transfer during Sliding

In this section, we present a concrete example in which the algorithm described before can be used. The case study taken into account needs only one method from the algorithm *ALG* but it allows us to understand easier how the algorithm works. Many more complex examples using the proposed algorithm will be given in a future article.
In the study of entropy transfer during sliding presented in [3], the experiments and their subsequent computations started out with a generated surface. It was considered that this surface contained sites in one dimension and each site was described by its height. The data structure capable to retain the information of the surface is a simple one dimensional array which maps the heights as values and their corresponding sites as indices. Therefore, for a number of $n \in \mathbb{N}^*$ sites we need an array of length n which will contain at each i-th element the height $h_i \in \mathbb{R}$ of the i-th site.
We expressed the definition of the surface using an informatics data structure, so we consider transforming it further so it can match the relational database theory. We need a simple relation R in order to map the contents of the array into its associated table. The attributes set of R would minimally be $A(R) = (A_1, A_2)$, where A_1 represents the site and A_2 is the corresponding height. The relation needs to have a unique constraint defined on the attribute A_1, in order to ensure that the same site will not have two different heights. The constraints need to be $C = (C_C, C_U, C_E)$, where $C_U = \{\{A_1\}\}$ is the set of unique constraints. We will consider the external constraints to be unnecessary here, as in this case the data model is very simple. Therefore $C_E = \varnothing$. The conditional constraints can be more or less restrictive, depending on the purpose of the required data, so we will only consider that the height at each site needs to be strictly positive, $h_i > 0$,

$i \in \overline{1,n}$. Thus, the set of conditional constraints is formally defined as $C_C = \{P_{\{A_2\}}\}$, where $P_{\{A_2\}} = \{A_2 > 0\}$.

We have a relation schema, so next we will need actual values. We consider the domains associated to the attributes of R to be $D(A_1) = \{i \in \mathbb{N}^* \mid i \le 1000\}$ and $D(A_2) = [-2;2] \subset \mathbb{R}$.

There are two remarks that can instantly be made so far, but which will not prevent by any means the algorithm to run its course without error. The first observation is that $D(A_2)$ is an infinite set in the theory of real numbers so we cannot determine all the numbers that belong to it. In informatics, however, a number can only be expressed with a certain precision of digits, which is also limited by the memory available to store it, so in this case infinite numbers can only be approximated more or less.

Therefore, our domain will contain only the numbers in the given interval which have, for example, a precision of two digits, thus making it a finite set of numbers. The second observation is that the conditional constraint $A_2 > 0$ can be eliminated if we consider the domain $D(A_2)$ to contain strictly positive real numbers: $D(A_2) = (0;2]$. The reason for which the constraint is defined is to illustrate the possibility of introducing conditional constraints.

The algorithm performs the cross product between the domains of the attributes A_1 and A_2 and keeps only the tuples that satisfy the conditional constraints. In our case, the pairs that contain negative heights are discarded and the result looks like:

$$\begin{array}{cccccc} & 0.01 & & 0.01 & & 0.01 \\ 1 & 0.02 & 2 & 0.02 & \dots & 1000 & 0.02 \\ & \vdots & & \vdots & & & \vdots \\ & 2.00 & & 2.00 & & & 2.00 \end{array}. \tag{11}$$

The valid pairs are grouped by the projection on the set of attributes $X = \{A_1\}$, which represents the single unique constraint that we have defined. The algorithm selects a single pair from each group through the random selection method provided at the corresponding step of the algorithm. In our case we consider that the method makes a uniform selection, meaning that the pairs from each group have equal chances of being selected. The result could resemble: $\{(1,0.93),(2,1.07),\dots,(1000,1.40)\}$ or $\{(1,0.03),(2,0.11),\dots,(1000,1.97)\}$ or another of the 200^{1000} possible combinations.

6 Conclusion

The algorithm presented in this paper allows obtaining samples of databases with complex design that might also have to store uncertain or random data. Even for attributes domains having small cardinality, a large database can be generated. The random feature of our algorithm is justified by two aspects: the values of at least one domain can follow a certain probability distribution and the selection of a unique row can be randomly performed using a specific algorithm. In our future work, we will provide an optimization of the processing of the very restrictive constraints, which could result in the generation of empty tables due to the random selection of tuples.

The main contribution of the research presented in this article consists in the fact that, using our approach on relations and constraints, it provides the suitable means to sample relational databases. The benefit of such a sample is that it can be used for various purposes, including analysis and testing. As an example, we considered the case study regarding the generation of surface data.

The algorithm completely processes the databases constraints, allowing the implementation of any model, as it exhaustively covers all the foreseeable situations in database modeling.

Acknowledgements

This work was supported by the strategic grant POSDRU/89/1.5/S/58852, Project "Post-doctoral programme for training scientific researchers" co financed by the European Social Found within the Sectorial Operational Program Human Resources Development 2007-2013.

References

1. Abiteboul, S.; Hull, R.; Vianu, V., *Foundations of Databases*; Addison-Wesley, 1995.
2. Bisbal, J.; Grimson, J.; Bell, D. A Formal Framework for Database Sampling. *Information and Software Technology* **2005**, *47*, 819-828.
3. Fleurquin, P.; Fort, H.; Kornbluth, M.; Sandler, R.; Segall, M.; Zypman, F. Negentropy Generation and Fractality in the Dry Friction of Polished Surfaces. *Entropy* **2010**, *12*, 480-489.
4. Katona, G.O.H. Random Databases with Correlated Data. *Conceptual Modelling and Its Theoretical Foundations* **2012**, *7260*, 29–35.
5. Kifer, M.; Bernstein, A.; Lewis, P. *Database Systems. An Application-Oriented Approach*; Addison Wesley, 2005.
6. Lutu, P. Database Sampling for Data Mining. *Encyclopedia of Data Warehousing and Mining* **2009**, 604-609.
7. Popescu, D. *Combinatorics and Graph Theory* (in Romanian); Romanian Mathematical Society Publishing House, Bucharest, 2005.
8. Ross, S. *Simulation*; Academic Press, San Diego, London, 1997.
9. Seleznjev, O.; Thalheim, B. Random Databases with Approximate Record Matching. *Methodology and Computing in Applied Probabability* **2008**, *12*, 63-89.
10. Tchangani, A.P. A Model to Support Risk Management Decision-Making. *Studies in Informatics and Control* **2011**, *20*, 209-220.
11. Tomescu, I. *Combinatorics and Graph Theory* (in Romanian); University of Bucharest Publishing House, Bucharest, 1978.
12. Văduva, I. *Simulation Models* (in Romanian); University of Bucharest Publishing House, Bucharest, 2005.
13. Velcescu, L. Relational operators in heterogeneous random databases. *IEEE Proceedings of the 11th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, IEEE Computer Society Press, **2009**, 407-413.

VELCESCU LETIŢIA
University of Bucharest
Department of Informatics
14, Academiei, 010014, Bucharest
ROMANIA
E-mail: letitia@fmi.unibuc.ro

DANA SIMIAN
University "Lucian Blaga" of Sibiu
Department of Mathematics and Informatics
15, Ratiu, Sibiu
ROMANIA
E-mail: dana.simian@ulbsibiu.ro

MARIUS MARIN
ROMANIA
E-mail: marin_marius_89@yahoo.ro

Third International Conference
Modelling and Development of Intelligent Systems
October 10 - 12, 2013
"Lucian Blaga" University
Sibiu - Romania

Example of developing a loyalty program using CRM, SQL-queries and Rapid Miner tool

Iryna Zolotaryova, Iryna Garbuz, Mykhailo Dorokhov

Abstract

The object of paper is the implementing of loyalty programs using methods for intelligent processing of information and analysis the effectiveness of such a program. The subject is researching the loyalty program effectiveness based on Data Mining for the company "Auto-Maximum". The aim is to study the effectiveness of loyalty program based on Data mining. Results of performance are analyzed and described concept of direct marketing and loyalty programs features, research loyalty program development tools and algorithms for data mining. The results can be used in research institutions different companies that are aimed at marketing.

1 Introduction: defining main points of a loyalty program

The process of developing a loyalty program we start with describing the company for that we will develop the loyalty program. The firm "Auto-Maximum" has over 15 years of successful experience in the Ukrainian market of cosmetic service center in Kharkiv. Today the station is a recognized leader in this field, providing services to the population and not just implementing technology cosmetic service of cars, but also a whole range of new products, which includes everything you need for a successful business organization.

At the station the following services are offered for the cosmetic maintenance of vehicles: repair chips and cracks, installation (replacement) auto glass; toning of car windows; book lights; removing dents without paint; depth cleaning salon; professional polished body and glasses; bumper repair; accurate color matching; touch up minor defects; painting of individual components and car parts; installation of xenon light.

Before starting to develop the loyalty program itself, it is necessary to define the basic components of the program and the criteria.

Objectives. As with any project, the development of loyalty program begins with setting goals. A common mistake here is the lack of clear definitions. Of course, the key is to increase customer loyalty. To do this, you must define the parameters by which it will be possible to evaluate its success and effectiveness. If we consider a comprehensive loyalty, let, for example, indicators of transactional loyalty will be to increase the number of clients, done depleting repeat purchase, 40%, and the index increase customer satisfaction by 30%. Of course, that prior to the development of loyalty program we define for themselves their original positions in order to understand, from what we have to make a start and what we want to achieve.

Target Audience. Program is aimed at anyone? Who do we want to keep? Whose loyalty we want to raise? In accordance with the determined conditions, and the program. For example, we want

our clients are committed to those who commit at least 5 purchases per year (if it comes about, say, a clothing store), or uses our products for at least 1 time per week (if we're talking about food). In this case, once again I would like to stress the importance of market research and identify key basic parameters prior to the development of loyalty program.

Type of program. The best known and most widely used instrument - discount programs. Their essence is to provide customer benefits in the form of a refund of the paid value of the goods at the moment of purchase. There is a purely material gain. The second, also quite common - among lotteries that have made certain purchases in a given period of time. And even if the prize is not quite need a man - all the same emotions that accompanied the receipt of the prize, will leave a positive impression of your company. Another variety, have been gaining in popularity - accumulative discount programs. They benefit directly dependent on the participant: the more often and for a large amount buy, the more benefit you get. The fourth type - the bonus program. Their essence is that when shopping, the customer receives a certain conditional points, having accumulated a certain number of which he has the right to exchange them for goods or services on your own. By the way, on a product (as opposed to the lottery) this person is wanted and needed.

The other important component of customer loyalty programs are gift certificates in the form of a plastic card. This option is much more practical and presentable than usual, the paper certificate. Gift card will reflect your corporate identity, advertising your organization, and after use can be presented as a discount card, or used in the prize draw.

Privileges. The most complex, interesting and creative stage - the definition of what to offer the customer, in addition to the main component - the bonuses. And there scope for creativity is so wide that you can lose sight of the most important - the needs of customers! It is they who must determine the entire range of additional benefits. We can distinguish three stages of its preparation: a preliminary study of the resulting list on a limited sample of clients; a large-scale survey; creative development of all possible privileges. The final list is generated taking into account factors such as the feasibility of the franchise, the competence of the company in its implementation, and cost.

Financial concept. The most sensitive issue is related to the assessment of future expenses for the program and they should be met. Thus, the costs associated with the cost of accrual of bonuses, discounts, production of advertising and souvenir production, club cards, acquisition or development of specialized software, compensation of employees responsible for the operation of the program. Covering the costs may come at the expense of the annual contribution of participants, foreclosures club card, etc. Also, there are bottlenecks, such as, for example, the account of the bonuses in the company's accounting or determination techniques for tracing the influence of loyalty programs for sales, profitability and revenue. All this requires a thorough study before implementation software.

Management and Communication. There are three areas of communication: between the company and customers, between the company and the external environment, as well as within the company. The system of interaction with the participants can be built, for example, on the basis of telephone communication with a call-center manager and loyalty programs, newsletters, holiday greetings, prompt responses to emails and customer complaints on the message in the forum. Communication with the external environment may include media publications, participation in conferences on loyalty marketing. Intra-corporate communications related to the interaction of all departments of the company, assess the effectiveness of its activities, etc.

Technology. Requires careful attention to the creation of a single control center loyalty program, coordinating key areas of its operation, the center for the processing of incoming calls; IT-system maintenance program; logistics system and algorithm implementation procedures of the program.

Database. Loyalty program - a great tool to collect and store data about customers. Before starting the program is to determine what data and how much should be entered into the database, how and with what frequency analyzed what this will require resources, both technical and human. Unfortunately, many companies implementing loyalty programs and have extensive databases, they are used inefficiently. The reasons for this - lack of knowledge of how to efficiently use the information collected, how to develop individual offers for each customer

segment, the technical complexity of the implementation of analytical processes, the unreliability of data, etc.

Closing the program. The problem to which few people paid attention to when you start the program because of optimistic beginning. But do not forget that every project has its own life cycle. There may come a time when the program will cease to be effective. Decide in advance with the critical exponents for the attainment of which is necessary to minimize the program. Will it be transformed into something new normed? If not, how staff will be disbanded and its service? How does the database will be used? These and other questions worth pondering in advance.

The final concept of the program include such elements: objectives (the increase in the number of repeat orders and increase the return up to 15%), target audience (group of clients with large number of orders for a large sum), type of program (discount), management and communication (between company and customers), technology (Data mining. CRM, Rapid Miner).

2 Developing of the software for collecting information

To develop our loyalty program we must have the information about: information about range of services provided by this company; general information about clients; information about frequence of services and sum for them; information about cooperation clients and the firm etc.

So, to answer to these questions we create a special program. The main aim of this program is to provide an opportunity to collect information about clients and service. This program is a kind of Customer Relation Management. The manager of the firm record information about every service, order etc during the day. As result, we accumulated data about clients, services and provided services. Moreover, this program provides an opportunity to get information for making decision. But, first of all, let us consider the model of the data base which our application bases on.

The normal ER-model (Fig. 1) consists of 4 entities: “Customer”, “Service”, “Sale header” ,“Sale detail”. Entity “Customer” includes information about clients such as Name, Surname, Phone, Fax etc. Entity “Service” contains list of services provided by the company. And entities “Sale header” and “Sale detail” contain information about orders: date, client, what services were ordered and the sum of order. There are connections between entities such as "one to many".

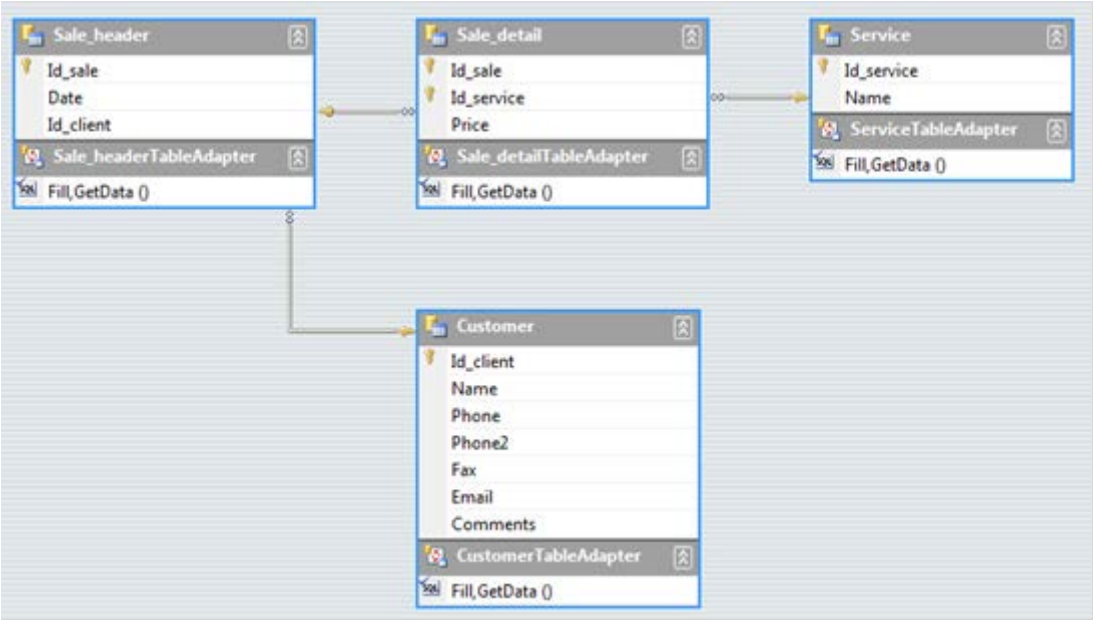


Figure 1. Logic model of data base

The logic model matches completely with physical model of data base (Fig. 2).

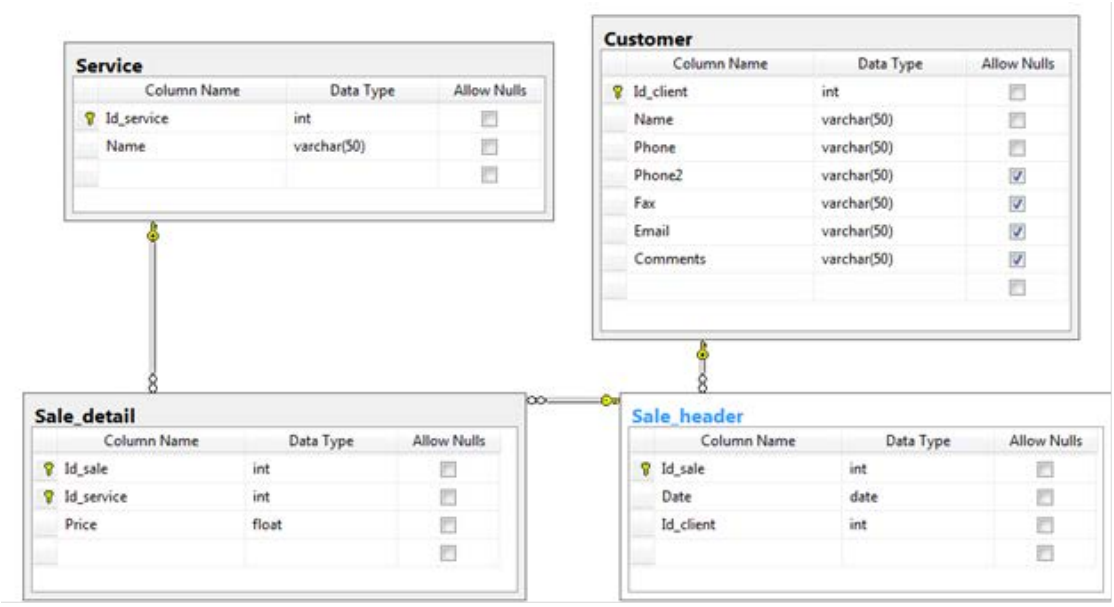


Figure 2. Physical model of data base

Also we construct graphic user interface which consists of units (Clients, Services, Orders and Reports) with main window and additional dialog windows. Also, the program provides an analytical functionality. With the help of this program, a manager can get useful information for making decision such as monthly orders, quantity of services for a month, 3 months and year.

3 Analyzing of tools for Data Mining

At first we analyzed (Fig. 3) statistics about data mining/analytic tools which are most popular in the past 12 months for a real project. Direct comparison votes may not be representative, because of different verification strategies in 2011 and 2012, but clearly that the leading open source tools were Rapid Miner, R, and KNIME. Among commercial tools, the top tools were SAS, MATLAB, and IBM SPSS Modeler (former Clementine).

Rapid Miner Tool is one of the leading open source data mining software suites. With more than 400 data mining modules or operators, it is one of the most comprehensive and most flexible data mining tools available. With over 10,000 downloads from SourceForge.net each month and more than 300,000 downloads in total, it is also one of the most widespread-used data mining tools. According to polls by the popular data mining web portal KDnuggets.com among several hundred data mining experts, Rapid Miner was the most widely used open source data mining tool and among the top three data mining tools overall in 2011 and 2012.

Rapid Miner supports all steps of the data mining process from data loading, pre-processing, visualization, interactive data mining process design and inspection, automated modeling, automated parameter and process optimization, automated feature construction and feature selection, evaluation, and deployment. Rapid Miner can be used as stand-alone program on the desktop with its graphical user interface (GUI), on a server via its command line version, or as data mining engine for your own products and Java library for developers.

Rapid Miner is provided by Rapid-I, an open source data mining and business intelligence software and consulting company based in Dortmund, Germany. Rapid Miner has users in more than 30 countries and Rapid-I serves customers on four continents. Rapid-I won the Open Source Business Award 2008.

Which data mining/analytic tools you used in the past 12 months for a real project (not just evaluation) [912 voters]	
RapidMiner (345)	37.8%
R (272)	29.8%
Excel (222)	24.3%
KNIME (175)	19.2%
Your own code (168)	18.4%
Pentaho/Weka (131)	14.3%
SAS (110)	12.0%
MATLAB (84)	9.2%
IBM SPSS Statistics (72)	7.9%
Other free tools (67)	7.3%
IBM SPSS Modeler (former Clementine) (67)	7.3%
Microsoft SQL Server (63)	6.9%
Statsoft Statistica (57)	6.2%
Other commercial tools (56)	6.1%
SAS Enterprise Miner (50)	5.5%
Zementis (34)	3.7%
Orange (25)	2.7%
Oracle DM (19)	2.1%
KXEN (19)	2.1%
Salford CART Mars other (15)	1.6%
VisuaLinks (12)	1.3%
Viscovery (10)	1.1%
Angoss (8)	0.9%
TIBCO Insightful Miner (7)	0.8%
Miner3D (7)	0.8%
REvolution Computing (4)	0.4%
Megaputer Polyanalyst/TextAnalyst (3)	0.3%
Portrait Software (2)	0.2%
Data Applied (2)	0.2%
Centrifuge (2)	0.2%

Figure 3. Survey Results

Besides Rapid Miner, Rapid-I offers data mining training courses, consulting, professional support, adaptations and extensions of Rapid Miner, individual software development, integration, and other data mining services.

Some of important features of Rapid Miner are listed as follows: Rapid Miner (previously known as YALE) is based on modular operator concept which facilitates rapid prototyping of data mining processes by way of nesting operator chains and using complex operator trees; a large numbers of operators (more than four hundred) defined in Rapid Miner along with its plugins cover nearly all key aspects of Data mining handling data transparently and without the need to know the different data formats or different data views; Rapid Miner provides flexible operators for data input and data output in different file formats such as excel files, files, SPSS files, data sets from well-known databases such as Oracle, my SQL, Microsoft SQL Server, Sybase, and dBase.

Owing to the modular operator concept, the data mining processes are optimized because, by substituting or replacing one particular operator at a time and leaving rest of the data mining process design untouched, its performance can be evaluated.

Rapid Miner follows a multi-layered data view concept which enables it to store different views

on the same data table and therefore facilitates cascading multiple views in layers through a central data table. Rapid Miner data core is typically similar to a standard database management system. Rapid Miner has a flexible interactive design which lets user to additional meta data on the available data sets to enable automated search and optimized preprocessing which are both needed for an effective data mining processes. Rapid Miner also acts as a powerful scripting language engine along with a graphical user interface.
Since using Rapid Miner, data mining processes are designed as operator trees defined in XML, where operators are not defined in a graph layout so as to be positioned and connected by a user. Therefore data flow normally follows "depth first search," resulting in optimization of data mining processes.

4 Clustering of the customer database and choosing the target group

Clustering is a data mining method that analyzes a given data set and organizes it based on similar attributes [1-4]. Clustering can be performed with pretty much any type of organized or semi-organized data set, including text, documents, number sets, census or demographic data, etc. The core concept is the cluster, which is a grouping of similar objects. Clusters can be any size – theoretically, a cluster can have zero objects within it, or the entire data set may be so similar that every object falls into the same cluster [5-8].
As one of the main benefits from the application of the loyalty program is an opportunity to focus on a specific group of customers who make the most out of to a company, so an important point of their promotion effectiveness is the process of a segmentation of client base and selection of the most attractive consumers. And then you can build a relationship with clients in certain segments that have common characteristics. This allows you to create specific marketing programs.
To cluster clients of the company we use annual report presented by our CRM. This report contains information about how many services for what amount of money the clients made during the half of year. So, in this analytical report we summarize quantity of services and their cost for each client separately. This information helps us choose the most interesting group of clients for us. The most interesting group for us is the group that ordered more services and for the largest sum than the average amount.
Clustering is a great first step to use when looking at a large data set. In order to perform clustering, some setup is required. First, the data set must be prepared and cleaned (Replace Missing Values). Second, the numerical data must be separated into a subset (Work on Subset). Third, the clustering algorithm must be defined and applied (Clustering). Lastly, the output must be examined in order to check for quality and usefulness.
There are many different types of clustering algorithms. Some of the most advanced methods in 2012 revolve about support-vector models (SVM), the CLOPES and COBWEB algorithms, or clustering by expectancy. Unfortunately, these clustering methods require an intense amount of computing power. The K-Means algorithm is the simplest clustering method and also probably the most efficient given limited technology. It may not be cutting edge, but the results are still valid and useful for any data miner looking for the broadest of insights.
First of all, we import our report to Rapid Miner. Then with the help of Rapid Miner's operators we build the model of clustering process (Fig. 4). Our model contains 3 operators: Retrieve as input, nominal to numerical as a convertor and clustering operator as the main component. We use K-means algorithm and determine that all data will be divided into 5 clusters (k=5). We choose k-means algorithm for several reasons: as we have not large database this algorithm works really fast; simplicity of algorithm's logic; opportunity to choose quantity of clusters etc.
Retrieve reads an object from the data repository. Nominal to Numerical operator changes the type of selected non-numeric attributes to a numeric type. It also maps all values of these attributes to numeric values. Clustering operator performs clustering using the k-means algorithm. Clustering is concerned with grouping objects together that are similar to each other and

dissimilar to the objects belonging to other clusters. Clustering is a technique for extracting information from unlabeled data. K-means clustering is an exclusive clustering algorithm i.e. each object is assigned to precisely one of a set of clusters.

Figure 4. Clustering Process Model

The clustering process results are grouped into 5 clusters (Fig. 5). Centroid result table presents the average characteristics of each cluster. We will choose cluster 4, reasons are followed: this cluster has high level of average amount of orders; this cluster has high level of average amount of money spent for orders; this cluster has 28 items, so it is not very large. But according to 80/20 Pareto theory, 20 percent of clients generate 80 percent of profit; this cluster has the most close location of its items.

Text View | Folder View | Graph View | Centroid Table | Centroid Plo

Attribute	cluster_0	cluster_1	cluster_2	cluster_3	cluster_4
Количество	1.847	10.857	4.882	3.188	7.214
Сумма	645.678	9928.571	3565.263	1873.419	6505.357

```
Cluster 0: 236 items
Cluster 1: 7 items
Cluster 2: 76 items
Cluster 3: 160 items
Cluster 4: 28 items
Total number of items: 507
```

Figure 5. Cluster Model Results and Centroid Table

Market basket analysis aims to detect relationships or associations between specific items in a large "catalog" of objects. A simple example would be the occurrence of diapers and baby formula in the same sales transaction. However the real value from association rules analysis is finding connections between seemingly non-intuitive items: it would be indeed surprising (and valuable) to the retailer if it was found that there was a strong association between beer and diapers! But how do you know if this association rule has statistical value and not just some random coincidence? One way to test this is by calculating the Lift Ratio.

We discussed a couple of key measures which are needed to correctly apply association rules for market basket analysis: Support and Confidence. A third quantity of importance is lift. Lift Ratio is defined as the ratio between the confidence of a rule and a benchmark confidence. Benchmark confidence is simply the confidence of the two items if they were independent. Lift ratio higher than 1.0 indicates that there is some value to the rule.

There are three steps for generating usable recommender systems using a data mining tool such a Rapid Miner: data preparation, rule generation and rule application. Data may consist of products sold, transaction details and customer ids if we are discussing retail. Data has to be transformed into a format where each row is a separate transaction id (or session id) and each column is a product sold (page viewed). The cells of this table will be 0's or 1's depending upon whether that particular product was sold during a given transaction. To transform data we used the SQL query.

The well-known algorithm proceeds in two steps within the support-confidence framework (minimum support and confidence thresholds have to be fixed by the user) in order to extract association rules:

• find frequent item sets (the sets of items which occur more frequently than the minimum support threshold) with the frequent item set property (any subset of a frequent item set is frequent; if an item set is not frequent, none of its supersets can be frequent)for efficiency reasons. Thus starting from k = 1, generates item sets of size k + 1 from frequent item sets of size k,

• generate rules from frequent item sets and filter them with the minimum confidence threshold.

Our process model consists of four elements: input Retrieve, Numerical to Binominal for conversion the FP-Growth element for calculating frequencies of items in the data and Create Association Rules element for creating rules. Then, we describe all components of our model.

Retrieve (Rapid Miner Core) can be used to access the repositories. It should replace all file access, since it provides full metadata processing, which eases the usage of Rapid Miner a lot. In contrast to accessing a raw file, it provides the complete metadata of the data, so all metadata transformations are possible.

Numerical to Binominal changes the type of the selected numeric attributes to a binominal type. It also maps all values of these attributes to corresponding binominal values. The Numerical to Binominal operator changes the type of numeric attributes to a binominal type (also called binary). This operator not only changes the type of selected attributes but it also maps all values of these attributes to corresponding binominal values. Binominal attributes can have only two possible values i.e. 'true' or 'false'. If the value of an attribute is between the specified minimal and maximal value, it becomes 'false', otherwise 'true'. Minimal and maximal values can be specified by the min and max parameters respectively. If the value is missing, the new value will be missing. The default boundaries are both set to 0.0, thus only 0.0 is mapped to 'false' and all other values are mapped to 'true' by default.

FP-Grow the efficiently calculates all frequent items from the given Example Set using the FP-tree data structure. It is compulsory that all attributes of the input Example Set should be binominal. In simple words, frequent item sets are groups of items that often appear together in the data. It is important to know the basics of market-basket analysis for understanding frequent item sets.

The market-basket model of data is used to describe a common form of a many-to-many relationship between two kinds of objects. On the one hand, we have items, and on the other we have baskets, also called 'transactions'. The set of items is usually represented as set of attributes. Mostly these attributes are binominal. The transactions are usually each represented as examples of the Example Set. When an attribute value is 'true' in an example; it implies that the corresponding item is present in that transaction. Each transaction consists of a set of items (an item set). Usually it is assumed that the number of items in a transaction is small, much smaller than the total number of items i.e. in most of the examples most of the attribute values are 'false'. The number of transactions is usually assumed to be very large i.e. the number of examples in the Example Set is assumed to be large. The frequent-item sets problem is that of finding sets of items that appear together in at least a threshold ratio of transactions. This threshold is defined by the 'minimum support' criteria. The support of an item set is the number of times that item set appears in the Example Set divided by the total number of examples. The 'Transactions' data set at "Samples/data/Transactions" in the repository of Rapid Miner is an example of how transactions data usually look like.

The discovery of frequent item sets is often viewed as the discovery of 'association rules', although the latter is a more complex characterization of data, whose discovery depends fundamentally on the discovery of frequent item sets. Association rules are derived from the frequent item sets. The FP-Growth operator finds the frequent item sets and operators like the Create Association Rules operator uses these frequent item sets for calculating the rules.

This operator calculates all frequent item sets from an Example Set by building a FP-tree data structure on the transaction data base. This is a very compressed copy of the data which in many cases fits into main memory even for large data bases. All frequent item sets are derived from this FP-tree. Many other frequent item set mining algorithms also exist e.g. the Apriori algorithm. A major advantage of FP-Growth compared to Apriori is that it uses only 2 data scans and is therefore often applicable even on large data sets.

Please note that the given Example Set should contain only binominal attributes, i.e. nominal attributes with only two different values. If your Example Set does not satisfy this condition, you may use appropriate preprocessing operators to transform it into the required form. The discretization operators can be used for changing the value of numerical attributes to nominal

attributes. Then the Nominal to Binominal operator can be used for transforming nominal attributes into binominal attributes.
Please note that the frequent item sets are mined for the positive entries in your Example Set, i.e. for those nominal values which are defined as positive in your Example Set. If data does not specify the positive entries correctly, you may set them using the positive value parameter. This only works if all your attributes contain this value.
This operator has two basic working modes:

- finding at least the specified number of item sets with highest support without taking the 'min support' into account. This mode is available when the find min number of item sets parameter is set to true. Then this operator finds the number of item sets specified in the min number of item sets parameter. The min support parameter is ignored in this case.
- finding all item sets with a support larger than the specified minimum support. The minimum support is specified through the min support parameter. This mode is available when the find min number of item sets parameter is set to false.

5 Criterion and creating of Association Rules

Create Association Rules generates a set of association rules from the given set of frequent item sets. Association rules are if/then statements that help uncover relationships between seemingly unrelated data. An example of an association rule would be "If a customer buys eggs, he is 80% likely to also purchase milk." An association rule has two parts, an antecedent (if) and a consequent (then). An antecedent is an item (or item set) found in the data. A consequent is an item (or item set) that is found in combination with the antecedent.
Association rules are created by analyzing data for frequent if/then patterns and using the criteria support and confidence to identify the most important relationships. Support is an indication of how frequently the items appear in the database. Confidence indicates the number of times the if/then statements have been found to be true. The frequent if/then patterns are mined using the operators like the FP-Growth operator. The Create Association Rules operator takes these frequent item sets and generates association rules. Such information can be used as the basis for decisions about marketing activities such as, e.g., promotional pricing or product placements. In addition to the above example from market basket analysis association rules are employed today in many application areas including Web usage mining, intrusion detection and bioinformatics.
Let describe specifies the criterion which is used for the selection of rules.
Confidence. The confidence is defined conf(X implies Y)=sup (X ∪Y)/ sup (X) . Be careful when reading the expression: here sup (X∪Y) means "support for occurrences of transactions where X and Y both appear", not "support for occurrences of transactions where either X or Y appears". Confidence ranges from 0 to 1. Confidence is an estimate of Pr(Y | X), the probability of observing Y given X. The support sup (X) of an item set X is defined as the proportion of transactions in the data set which contain the item set.
Lift. The lift of a rule is defined as lift(X implies Y) = sup (X ∪ Y)/((sup (Y) x sup (X)) or the ratio of the observed support to that expected if X and Y were independent. Lift can also be defined as lift(X implies Y) =conf(X implies Y)/ sup (Y). Lift measures how far from independence are X and Y. It ranges within 0 to positive infinity. Values close to 1 imply that X and Y are independent and the rule is not interesting.
Conviction. Conviction is sensitive to rule direction i.e. conv(X implies Y) is not same as conv(Y implies X). Conviction is somewhat inspired in the logical definition of implication and attempts to measure the degree of implication of a rule. Conviction is defined as conv(X implies Y) =(1 - supp(Y))/(1 – conf(X implies Y)).
Gain. When this option is selected, the gain is calculated using the gain theta parameter.
Laplace. When this option is selected, the Laplace is calculated using the laplace k parameter.
As result of the process, corresponded rules with confidence 0.6 has been created. We obtained seven rules. As we defined the confidence equal to or more than 0.6, all our rules satisfy our limits. We can see that results contain two or even three services (units).

The suitable presentation of results is graph (Fig. 6). The relations between services include confidence and support parameters.

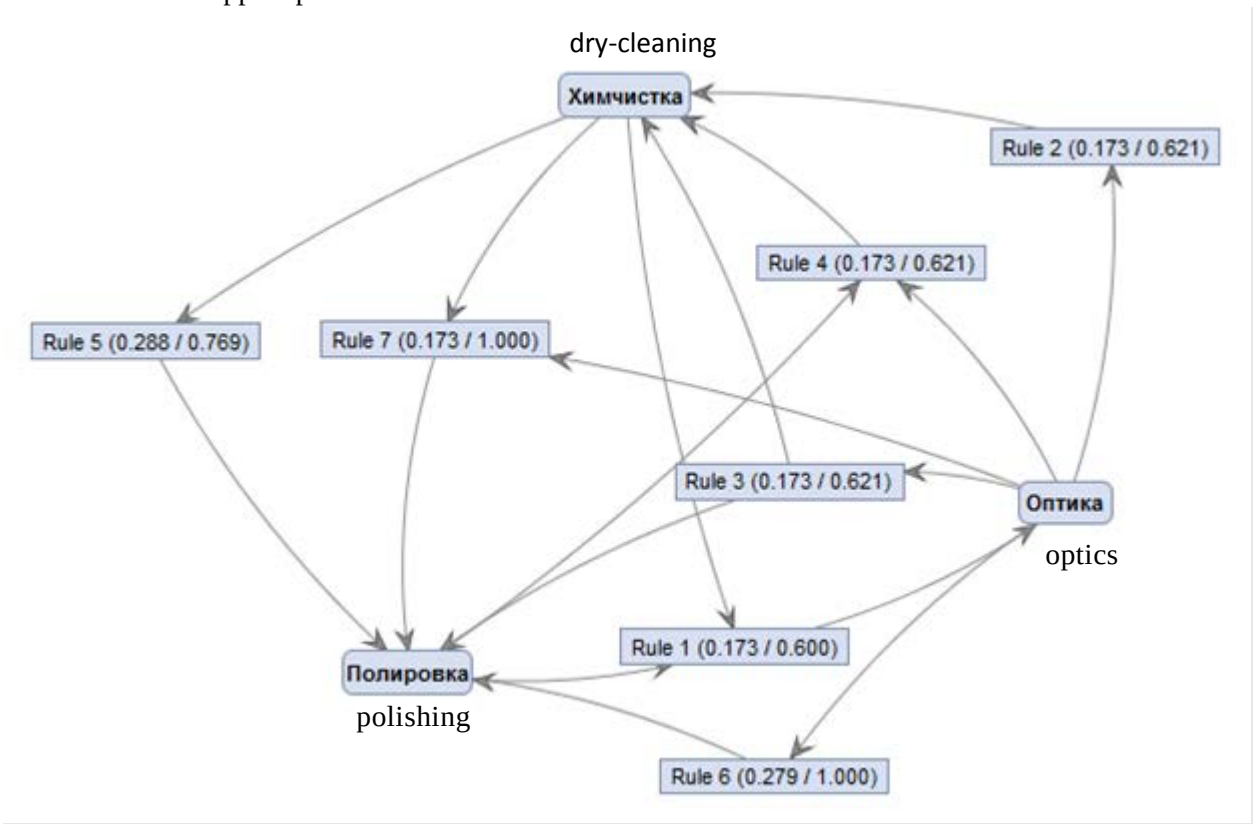


Figure 6. Results in the graph form

According to resulted association rules, we have strong relations between three services: “polishing”, “dry-cleaning” and “optics”. All rules cover these services. So, we should create the special offer for our clients that include these services. Prices for “polishing”, “dry-cleaning” and “optics” are presented at Fig. 7.

Service	Price
polishing	800 UAH
dry-cleaning	1200 UAH
optics	200 UAH
Sum for a package:	2200 UAH

Figure 7. Prices for services

Prime prices for “polishing”, “dry-cleaning”, “optics” are presented at Fig. 8. According these can be provided discount 15% for services package for clients without cost lost for the company.

Service	Price
Polishing	650 UAH
dry-cleaning	900 UAH
Optics	50 UAH
Sum for a package:	1600 UAH

Figure 8. Prime prices for services

6 Conclusions: evaluating of loyalty program effectiveness

The goal of any loyalty program - is to increase the efficiency of marketing and sales channels, increase customer loyalty, and as a consequence - increase profits. The goal of our program is the increase of quantity of repeat orders and, consequently, the increase of return. We can compare quantity of sets with selected services in one order during the quarter before and after implementing the loyalty program. So, we will analyze 4th quarter of 2012 year and 1st quarter of 2013 year. You can see result on diagrams (Fig. 8). For 4th quarter 2012 we have 15 sets of services with 2 repeat orders in it and for 1st quarter of 2013 - 24 (7). So, we can see increase in number of orders - 40% and in number of repeat orders – 350%.

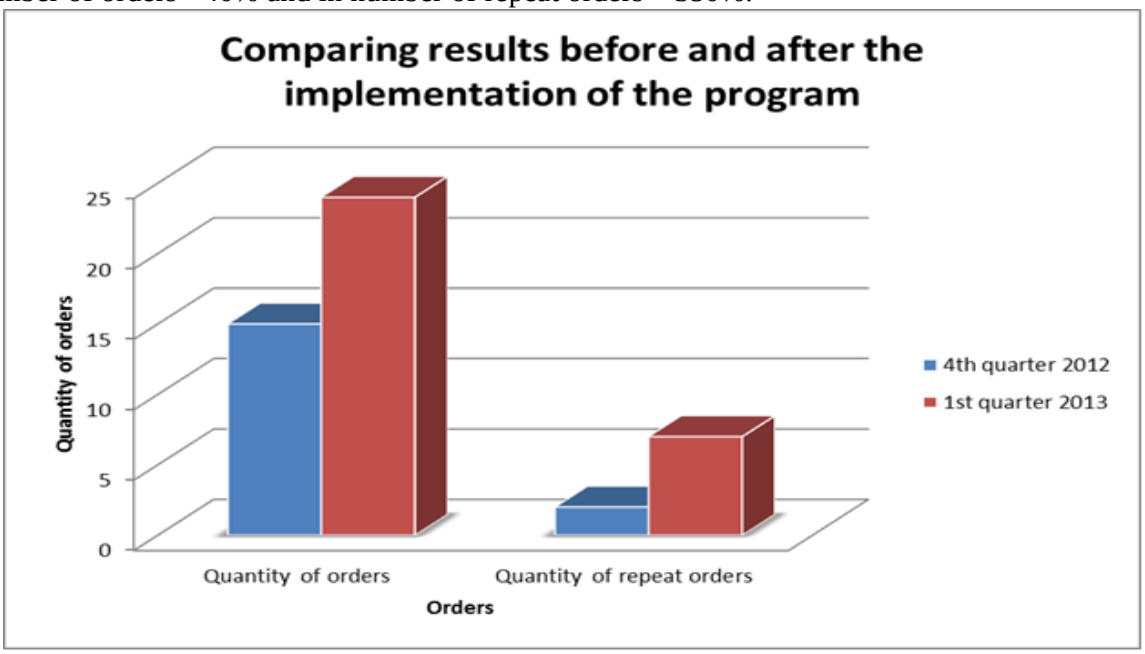


Figure 8. Diagram “Comparing results before and after the implementation of the program”

Also, we calculate the profit from this program (Fig. 9). In the 4th quarter 2012 company profits were 33.000 UAH, in 1st quarter 2013 they are 41140 UAH. The profit increase amounted 25%.

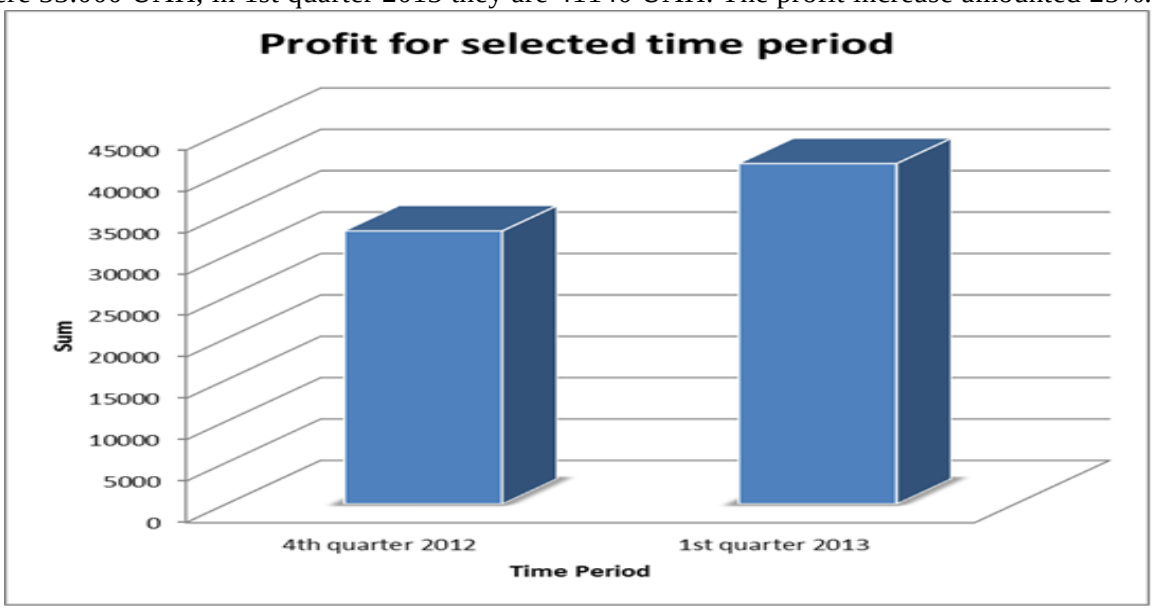


Figure 9. Diagram “Profit for selected time period”

Finally, our loyalty program resulted in not only the growth of client's orders included repeat orders but also the increase of company profit. And we can see these results only after 3 months. This is great result. We achieved our goals during 3 months of implementing of our program.

References

[1] A. Romeu. Cluster Detection in Laboratory Auction Data: A Model-Based Approach. *Panoeconomicus*, 58(4): 473–488, 2011.
[2] H. Shah-Hosseini. Binary Tree Time Adaptive Self-Organizing Map. *Neurocomputing*, 74 (11): 1823-1839, 2011.
[3] H. Chu Chai. Intelligent value –based customer segmentation method for campaign management: A case study of automobile retailer. *Expert Systems with Applications*, 34(4): 2754-2762, 2008.
[4] Y.S. Kim and W.N. Street. An intelligent system for customer targeting: A data mining approach. *Decision Support Systems*, 37, 215–228, 2004.
[5] R. Ng and J. Han. Efficient and Effective Clustering Method for Spatial Data Mining. *Proc. of 1994 Int'l Conf. on Very Large Data Bases (VLDB'94)*, Santiago, 1994, 144-155.
[6] J.Sander and M. Ester M. Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and its Applications. *Data Mining and Knowledge Discovery*, Vol. 2, No. 2, 169-194, 1998.
[7] P. Cheeseman and J. Stutz. Bayesian Classification (AutoClass): Theory and Results. *Advances in Knowledge Discovery and Data Mining*, AAAI/MIT Press, 1996, 153-180.
[8] T. Zhang and R. Ramakrishnan. BIRCH: An Efficient Data Clustering Method for Very Large Databases. *Proc. of ACM SIGMOD Int'l Conf. on Data Management*, Canada, 1996.

ZOLOTARYOVA Iryna
Kharkiv National University of Economics
Department of Informations Systems
Kharkiv
UKRAINE
izolotaryova@gmail.com

GARBUZ Iryna
Kharkiv National University of Economics
Department of Informations Systems
Kharkiv
UKRAINE
kafis@hneu.edu.ua

DOROKHOV Mykhailo
Kharkiv National University of Economics
Department of Informations Systems
Kharkiv
UKRAINE
michaeldorokhov@gmail.com

List of authors

Octavian AGRATINI	Babeş-Bolyai University Faculty of Mathematics and Computer Science, Cluj-Napoca ROMANIA E-mail: agratini@math.ubbcluj.ro
Kiril ALEXIEV	Institute of Information and Communication Technologies Mathematical Methods for Sensor Information Processing Sofia, 25A Acad. G. Bonchev Str. BULGARIA E-mail: alexiev@bas.bg
Radu BALAJ	Technical University of Cluj-Napoca Department of Computer Science Baritiu 28, RO-400391, Cluj-Napoca ROMANIA E-mail: radu.balaj@student.utcluj.ro
Alina BĂRBULESCU	Doctoral School of Civil Engineering Technical University of Civil Engineering, Bucharest ROMANIA E-mail: alinadumitriu@yahoo.com
Mircea Florian BOIAN	Faculty of Mathematics and Computer Science “Babes Bolyai” University Department of Computer Science 1 M. Kogalniceanu Street, Cluj Napoca ROMANIA E-mail: florin@cs.ubbcluj.ro
Calin BUCUR	“Lucian Blaga” University of Sibiu Doctoral School of Faculty of Economic Sciences No.17 Dumbravii Way, Sibiu ROMANIA E-mail: calin.bucur@yahoo.com
Claudia CARSTEA	George Baritiu University of Brasov Department of Mathematics and Informatics Str. Lunii 6, 500327, Brasov ROMANIA E-mail: claudia.carstea@universitateagbaritiu.ro
Stelian CIUREA	“Lucian Blaga” University of Sibiu Faculty of Engineering, Department of Computer and Electrical Engineering E. Cioran Str, No. 4, Sibiu-550025, ROMANIA, E-mail: stelian.ciurea@ulbsibiu.ro

Daniela DĂNCIULESCU
Department of Informatics
Faculty of Exact Sciences: Mathematics and Informatics
University of Craiova
A. I. Cuza Street, No. 13, 200585, Craiova, Dolj,
ROMANIA
E-mail: danadanciulescu@gmail.com

Nicoleta ENACHE-DAVID
Transilvania University of Brasov
Department of Mathematics and Computer Science
B-dul Eroilor 29, 500036, Brasov
ROMANIA
E-mail: nicoleta.enache@unitbv.ro

Dan Stelian DEAC
Vasile Goldiș Western University Arad
Department of Informatics
B-dul Revoluției Nr.85-86, 310025, Arad,
ROMANIA
E-mail: dndeac@gmail.com

Mykhailo DOROKHOV
Kharkiv National University of Economics
Department of Informations Systems,
Kharkiv, pr. Lenina, 9a, 61001
UKRAINE
E-mail: michaeldorokhov@gmail.com

Iryna GARBUZ
Kharkiv National University of Economics
Department of Informations Systems,
Kharkiv, pr. Lenina, 9a, 61001
UKRAINE
E-mail: kafis@hneu.edu.ua

Adrian GROZA
Technical University of Cluj-Napoca
Department of Computer Science
Baritiu 28, RO-400391 Cluj-Napoca,
ROMANIA
E-mail: adrian@cs-gw.utcluj.ro

Daniel HUNYADI
"Lucian Blaga" University of Sibiu
Faculty of Sciences
Dr. Ioan Rațiu St. No. 5-7, Sibiu
ROMANIA
E-mail: daniel.hunyadi@ulbsibiu.ro

George MANIU
"Lucian Blaga" University of Sibiu
Faculty of Sciences
Dr. Ioan Rațiu St. No. 5-7, Sibiu
ROMANIA
E-mail: costelmaniu@yahoo.com

Ionela MANIU
"Lucian Blaga" University of Sibiu
Faculty of Sciences
Dr. Ioan Rațiu St. No. 5-7, Sibiu
ROMANIA
E-mail: ionela.maniu@yahoo.ro

Marius MARIN
ROMANIA
E-mail: marin_marius_89@yahoo.ro

Ioana MOISIL
"Lucian Blaga" University of Sibiu
"Hermann Oberth" Faculty of Engineering
Department of Computer Science and Electrical Engineering
Emil Cioran Street, No. 4, 550025, Sibiu
ROMANIA
E-mail: im25sibiu@gmail.com

Vasile MORARU
Technical University of Moldova
Applied Informatics Department
168, Stefan cel Mare Str., Chisinau, 2004
Republic of MOLDOVA
E-mail: moraru@mail.utm.md

Mircea MUȘAN
"Lucian Blaga" University of Sibiu
Faculty of Sciences
Department of Mathematics and Informatics
Dr. I. Ratiu Street, No. 5-7, 550012, Sibiu
ROMANIA
E-mail: musanmircea@yahoo.com

Iryna OLOTARYOVA
Kharkiv National University of Economics
Department of Informations Systems,
Kharkiv, pr. Lenina, 9a, 61001
UKRAINE
E-mail: izolotaryova@gmail.com

Emanuela PETREANU
Transilvania University of Brasov
Department of Mathematics and Computer Science
B-dul Eroilor 29, 500036, Brasov
ROMANIA
E-mail: emanuela.petreanu@gmail.com

Alina Elena PITIC
"Lucian Blaga" University of Sibiu
Faculty of Sciences
Department of Mathematics and Informatics
Dr. I. Ratiu Street, No. 5-7, 550012, Sibiu
ROMANIA
E-mail: alinap29@yahoo.com

Alexandra-Mihaela POP
University "Lucian Blaga" of Sibiu
Faculty of Engineering
Department of Industrial Engineering and Management
Emil Cioran Streer, No. 4, 550025, Sibiu
ROMANIA
E-mail: alexandrapop_6@yahoo.com

Ioan POP
"Lucian Blaga" University of Sibiu
Faculty of Sciences
Department of Mathematics and Informatics
Dr. I. Ratiu Street, No. 5-7, **550012,** Sibiu
ROMANIA
E-mail: ioan.pop@ulbsibiu.ro

Anca RALESCU
University of Cincinnati
UNITED STATES
Email: anca.ralescu@uc.edu

Ahmad RAWASHDEH
University of Cincinnati
UNITED STATES
Email: rawashay@mail.uc.edu

Livia SANGEORZAN
Transilvania University of Brasov
Department of Mathematics and Computer Science
B-dul Eroilor 29, 500036, Brasov
ROMANIA
E-mail: sangeorzan@unitbv.ro

Klaus Bruno SCHEBESCH
Vasile Goldiș, Western University Arad
Department of Informatics
B-dul Revoluției Nr.85-86, 310025 Arad,
ROMANIA
E-mail: kbschebesch@uvvg.ro

Dana SIMIAN
"Lucian Blaga" University of Sibiu
Faculty of Sciences
Department of Mathematics and Informatics
Dr. I. Ratiu Street, No. 5-7, **550012,** Sibiu
ROMANIA
E-mail: d_simian@yahoo.com

Laura Florentina STOICA
"Lucian Blaga" University of Sibiu
Faculty of Sciences
Department of Mathematics and Informatics
Dr. I. Ratiu Street, No. 5-7, 550012, Sibiu
ROMANIA
E-mail: laura.cacovean@ulbsibiu.ro

Florin STOICA
"Lucian Blaga" University of Sibiu
Faculty of Sciences
Department of Mathematics and Informatics
Dr. I. Ratiu Street, No. 5-7, 550012, Sibiu
ROMANIA
E-mail: florin.stoica@ulbsibiu.ro

Saddika TARABIE
Tishrin University
Faculty of Sciences, Latakia
SYRIA
E-mail: sadikatorbey@yahoo.com

Radu TRÎMBIȚAȘ
Babeș-Bolyai University
Faculty of Mathematics and Computer Science, Cluj-Napoca
ROMANIA
E-mail: tradu@math.ubbcluj.ro

Anca VASILESCU
Transilvania University of Bras¸ov
Department of Mathematics and Computer Science
Iuliu Maniu Street 50, 500091 Braşov
ROMANIA
E-mail: vasilex@unitbv.ro

Letiţia VELCESCU
University of Bucharest
Department of Informatics
14, Academiei, 010014, Bucharest
ROMANIA
E-mail: letitia@fmi.unibuc.ro

Sergiu ZAPOROJAN
Technical University of Moldova
Computer Science Department
168, Stefan cel Mare Str., Chisinau, 2004
Republic of MOLDOVA
E-mail: zaporojan_s@yahoo.com

ISSN 2067-3965