Zbornik 17. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2014

Zvezek A

Proceedings of the 17th International Multiconference

INFORMATION SOCIETY – IS 2014

Volume A

Inteligentni sistemi
Intelligent Systems

Uredila / Edited by
Rok Piltaver, Matjaž Gams

http://is.ijs.si

7.−8. oktober 2014 / October 7th−8th, 2014
Ljubljana, Slovenia

Zbornik 17. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2014

Zvezek A

Proceedings of the 17th International Multiconference

INFORMATION SOCIETY – IS 2014

Volume A

Inteligentni sistemi

Intelligent Systems

Uredila / Edited by

Rok Piltaver, Matjaž Gams

http://is.ijs.si

7. - 8. oktober 2014 / October 7th - 8th, 2014
Ljubljana, Slovenia

Urednika:

Rok Piltaver
Odsek za inteligentne sisteme
Institut »Jožef Stefan«, Ljubljana

Matjaž Gams
Odsek za inteligentne sisteme
Institut »Jožef Stefan«, Ljubljana

Založnik: Institut »Jožef Stefan«, Ljubljana
Priprava zbornika: Mitja Lasič, Vesna Lasič, Lana Zemljak
Oblikovanje naslovnice: Vesna Lasič, Mitja Lasič

Ljubljana, oktober 2014

CIP - Kataložni zapis o publikaciji
Narodna in univerzitetna knjižnica, Ljubljana

004.89(082)(0.034.2)

MEDNARODNA multikonferenca Informacijska družba (17 ; 2014 ; Ljubljana)
Inteligentni sistemi [Elektronski vir] : zbornik 17. mednarodne multikonference - IS 2014, 7-8 oktober 2014, Ljubljana, Slovenija : zvezek A = Intelligent systems : proceedings of the 17th International Multiconference Information Society - IS 2014, October 7th-8th, 2014, Ljubljana, Slovenia : volume A / uredila/edited by Rok Piltaver, Matjaž Gams. - El. knjiga. - Ljubljana : Institut Jožef Stefan, 2014

Način dostopa (URL): http://library.ijs.si/Stacks/Proceedings/InformationSociety

ISBN 978-961-264-071-2 (pdf)
1. Gl. stv. nasl. 2. Vzp. stv. nasl. 3. Dodat. nasl. 4. Piltaver, Rok
27986727

PREDGOVOR MULTIKONFERENCI INFORMACIJSKA DRUŽBA 2014

Multikonferenca Informacijska družba (http://is.ijs.si) s sedemnajsto zaporedno prireditvijo postaja tradicionalna kvalitetna srednjeevropska konferenca na področju informacijske družbe, računalništva in informatike. Informacijska družba, znanje in umetna inteligenca se razvijajo čedalje hitreje. Čedalje več pokazateljev kaže, da prehajamo v naslednje civilizacijsko obdobje. Npr. v nekaterih državah je dovoljena samostojna vožnja inteligentnih avtomobilov, na trgu pa je moč dobiti kar nekaj pogosto prodajanih tipov avtomobilov z avtonomnimi funkcijami kot »lane assist«. Hkrati pa so konflikti sodobne družbe čedalje bolj nerazumljivi.

Letos smo v multikonferenco povezali dvanajst odličnih neodvisnih konferenc in delavnic. Predstavljenih bo okoli 200 referatov, prireditev bodo spremljale okrogle mize, razprave ter posebni dogodki kot svečana podelitev nagrad. Referati so objavljeni v zbornikih multikonference, izbrani prispevki bodo izšli tudi v posebnih številkah dveh znanstvenih revij, od katerih je ena Informatica, ki se ponaša s 37-letno tradicijo odlične evropske znanstvene revije.

Multikonferenco Informacijska družba 2014 sestavljajo naslednje samostojne konference:

- Inteligentni sistemi
- Izkopavanje znanja in podatkovna skladišča
- Sodelovanje, programska oprema in storitve v informacijski družbi
- Soočanje z demografskimi izzivi
- Vzgoja in izobraževanje v informacijski družbi
- Kognitivna znanost
- Robotika
- Jezikovne tehnologije
- Interakcija človek-računalnik v informacijski družbi
- Prva študentska konferenca s področja računalništva
- Okolijska ergonomija in fiziologija
- Delavnica Chiron.

Soorganizatorji in podporniki konference so različne raziskovalne in pedagoške institucije in združenja, med njimi tudi ACM Slovenija, SLAIS in IAS. V imenu organizatorjev konference se želimo posebej zahvaliti udeležencem za njihove dragocene prispevke in priložnost, da z nami delijo svoje izkušnje o informacijski družbi. Zahvaljujemo se tudi recenzentom za njihovo pomoč pri recenziranju.

V 2014 bomo drugič podelili nagrado za življenjske dosežke v čast Donalda Michija in Alana Turinga. Nagrado Michie-Turing za izjemen življenjski prispevek k razvoju in promociji informacijske družbe je prejel prof. dr. Janez Grad. Priznanje za dosežek leta je pripadlo dr. Janezu Demšarju. V letu 2014 četrtič podeljujemo nagrado »informacijska limona« in »informacijska jagoda« za najbolj (ne)uspešne poteze v zvezi z informacijsko družbo. Limono je dobila nerodna izvedba piškotkov, jagodo pa Google Street view, ker je končno posnel Slovenijo. Čestitke nagrajencem!

Niko Zimic, predsednik programskega odbora
Matjaž Gams, predsednik organizacijskega odbora

FOREWORD - INFORMATION SOCIETY 2014

The Information Society Multiconference (http://is.ijs.si) has become one of the traditional leading conferences in Central Europe devoted to information society. In its 17th year, we deliver a broad range of topics in the open academic environment fostering new ideas which makes our event unique among similar conferences, promoting key visions in interactive, innovative ways. As knowledge progresses even faster, it seems that we are indeed approaching a new civilization era. For example, several countries allow autonomous card driving, and several car models enable autonomous functions such as "lane assist". At the same time, however, it is hard to understand growing conflicts in the human civilization.

The Multiconference is running in parallel sessions with 200 presentations of scientific papers, presented in twelve independent events. The papers are published in the Web conference proceedings, and a selection of them in special issues of two journals. One of them is Informatica with its 37 years of tradition in excellent research publications.

The Information Society 2014 Multiconference consists of the following conferences and workshops:

- Intelligent Systems
- Cognitive Science
- Data Mining and Data Warehouses
- Collaboration, Software and Services in Information Society
- Demographic Challenges
- Robotics
- Language Technologies
- Human-Computer Interaction in Information Society
- Education in Information Society
- 1st Student Computer Science Research Conference
- Environmental Ergonomics and Psysiology
- Chiron Workshop.

The Multiconference is co-organized and supported by several major research institutions and societies, among them ACM Slovenia, SLAIS and IAS.

In 2014, the award for life-long outstanding contributions was delivered in memory of Donald Michie and Alan Turing for a second consecutive year. The Programme and Organizing Committees decided to award the Prof. Dr. Janez Grad with the Michie-Turing Award. In addition, a reward for current achievements was pronounced to Prof. Dr. Janez Demšar. The information strawberry is pronounced to Google street view for incorporating Slovenia, while the information lemon goes to cookies for awkward introduction. Congratulations!

On behalf of the conference organizers we would like to thank all participants for their valuable contribution and their interest in this event, and particularly the reviewers for their thorough reviews.

Niko Zimic, Programme Committee Chair
Matjaž Gams, Organizing Committee Chair

KONFERENČNI ODBORI
CONFERENCE COMMITTEES

International Programme Committee

Vladimir Bajic, South Africa
Heiner Benking, Germany
Se Woo Cheon, Korea
Howie Firth, UK
Olga S. Fomichova, Russia
Vladimir A. Fomichov, Russia
Vesna Hljuz Dobric, Croatia
Alfred Inselberg, Izrael
Jay Liebowitz, USA
Huan Liu, Singapore
Henz Martin, Germany
Marcin Paprzycki, USA
Karl Pribram, USA
Claude Sammut, Australia
Jiri Wiedermann, Czech Republic
Xindong Wu, USA
Yiming Ye, USA
Ning Zhong, USA
Wray Buntine, Finland
Bezalel Gavish, USA
Gal A. Kaminka, Israel
Mike Bain, Australia
Michela Milano, Italy
Derong Liu, Chicago, USA
Toby Walsh, Australia

Organizing Committee

Matjaž Gams, chair
Mitja Luštrek
Lana Zemljak
Vesna Koricki-Špetič
Mitja Lasič
Robert Blatnik
Mario Konecki
Vedrana Vidulin

Programme Committee

Nikolaj Zimic, chair
Franc Solina, co-chair
Viljan Mahnič, co-chair
Cene Bavec, co-chair
Tomaž Kalin, co-chair
Jozsef Györkös, co-chair
Tadej Bajd
Jaroslav Berce
Mojca Bernik
Marko Bohanec
Ivan Bratko
Andrej Brodnik
Dušan Caf
Saša Divjak
Tomaž Erjavec
Bogdan Filipič
Andrej Gams
Matjaž Gams
Marko Grobelnik
Nikola Guid
Marjan Heričko
Borka Jerman Blažič Džonova
Gorazd Kandus
Urban Kordeš
Marjan Krisper
Andrej Kuščer
Jadran Lenarčič
Borut Likar
Janez Malačič
Olga Markič
Dunja Mladenič
Franc Novak
Vladislav Rajkovič
Grega Repovš
Ivan Rozman
Niko Schlamberger
Stanko Strmčnik
Jurij Šilc
Jurij Tasič
Denis Trček
Andrej Ule
Tanja Urbančič
Boštjan Vilfan
Baldomir Zajc
Blaž Zupan
Boris Žemva
Leon Žlajpah
Igor Mekjavić
Tadej Debevec

KAZALO / TABLE OF CONTENTS

Zbornik 17. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2014

Zvezek A

Proceedings of the 17th International Multiconference

INFORMATION SOCIETY – IS 2014

Volume A

Inteligentni sistemi

Intelligent Systems

Uredila / Edited by

Rok Piltaver, Matjaž Gams

http://is.ijs.si

7. - 8. oktober 2014 / October 7th - 8th, 2014
Ljubljana, Slovenia

PREDGOVOR

Konferenca Inteligentni sistemi je od leta 1997 naprej sestavni del multikonference Informacijska družba. Poglavitne teme so inteligentni sistemi in inteligentne storitve informacijske družbe, oz. programski sistemi informacijske družbe, tehnične rešitve v inteligentnih sistemih, možnosti njihove praktične uporabe, pa tudi trendi, perspektive, nujni ukrepi, prednosti in slabosti, priložnosti in nevarnosti, ki jih v informacijsko družbo prinašajo inteligentni sistemi.

V 2014 se nesluten razvoj informacijske družbe in zlasti umetne inteligence nadaljuje s čedalje hitrejšim tempom. V nekaterih državah po svetu že vozijo avtonomni avtomobili, npr. »Google car«. Nekoč utopične ideje Raya Kurtzweila o točki singularnosti in preskoku v novo človeško ero, se tako zdijo čedalje bližje. Hkrati pa se razlike med ljudmi povečujejo in nihče prav dobro ne razume družbenih sprememb, ki smo jim priča.

Tudi letos konferenca Inteligentni sistemi sestoji iz mednarodnega dela in delavnice; prispevki so tako v slovenskem kot angleškem jeziku. Sprejetih je več kot 25 prispevkov, ki so bili recenzirani s strani vsaj dveh anonimnih recenzentov, avtorji pa so jih popravili po navodilih recenzentov. Večina prispevkov obravnava raziskovalne dosežke Odseka za inteligentne sisteme Instituta »Jožef Stefan«. Hkrati s predstavitvijo poteka tudi aktivna analiza prispevkov vsakega predavatelja in diskusija o bodočih raziskavah.

Rok Piltaver in Matjaž Gams, predsednika konference

PREFACE

The Intelligent Systems conference remains one of the fundamental parts of the multiconference Information Society since its beginnings in 1997. The conference addresses important aspects of information society: intelligent computer-based systems and the corresponding intelligent services, technical aspects of intelligent systems, their practical applications, as well as trends, perspectives, advantage and disadvantages, opportunities and threats that are being brought by intelligent systems into the information society.

As a trend, the progress in information society and intelligent systems increases further in recent years. For example, some countries already enacted autonomous car driving. Once regarded as utopist, the ideas of Ray Kurtzweil that the human civilization will embrace a new, intelligent era, are becoming widely accepted. At the same time, it seems that nobody fully understands the emerging changes in human society.

The conference consists of an international event and a workshop, and presents over 25 papers written in both English and Slovenian language. The papers have been reviewed by at least two anonymous reviewers and the authors have modified their papers according to the remarks. Papers from the Jozef Stefan Institute - Department of Intelligent Systems are presented separately. Each presentation consists of the classical paper report, and further includes analysis of researcher's achievements and future research plans of each presenter.

Rok Piltaver and Matjaž Gams, Conference Chairs

MULTIOBJECTIVE OPTIMISATION OF WATER HEATER SCHEDULING

Jure Brence[1*], *Žiga Gosar*[1*], *Vid Seražin*[1*], *Jernej Zupančič*[2], *Matjaž Gams*[2]

[1]Faculty of mathematics and physics, University of Ljubljana, Jadranska ulica 19, 1000 Ljubljana

[2]Department of Intelligent Systems, Jozef Stefan Institute, Jamova cesta 39, 1000 Ljubljana
e-mail: {jure.brence, ziga.gosar, vid.serazin}@student.fmf.uni-lj.si, {jernej.zupancic, matjaz.gams}@ijs.si

ABSTRACT

In this paper we present our work on the optimisation of water heater scheduling. The goal is to develop intelligent strategies for controlling the electric heater and heat pump in commercial combined water heaters. Strategies try to find the best compromise between comfort and price, based only on information about the temperature of water in the reservoir. A simulation and testing environment has been implemented to compare the performance of existing and new strategies.

1 INTRODUCTION

Hot water heating is the biggest component of electricity consumption in residential homes, contributing as much as 20% to the total electricity consumption in an average Slovenian household [11]. Water heater manufacturers continually develop improvements to the mechanical aspects of water heating. However, the potential for savings by smarter power scheduling is quite unexplored. Most water heater controllers tend to keep water temperature at pre-set levels throughout the day, with the exception of user-defined schedules. This results in increased heat loss and, more importantly, bigger loads on the power grid during peak hours. An intelligent controller would be able to find and optimised schedule of water heating, customised for the habits and wishes of users. It is important not only to minimise the price of heating, but to do so with a minimal increase in user discomfort level.

2 RELATED WORK

Some research on the topic of electric water heaters has already been done. All stated sources are dealing with devices using only an electric heater, whereas our research focuses on combined devices. Much of existing work perceives user discomfort as a constraint, rarely incorporating it as one of the objectives.

In [1] solutions are provided for an electric water heater that is connected to an electrical grid where the electricity tariff is dynamically changed in real time, and mainly focuses on optimisation in regard to this tariff system. In [2] a model is presented, which addresses the extraction of household water usage patterns with the goal of peak-shaving and reducing the load on the power-grid. In [3] similar goals are addressed, while approaching the problem from a different angle, utilising fuzzy logic to control electric water heaters. Similarly, in [4] the focus is on a solution that decreases peak load on the grid by scheduling heating outside peak hours. In [5] a simulation platform to model electric water heaters and test demand response control strategies in a smart grid is introduced.

3 THE PROBLEM

The aim is to develop intelligent strategies for the scheduling of water heating. There are several types of water heaters on the market, the difference being their source of energy. The most interesting are combined water heaters that have both an electric heater and a heat pump at its disposal. The control unit of a combined water heater is able to control the different heaters separately. At any given moment the controller decides whether a heater is to be turned off or on. Water heaters typically have a single thermometer installed, usually on the top of the water reservoir. This measurement is the only information an intelligent controller gets about the state of the water in the reservoir and the consumption habits of the users.

The development of intelligent strategies is a multi-objective optimisation problem. The first objective is the electricity cost and the second is some measure of discomfort of the users. Any strategy will have to be a trade-off between the two. Our solution will be a set of strategies, among which the user will be able to choose the one with the desired trade-off between price and comfort.

4 STRATEGIES

We have implemented a number of different strategies. Each falls into one of two categories that differ by the information that is available to the controller.

The simplest strategies are static strategies that use only predefined settings and current measurements. These could be date, time, temperature and the temperature in the previous minute. Static strategies follow a predefined set of rules. While they do not learn or modify their behaviour, different

[*] These authors equally contributed to the paper.

rule-sets may be defined for different periods of the day, or days in the week. Some static strategies:

1. **On-Off Control** (*lower T, upper T, electric heater, heat pump*) is the strategy used in most commercial water heaters. Sometimes called Bang-Bang Control. The boolean constants *electric heater* and *heat pump* specify if the strategy is allowed to use electric heater and heat pump. When the temperature drops below *lowerT* all available heat sources are turned on until *upperT* temperature is reached.

2. **Intervals** (*list of intervals with appropriate strategies*) uses different strategies in different parts of the day (e.g. when electricity is cheaper or when the user expects higher water consumption). At initialisation we can specify any number of intervals and corresponding strategies. One example is a sub-strategy called Heat Less at Noon which uses On-Off Control(40, 41, False, True) between 9 am and 3 pm and On-Off Control(45, 50, False, True) at other times. This strategy is similar up to constants values to some real strategies consumers use.

3. **New On-Off** (*lower boundary for electric heater, lower boundary for heat pump*). When the temperature drops below the predefined *lower boundary for electric heater* the electric heater turns on until the temperature is higher than this boundary. The heat pump works on the same principle but with a different boundary temperature, which is usually higher.

4. **Rules Z**. A day is divided in N regions that are set by the user. In each region a set of boundary temperatures, as well as boundary temperature changes is defined. The two different heaters are turned on or off based on the boundary conditions for the current region.

Oracle strategies are given the future water consumption schedule that they use to calculate the plan of how and when they will heat the water. We use these strategies to get the best trade-off between discomfort and price. There is no other strategy with a strictly better performance in both objectives.

1. **Brute Force** makes decisions at discrete time intervals of predefined length, usually 1 or 10 minutes. At every step four options are available: no heating, only electric heater, only heat pump, or both. Brute force simulates every possibility, looking for the optimal one. Theoretically every possible strategy would be tested by Brute Force, allowing us to find the true Pareto front. This approach is not practical due to its computational inefficiency.

2. **Bulk** starts with a decision to never heat. It simulates the water heater until it reaches discomfort. Then it starts rewinding time back and turning the heat pump on until the discomfort reaches zero. If this cannot be achieved with the heat pump alone, Bulk begins utilising the electric water heater. This way, any heating is done directly before water consumption. It can also be modified to heat during the lower price tariff to accumulate heat. This way Bulk produces a result with minimal discomfort at an almost minimal price.

There is also a third category of strategies that learn from the past and adjust their decision-making to best fit the user habits. This kind of strategies are the final goal of our research.

5 METHODS

The basic method applied in this research is the testing and comparison of various scheduling strategies. We utilise computer simulations, as running these tests on real water heaters would require a lot of time and resources, which we do not have at our disposal. To this purpose, we have developed a water heater simulator and a water consumption simulator.

5.1 Water heater simulation

Real specifications [9, 10] of commercial water heaters were used, namely: dimensions of the reservoir, power of heaters, coefficient of performance (COP) of the heat pump, thermal conductivity of the insulation and maximum flow rate. Typical water heaters are shaped cylindrically, with cold water entering the reservoir at the bottom and hot water leaving on the top. The position of the heating element varies with the model. Some manufacturers choose to position the heater at the bottom, to encourage the convection of hot water, others attempt to heat uniformly along the vertical axis, or some other option. In current tests water is heated uniformly. Combined water heaters have two types of heaters: electric heater and heat pump. With the electric heater, the thermal power it produces is equal or close to equal to the electric power it consumes. As such, its heating power is fixed. The heat pump, on the other hand, produces more thermal energy than the amount of electric energy it uses. The ratio between the two – COP – typically falls into a range from 2 to 5.5. The COP of a heat pump depends on the temperature of the heat source, often the outside air, and the temperature of water. During the heating process, as water temperature increases, the COP drops.

The simulation does not attempt to simulate the complex thermodynamics and fluid mechanics happening in the water heater. It rather uses a simplified model that manages to emulate the responses of the built-in thermometer to various inputs. The water in the reservoir is divided into 20 layers along the vertical axis. All the water in one layer has the same temperature. The water heater is simulated with a one minute step. Each step, energy losses are calculated for each layer, taking into account the water temperature, outside temperature and the thermal conductivity of the container walls. Heat exchange between neighbouring layers is simulated with an experimentally set heat transfer coefficient to match real data. When heating is turned on, each layer receives its share of thermal energy. As the simulator receives a request for hot water, it removes the appropriate volume of water from the top layers and adds cold water layers at the bottom. The number of layers and their individual size is kept in check by joining neighbouring layers with similar temperatures. Manufacturers usually take special care to minimise the mixing of water in the reservoir. This provides the consumer with a better experience that is, the outgoing water stays at the almost the same temperature during a shower, unless all hot water is used up. Our simulator is able to reproduce this behaviour to a sufficient degree.

5.2 Water consumption simulation

A number of sources [6, 7, 8] for water consumption measurement were used to develop a simulation of water consumption during weekdays and weekends. When a household with a specified number of members is generated, each individual is assigned a semi-random consumption pattern. A specific consumption schedule is then generated based on the patterns of individuals, with added variance using Gaussian distributions. We separate two types of events. 3 to 10 small events (e.g. washing hands) per user are randomly scattered throughout the day, taking one minute and using less than 1 litre of hot water. Large events (showers) happen 1 to 3 times per day per user, and take between 5 and 30 minutes, with 1-6 litres of water per minute at a mean 38°C.

Both real and simulated water consumptions vary greatly on a day to day basis. Figure 1 shows the distribution of simulated hot water consumption over a longer time period.

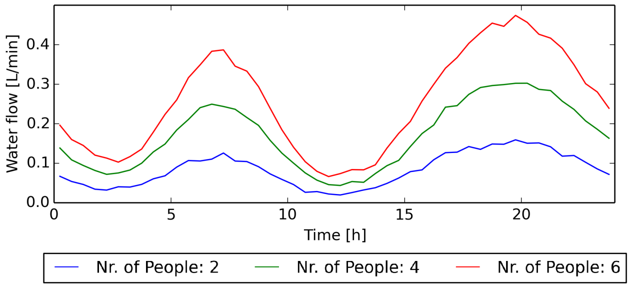


Figure 1: Simulated hot water consumption, averaged over 50 week days for 100 different households.

5.3 Discomfort and price

Different strategies under different water consumption profiles were evaluated using discomfort and price as criteria. Discomfort for a minute of our simulation is defined as:

$$discomfort = \begin{cases} 0 & \text{if } T_o \geq T_r \\ \frac{(T_r - T_o)*V}{1000} & \text{if } T_o < T_r \end{cases} \quad (1)$$

where T_r is the requested temperature by the user, T_o is the outflow temperature and V is the volume of water with the corresponding temperature. The total discomfort of a measurement is defined as the sum of individual discomforts.

The calculations of price have to take into account different price tariffs. The majority of Slovenian electricity providers use a two-tariff system, with the lower tariff from 22.00 to 6.00 and during weekends and the higher tariff from 6.00 to 22.00 during week days. The prices vary between suppliers. We use a lower tariff price of 0.04320 €/kWh and a higher tariff price of 0.07795 €/kWh [11].

6 TESTING AND RESULTS

At the beginning of the test, the generator of water consumption produces a semi-random plan of consumption. The water heater is simulated with a one minute step for the specified duration of the experiment, usually several weeks. Water is used according to the schedule, while the heating of the water heater is controlled by the tested strategy. The process is repeated for other strategies using the same consumption schedule. The whole experiment is ran multiple times with different consumption schedules. The result of the experiment are the average price and discomfort for each of the tested strategies (Figure 2).

As anticipated, Bulk achieves the best comfort, which is usually near zero. Other strategies with comparable comfort achieve it at a much higher price. A generally best performance is achieved by On-Off Control, using only the heat pump heater. Most of our static strategies are dominated by On-Off Control and Bulk.

Each strategy has a number of parameters that can be varied to achieve different results. By varying the boundaries of On-Off Control we produce three fronts, one for each type of heating (figure 2). Varying the parameters of Oracle strategies would produce another front. Ideal solutions would dominate On-Off Control – type strategies, while being dominated by Oracle strategies.

7 Conclusion

The project is aiming to develop intelligent strategies for the scheduling of water heating in commercial water heaters. So far we have developed a complete testing environment for comparing different strategies. We have implemented and tested most commercially used strategies. For comparison we have also implemented one Oracle strategy that achieves the best comfort possible.

In order to find a good approximation of the Pareto front for our consumption simulator, we intend to develop a set of Oracle strategies that are capable of achieving a specific tradeoff of price and comfort.

We also plan to utilise evolutionary algorithms to optimise the various static strategies. Finally, we want to develop intelligent strategies that adapt by learning from the past.

An immediate application of our system is to provide the user with a more intuitive way of choosing the most appropriate strategy. As it stands, users manually choose the On-Off

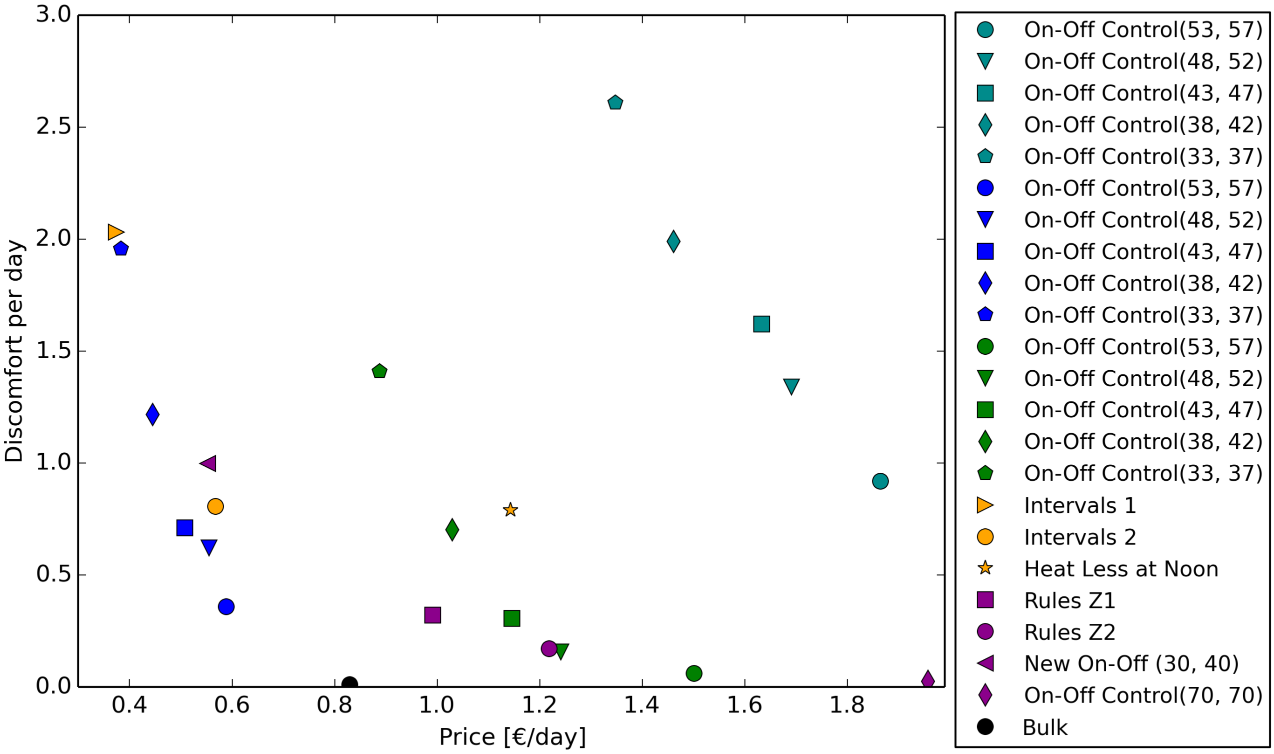


Figure 2: Averaged price and discomfort for a number of static strategies and Bulk. The colours dark cyan, blue and green represent static strategies using only the electric heater, only the heat pump and both, respectively. Strategies coloured orange are variations of Intervals. Other static strategies are coloured purple. Oracle strategies are black. Household with 4 members, using a 230 L water heater with a 1500 W electric heater and a heat pump with a heating power of 2000 W and a COP of 3.3 at 35°C.

Control settings, which is generally the preferred water temperature. With our simulation, users would only need to decide on a price to comfort trade-off, and the controller would choose the best strategy and settings to improve their comfort while lowering their costs.

References

[1] P. Du, N. Lu., Appliance Commitment for Household Load Scheduling. *Smart Grid, IEEE Transactions on Smart Grid* 2, no. 2 (2011): 411-419.

[2] L. Paull, H. Li, L. Chang. A novel domestic electric water heater model for a multi-objective demand side management program. *Electric Power Systems Research* 80, no. 12 (2010): 1446-1451.

[3] B. J. LaMeres, M. H. Nehrir, V. Gerez Controlling the average residential electric water heater power demand using fuzzy logic. *Electric Power Systems Research* 52, no. 3 (1999): 267-271.

[4] A. Moreau. Control Strategy for Domestic Water Heaters during Peak Periods and its Impact on the Demand for Electricity. *Energy Procedia* 12 (2011): 1074-1082.

[5] R. Diao, S. Lu, M. Elizondo, E. Mayhorn, Y. Zhang, N. Samaan. Electric Water Heater Modeling and Control Strategies for Demand Response. *In Power and Energy Society General Meeting, IEEE* (2012): 1-8.

[6] Energy Monitoring Company, Energy Saving Trust. Measurement of Domestic Hot Water Consumption in Dwellings (2008).

[7] B. Schoenbauer, D. Bohac, M. Hewett. Measured Residential Hot Water End Use. *ASHRAE Transactions* 118 (2012): 872-889.

[8] Equipment Energy Efficiency. Water Heating Data Collection and Analysis, 2012.

[9] Coolwex. DSW 300 - Navodila za inštalacijo in uporabo. *Klima center Horizont*, Maribor, 2010.

[10] Kronoterm. Katalog produktov, 2014. URL `http://www.kronoterm.com/wp-content/uploads/2014/katalog-40-mar-2014-web.pdf`. Acquired 20.8.2014.

[11] Statistični urad Republike Slovenije. Podatkovni portal SI-STAT, Okolje in naravni viri, Seznam tabel. URL `http://pxweb.stat.si/pxweb/Dialog/statfile2.asp`

ANALIZA NAKUPOV IN MODELIRANJE POSPEŠEVANJA PRODAJE V SPLETNI TRGOVINI

Matija Černe (Fakulteta za matematiko in fiziko,
Jadranska 19, 1000 Ljubljana, Slovenija),
Boštjan Kaluža, Mitja Luštrek
Odsek za inteligentne sisteme,
Inštitut Jožef Stefan
Jamova cesta 39, 1000 Ljubljana, Slovenija
Tel: +386 1 4773419; fax: +386 1 4251038
e-mail: matija.cerne@student.fmf.uni-lj.si
bostjan.kaluza@ijs.si
mitja.lustrek@ijs.si

POVZETEK

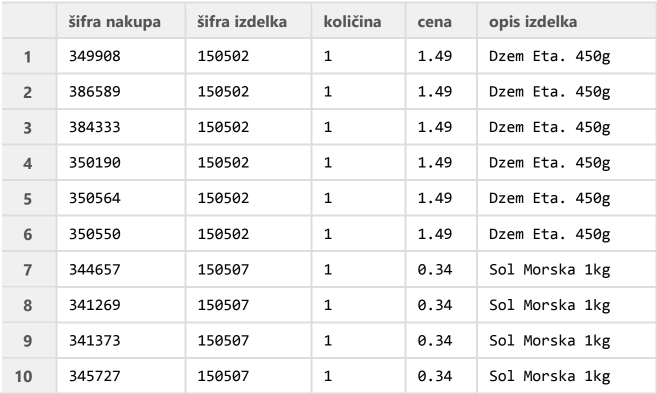
Analizirali smo podatke o nakupih v spletni trgovini. Cilja sta bila ugotoviti učinek spremembe cene na potrošnjo in identifikacija potrošnikovih preferenc v nekem trenutku. Pri analizi smo uporabljali tako mikroekonomske kot tudi statistične pristope. V grobem lahko metode analize razdelimo na dva sklopa – tiste, ki se osredotočajo na uporabnika in tiste, pri katerih je pomemben le artikel.

1 UVOD

Zanimajo so nas predvsem rezultati, ki bi jih lahko uporabili za priporočanje artiklov tako znanim (obstoječim), kot tudi neznanim (novim) uporabnikom. Pri priporočanju gre za to, da čimbolj natančno ugotovimo, kateri izdelek bi nekega uporabnika poleg kupljenih še utegnil zanimati, nato pa mu ta izdelek spletna prodajalna priporoči. Iz prodajalčevega vidika je to precej pomembno orodje pospeševanja prodaje, še posebej v kontekstu spletne trgovine. V nasprotju s klasično trgovino lahko tu v vsakem trenutku vidimo uporabnikovo košarico, pa tudi uporabnika samega lahko identificiramo, kar v praksi (za ne-uporabnike raznih kartic zvestobe) ni izvedljivo. Priporočila se na spletni strani izvedejo v obliki seznama priporočenih izdelkov, kar je izvedljivo v realnem času, če smo podatke predhodno pravilno obdelali.

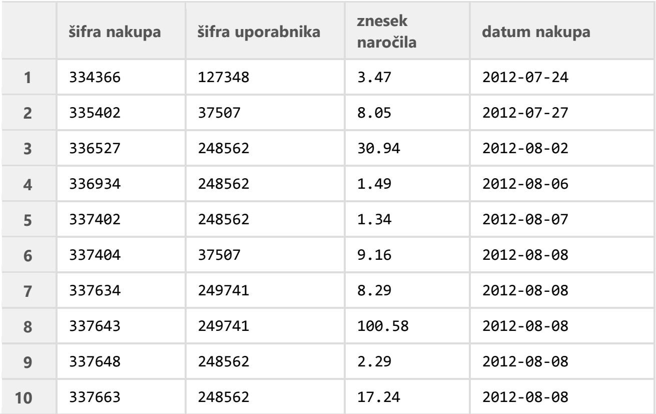
2 PODATKI

Na voljo smo imeli podatke o vseh nakupih, ki so se v spletni trgovini zgodili med 24. Julijem 2012 in 15. Januarjem 2014. Za posamezne izdelke tako vemo kdaj, koliko in po kakšni ceni so bili prodani. Ob kasnejši obdelavi smo sicer ugotovili, da obstaja možnost, da določeni podatki manjkajo (predvsem na začetku obdobja), vendar je to upoštevano v analizi oziroma pri rezultatih. Večina analize je temeljila na dveh datotekah, katerih izseka vidimo spodaj:



	šifra nakupa	šifra izdelka	količina	cena	opis izdelka
1	349908	150502	1	1.49	Dzem Eta. 450g
2	386589	150502	1	1.49	Dzem Eta. 450g
3	384333	150502	1	1.49	Dzem Eta. 450g
4	350190	150502	1	1.49	Dzem Eta. 450g
5	350564	150502	1	1.49	Dzem Eta. 450g
6	350550	150502	1	1.49	Dzem Eta. 450g
7	344657	150507	1	0.34	Sol Morska 1kg
8	341269	150507	1	0.34	Sol Morska 1kg
9	341373	150507	1	0.34	Sol Morska 1kg
10	345727	150507	1	0.34	Sol Morska 1kg

Podatki o nakupih (izsek). Št.vrstic: 15176



	šifra nakupa	šifra uporabnika	znesek naročila	datum nakupa
1	334366	127348	3.47	2012-07-24
2	335402	37507	8.05	2012-07-27
3	336527	248562	30.94	2012-08-02
4	336934	248562	1.49	2012-08-06
5	337402	248562	1.34	2012-08-07
6	337404	37507	9.16	2012-08-08
7	337634	249741	8.29	2012-08-08
8	337643	249741	100.58	2012-08-08
9	337648	248562	2.29	2012-08-08
10	337663	248562	17.24	2012-08-08

Podatki o prodanih izdelkih (izsek). Št. vrstic: 347332

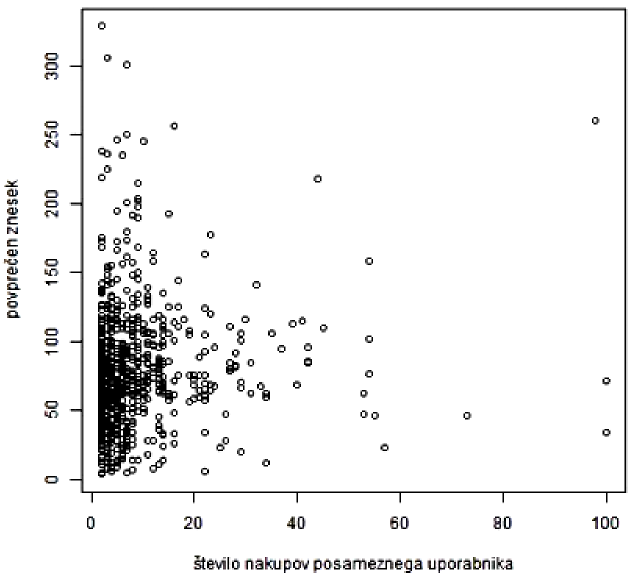
Poleg tega smo imeli tudi podatke o uporabnikih, ki so povedali ali je posamezen uporabnik fizična oseba ali podjetje, ter poštno številko njegovega prebivališča. Ker je cilj projekta priporočanje produktov fizičnim uporabnikom, smo se odločili da bomo obravnavali samo fizične osebe. Če bi obravnavali podatke obeh kategorij skupaj, bi zaradi velike razlike v obsegu potrošnje, pa tudi zaradi specifičnih potreb podjetnikov ki se navadno razlikujejo od potreb 'običajnih' potrošnikov, verjetno dobili precej nezanesljive rezultate. Zato smo najprej iz podatkov o nakupih izločili tiste, ki so jih opravila podjetja. Glede lokacije uporabnikov se nam v dosedanji analizi to ni zdel dovolj pomemben dejavnik pri potrošnikovih odločitvah in temu nismo posvečali posebne pozornosti.

Na voljo smo imeli še podatke o lastnostih izdelkov, vendar teh podatkov nismo obravnavali.

Ena od težav je bila ta, da nismo imeli točnih podatkov o datumih sprememb cen in smo tako datum spremembe morali aproksimirati z datumom, ko se je prvič zgodil nakup po novi ceni. To pa za izdelke, ki se ne kupujejo vsak dan (in takšnih je večina) pomeni, da se obdobja ko neka cena velja, lahko precej razlikujejo od resničnih obdobij. Ravno zaradi tega, pa tudi zaradi premajhne količine podatkov (kar bi imelo za posledico premalo zanesljive rezultate) smo se odločili, da v analizah, kjer je to pomembno, obravnavamo samo določeno število izdelkov, za katere imamo dovolj podatkov.

2.2 Vizualizacija podatkov

Graf prikazuje, kako so porazdeljeni uporabniki glede na število nakupov, ki jih opravijo (horizontalna os) in povprečno vrednost nakupa (vertikalna os). Vsaka pika predstavlja enega uporabnika:



SLIKA 1: *porazdelitev uporabnikov spletne trgovine*

Opazimo, da večina uporabnikov za svoj nakup zapravi okoli 70 eur, in v obravnavanem obdobju manj kot petnajstkrat nakupuje v spletni trgovini.

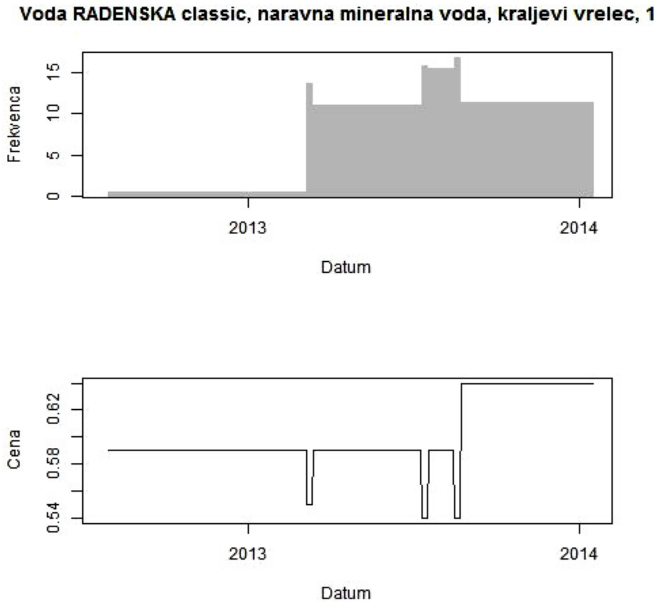
2.3 Obdelava podatkov

Za nadaljno uporabo je bilo potrebno združiti podatke o naročenih izdelkih in uporabnikih – torej pogledati, kateri 'order ID-ji' pripadajo kateremu uporabniku in nato za posamezno naročilo (order) združiti izdelke ki so bili kupljeni.

Za potrebe cenovne analize je bilo potrebno podatke transformirati v takšno obliko, da lahko razberemo informacijo o potrošnji ob določeni ceni. Natančneje, potrebovali smo neko mero za 'moč potrošnje' v določenem cenovnem obdobju (torej obdobju med dvema spremembama cene) in najbolj logična mera se je zdela frekvenca nakupov (enota: št.izdelkov/dan):

$$frekvenca = \frac{kolicina\ prodanih\ artiklov\ v\ obdobju}{dolzina\ obdobja}$$

Tu je sicer nastopil problem določitve obdobij ko velja neka cena, saj kot smo že prej omenili, nimamo točnih datumov sprememb. Problematični so bili predvsem primeri, ko se je nek nakup zgodil po ceni pred spremembo, vendar je bil zabeležen datum komaj v obdobju, ko je veljala naslednja cena – tako se je večkrat zgodilo tudi, da smo imeli na isti dan iste artikle prodane po različnih cenah. Možna razlaga za to je, da se je cena izdelka zabeležila ob izdaji računa, datum nakupa pa je obveljal kot datum plačila – ni namreč nujno, da je bil račun takoj plačan. Kakorkoli, v takšnih primerih je bilo potrebno 'izravnati šum' in naročila s staro ceno postaviti v prejšnje obdobje, sicer bi imeli ob nekaterih spremembah cene lahko hude distorzije v frekvenci nakupov. To je bilo (za nekatere obravnavane izdelke) narejeno kar ročno, saj bi bilo sicer pretežko dovolj dobro definirati, katerim naročilom je potrebno spremeniti datum.



SLIKA 2: *Graf frekvenc nakupov za izdelek 'Mineralna voda Radenska classic, kombiniran z grafom cen*

3. ANALIZA - METODE IN REZULTATI

Najprej smo analizirali potrošnjo v odvisnosti od cen (cenovna analiza), nato pa še analizirali nakupovalne navade desetih najbolj zanimivih uporabnikov.

3.1 Cenovna analiza

Pri cenovni analizi raziskujemo, kako se potrošnja (frekvenca nakupov) spreminja v odvisnosti od spremembe v ceni. Z uporabo mikroekonomskega pojma cenovne elastičnosti smo poskušali oceniti vpliv spremembe cene na potrošnjo istega oziroma sorodnih izdelkov. Nato nas zanima tudi, kaj se dogaja ob specifični kratkotrajni spremembi cene – akciji.

3.1.2 Cenovna elastičnost

Za učinek sprememb cene na potrošnjo istega izdelka cenovno elastičnost izračunamo tako:

$$\epsilon = \frac{\Delta Q}{\Delta p} \frac{p_1 + p_2}{q_1 + q_2}$$

Za učinek spremembe cene nekega izdelka na potrošnjo nekega drugega izdelka (substituta) potrebujemo koeficient križne elastičnosti. Ta nam pove, za koliko odstotkov se spremeni potrošnja dobrine B ob spremembi cene dobrine A za en odsototek:

$$\epsilon_{A,B} = \frac{\Delta Q_B}{\Delta p_A} \frac{p_{1,A} + p_{2,A}}{q_{1,B} + q_{2,B}}$$

V prvem primeru pričakujemo, da bo vrednost negativna (če se cena poveča, se troši manj nekega izdelka), posledično pa pri sorodnih izdelkih (komplementih) pričakujemo, da se bo potrošnja ob nespremenjeni ceni povečala. Za potrebe računanja križnih elastičnosti je bilo potrebno še enkrat naračunati frekvence nakupov (q), tokrat po datumih sprememb cene vseh ostalih izdelkov, ki jih opazujemo skupaj. Spremembe se seveda ne zgodijo samo enkrat, zato ob vsaki spremembi cene lahko izračunamo novo cenovno elastičnost (tako enostavno kot križno). Rezultate nato lahko aproksimiramo regresijsko ali pa izračunamo povprečje. Zaradi velike distorziranosti podatkov se je druga metoda izkazala za bolj primerno.

V naslednji tabeli so predstavljeni rezultati (povprečne elastičnosti) za skupino substitutov 'Mleka', kjer so bili rezultati še najbolj skladni s pričakovanji:

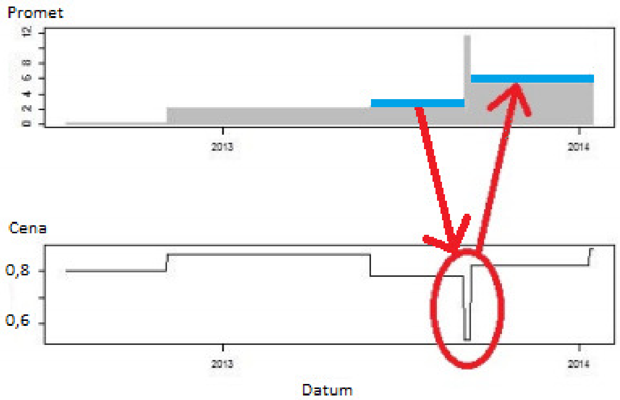
v %	Zelene doline 3,5	Alpsko 1,6	Alpsko 3,5	Zelene doline 1,6
Zelene doline 3,5	-0,16	0,29	0,23	0,2
Alpsko 1,6	1,19	0,4	0,47	0,47
Alpsko 3,5	0,6	0,39	2,42	0,28
Zelene doline 1,6	1,36	1,1	1,06	-1,2

Po diagonali so vrednosti izračunane po prvi formuli, na ostalih mestih pa po principu križne elastičnosti. Razlaga vrednosti v tabeli (gledamo zadnjo vrstico). Če se cena mleka znamke 1,6 poveča za 1%, tedaj se potrošnja (število prodanih artiklov na dan) mlek 3,5, Alpsko 1,6 in Alpsko 3,5 po vrsti poveča za 1,36 %, 1,1 %, in 1,06 %. Obenem se potrošnja mleka zmanjša za 1,02 %.

Še vedno seveda ne moremo z gotovostjo trditi, da se bo potrošnja spreminjala točno tako kot so vrednosti v tabeli, vendar vseeno vsaj okvirno vidimo, kateri izdelki so bolj, kateri pa manj občutljivi na spremembe v ceni (v tabeli izgleda da je Alpsko 3,5 najbolj, 1,6 pa najmanj cenovno stabilno). Vseeno ta metoda ni najbolj zanesljiva za napovedovanje potrošnje, saj lahko predvidevamo, da so spremembe odvisne tudi od številnih drugih dejavnikov (npr oglaševanje, substituti iz drugih trgovin, substituti ki jih nismo upoštevali pri analizi, šum na podatkih, …). V tabeli je to najbolj vidno pri Alpskem mleku 3,5, kjer izgleda da je 1% spremembe v ceni prinesel 2,42% spremembe (pozitivne!) v potrošnji. Ko pogledamo na graf frekvenc, pa opazimo, da je potrošnja dobrega pol leta od začetka merjenja skoraj nič, torej se lahko upravičeno vprašamo, ali je to res (kar je malo verjetno, glede na to da gre za enega najbolj prodajanih artiklov za katerega dobro vemo, da ni prišel v prodajo komaj pred enim letom) in če se je mogoče zgodila napaka pri knjiženju naročil – recimo če se je vmes zamenjala koda izdelka in to ni bilo popravljeno v bazi podatkov.

3.1.2 Analiza učinkov akcij

Ob opazovanju grafov prometa (prihodki od prodaje na dan; cena krat frekvenca) in cen v času za nekatere izdelke opazimo, da za kratkotrajne padce v ceni (akcije) frekvenca potrošnje za to obdobje naraste, kar seveda ni presenetljivo. Zanimivo pa je dejstvo, da se velikokrat potrošnja po koncu akcije (vrnitvi cene na isto ali višjo raven kot prej) ne vrne na raven pred akcijo, temveč ostane višja kot je bila tedaj. Ob tem velja poudariti, da na potrošnjo poleg samega znižanja cene gotovo vpliva tudi to, da se ob akciji tudi poveča promocija za izdelek (npr. objava v katalogu, reklama po televiziji). Poglejmo ta efekt na grafu za 'Mleko Lejko 1,5%' :



SLIKA 3: *Padec cene (akcija) je označen z rdečo elipso, promet pred in po akciji pa z modrima črtama*

Pri obravnavanih izdelkih (100 artiklov z najvišjo prodajo) se je ta situacija večkrat ponovila. Za definirano akcijo (v našem primeru je akcija definirana kot vsaka sprememba cene za vsaj -4% in v trajanju največ tri tedne) smo izsledke predstavili v tabeli, kjer smo izračunali, kakšna je procentualna sprememba v prometu med (2. stolpec) in po akciji (3. stolpec). Za izdelke, ki so imeli več akcij, smo izračunali povprečno spremembo, lahko pa bi uporabili tudi kakšno drugo metodo aproksimacije – recimo z metodo najmanjših kvadratov v tridimenzionalnem prostoru. Rezultati so predstavljeni v TABELI 1: *Učinki akcij* (ki se nahaja v Dodatku), vse vrednosti pa so v odstotkih.

Tabela je urejena po zadjem stolpcu, torej nam pove, katere izdelke (izmed 100 obravnavanih) se najbolj splača postaviti v akcijo, če želimo pozitiven efekt na promet tudi po akciji. Podatki (predvsem prvih nekaj) izgledajo precej nerealni in tako ekstremne vrednosti lahko pripišemo motnjam pri podatkih. Vseeno lahko opazimo, da se v splošnem akcija prodajalcu iz vidika povečevanja prometa splača – prvo se mu poveča promet zaradi povečane potrošnje, potem pa zaradi kombinacije povečane potrošnje in (ponovnega) dviga cene. Vendar pa lahko predvidevamo, da ob akciji zaradi učinka substitucije povzročimo padec prometa za druge, podobne izdelke.

3.2 Analiza nakupov uporabnikov

Izbrali smo deset uporabnikov z največ nakupi, saj nam to zagotavlja dovolj veliko količino podatkov za analizo vsakega posebej. Iste metode kot so predstavljene v tem razdelku seveda lahko uporabimo tudi pri uporabnikih z manj nakupi, vendar se s tem (za nekatere metode) znatno zmanjša točnost napovedi. Za te metode bi bilo zato v primeru praktične uporabe smiselno določiti neko spodnjo mejo za število nakupov, ki jih je uporabnik že opravil.

3.2.1 Identifikacija zaželenih in nezaželenih izdelkov

Radi bi opredelili odnos do izdelkov, ki jih obravnavani uporabnik kupuje. Natančneje, zanima nas, ali obstajajo izdelki, za katere lahko sklepamo, da jih je uporabnik kupil le enkrat in nato nikoli več? Takšnih izdelkov potem temu in njemu podobnim uporabnikom ne priporočamo, saj predvidevamo da uporabnik z izdelkom ni bil zadovoljen.

V Dodatku je izsek grafa (GRAF 1*: nakupi uporabnika*), ki prikazuje nakupe uporabnika. Graf je precej velik (natančneje, višina je število različnih artiklov, ki jih uporabnik kupi, v konkretnem primeru okoli 1000, dolžina pa število nakupov (251)). V vrsticah so predstavljeni izdelki, pika pa pomeni da je bil nek izdelek kupljen (nakupi so predstavljeni na ordinatni osi).

Verjetno nezaželeni izdelki za obravnavanega uporabnika so tisti, ki se pojavijo na grafu le enkrat – na izseku so obarvani s sivo. Mera gotovosti za to, da smo pravilno napovedali 'nezaželene izdelke' mora temeljiti na številu nakupov, ki jih uporabnik opravi po tem, ko kupi 'nezaželen izdelek' in na tipu izdelka (ali gre za izdelek ki se sicer troši pogosto).

Mogoče bi lahko tudi ugotovili, ali je uporabnik izdelek zamenjal za nek substitut (temu bi potem ocena 'zaželenosti' narasla). Ta problem je sicer zelo občutljiv na število nakupov.

3.2.2 Ciklična potrošnja izdelkov

Za izdelke, ki jih obravnavani uporabnik dovolj pogosto kupuje, poskušamo ugotoviti, ali jih kupuje v časovnih intervalih in le-te identificirati. Tudi ta problem je občutljiv na število nakupov. Uporabna vrednost te informacije je v tem, da lahko v danem trenutku predvidimo, ali se bo zgodil nakup nekega izdelka s strani obravnavanega uporabnika, ali ne.

Uporabimo statistični pristop – iščemo interval zaupanja, v katerem bi se z neko verjetnostjo zgodil naslednji nakup. Ta nam za določeno stopnjo (med 0 in 1) in ocene parametrov pove meje intervala, v katerem se nahaja neka slučajna spremenljivka (naslednji nakup), ki je porazdeljena isto kot so porazdeljeni podatki. Parametri so: perioda (povprečen čas, ki mine med dvema nakupoma določenega izdelka), standardni odklon (pove, kako močno varirajo časi med nakupi), in datum zadnjega nakupa.

Če privzamemo, da se trenutno nahajamo v času 2014-01-16 (prvi dan, za katerega nimamo več podatkov) predvidevamo, da bo uporabnik, v kolikor na ta dan opravi nakup, kupil izdelke, ki so v TABELI 2: *cikličnost potrošnje* ki se nahaja v Dodatku, obarvani rumeno (za te izdelke je 'trenutni' datum 2014-01-16 znotraj intervala). Gledamo.

V našem primeru je stopnjo zaupanja 0,9. Za izdelek s šifro 157869 ("Solata endivija", 2. vrstica v tabeli) bo tako glede na naše podatke veljala napoved, da se bo naslednji nakup z verjetnostjo 90 % zgodil v obdobju med 16. in 30. 1. 2014. Pri tem je potrebno poudariti, da bi se v praksi ocene parametrov računale sproti, torej bi se z akumulacijo podatkov natančnost napovedi povečevala.

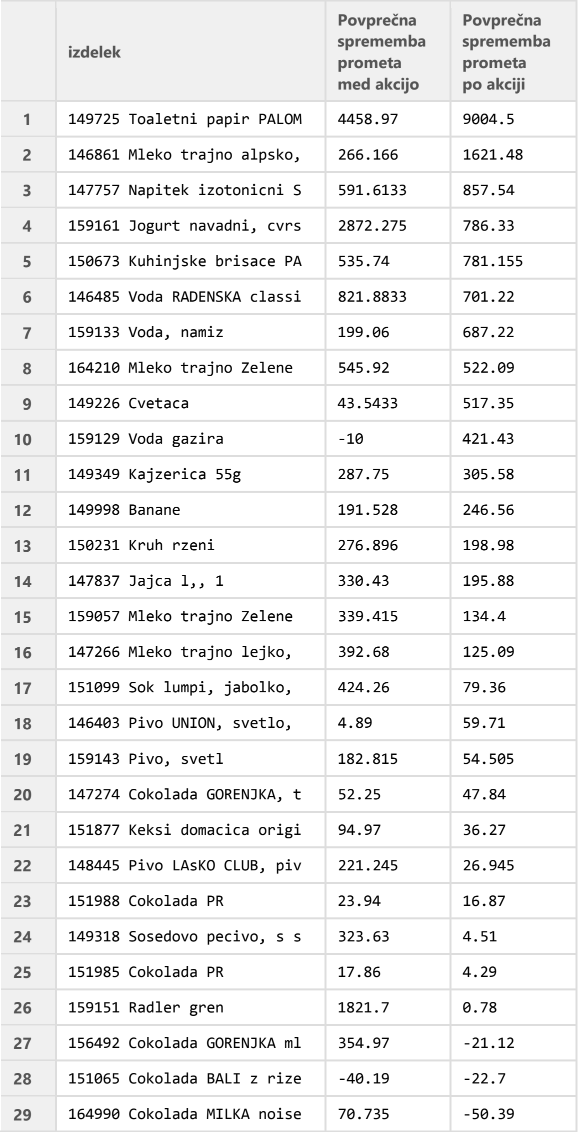
4. ZAKLJUČEK

Najprej smo opravili cenovno analizo, ki temelji na podatkih o prodanih izdelkih. Cilj analize je bil predvsem raziskati, kako se potrošnja odziva na spremembe v ceni. Tu smo ločili splošno obravnavo in obravnavo posebnih sprememb v ceni – akcij. Pridobljeni rezultati so bili v nekaterih primerih pričakovani, v drugih nekoliko manj.

Nato smo analizirali nakupovalne navade nekaterih uporabnikov, kar je uporabno predvsem za potrebe priporočanja in je tudi prvotni cilj projekta. Najprej smo se osredotočili na 'negativno selekcijo' priporočanja, torej smo poskušali identificirati izdelke ki jim bomo dali negativno utež. Tu je pomembno, da upoštevamo 'mero gotovosti', ki smo jo zaenkrat le opisno opredelili. Nato smo preverili, kaj lahko predvidimo o času nakupa nekega izdelka in po statistični analizi prišli do zaključka, da za dovolj obsežno količino podatkov lahko napovemo časovni interval, ko se zgodi naslednji nakup in povedali, kako bi to lahko bilo uporabno v smislu priporočanja.

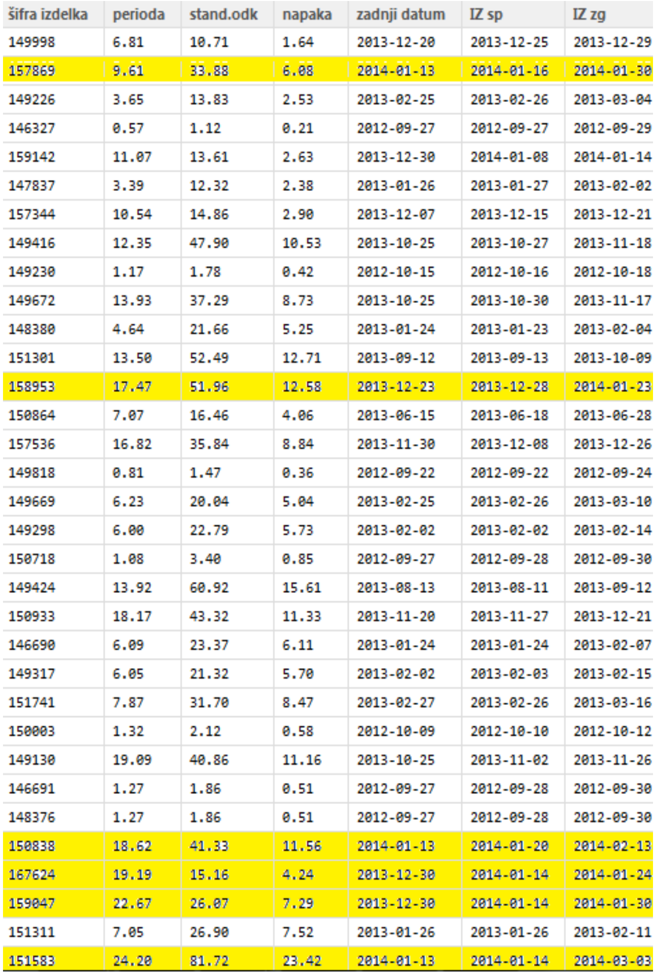
5. DODATEK

TABELA 1: *učinki akcij*



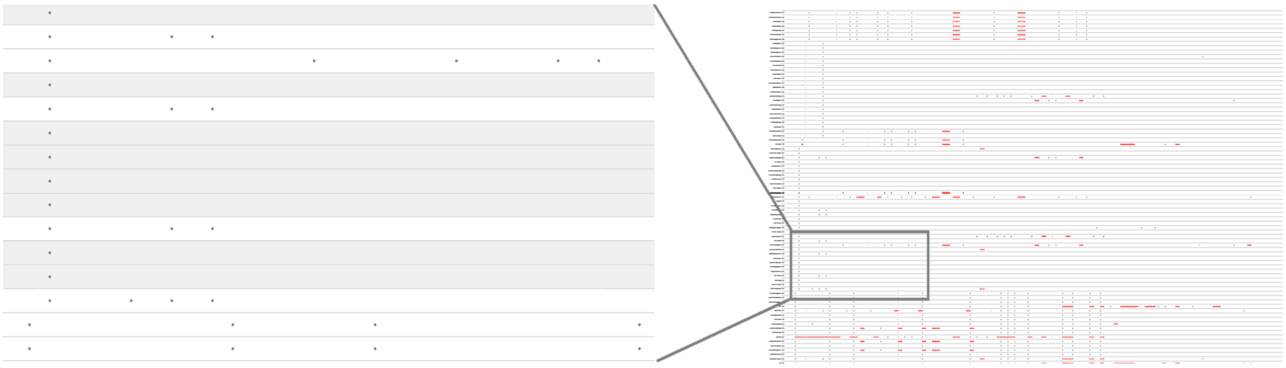
	izdelek	Povprečna sprememba prometa med akcijo	Povprečna sprememba prometa po akciji
1	149725 Toaletni papir PALOM	4458.97	9004.5
2	146861 Mleko trajno alpsko,	266.166	1621.48
3	147757 Napitek izotonicni S	591.6133	857.54
4	159161 Jogurt navadni, cvrs	2872.275	786.33
5	150673 Kuhinjske brisace PA	535.74	781.155
6	146485 Voda RADENSKA classi	821.8833	701.22
7	159133 Voda, namiz	199.06	687.22
8	164210 Mleko trajno Zelene	545.92	522.09
9	149226 Cvetaca	43.5433	517.35
10	159129 Voda gazira	-10	421.43
11	149349 Kajzerica 55g	287.75	305.58
12	149998 Banane	191.528	246.56
13	150231 Kruh rzeni	276.896	198.98
14	147837 Jajca l,, 1	330.43	195.88
15	159057 Mleko trajno Zelene	339.415	134.4
16	147266 Mleko trajno lejko,	392.68	125.09
17	151099 Sok lumpi, jabolko,	424.26	79.36
18	146403 Pivo UNION, svetlo,	4.89	59.71
19	159143 Pivo, svetl	182.815	54.505
20	147274 Cokolada GORENJKA, t	52.25	47.84
21	151877 Keksi domacica origi	94.97	36.27
22	148445 Pivo LAsKO CLUB, piv	221.245	26.945
23	151988 Cokolada PR	23.94	16.87
24	149318 Sosedovo pecivo, s s	323.63	4.51
25	151985 Cokolada PR	17.86	4.29
26	159151 Radler gren	1821.7	0.78
27	156492 Cokolada GORENJKA ml	354.97	-21.12
28	151065 Cokolada BALI z rize	-40.19	-22.7
29	164990 Cokolada MILKA noise	70.735	-50.39

TABELA 2: *cikličnost potrošnje*



šifra izdelka	perioda	stand.odk	napaka	zadnji datum	IZ sp	IZ zg
149998	6.81	10.71	1.64	2013-12-20	2013-12-25	2013-12-29
157869	9.61	33.88	6.08	2014-01-13	2014-01-16	2014-01-30
149226	3.65	13.83	2.53	2013-02-25	2013-02-26	2013-03-04
146327	0.57	1.12	0.21	2012-09-27	2012-09-27	2012-09-29
159142	11.07	13.61	2.63	2013-12-30	2014-01-08	2014-01-14
147837	3.39	12.32	2.38	2013-01-26	2013-01-27	2013-02-02
157344	10.54	14.86	2.90	2013-12-07	2013-12-15	2013-12-21
149416	12.35	47.90	10.53	2013-10-25	2013-10-27	2013-11-18
149230	1.17	1.78	0.42	2012-10-15	2012-10-16	2012-10-18
149672	13.93	37.29	8.73	2013-10-25	2013-10-30	2013-11-17
148380	4.64	21.66	5.25	2013-01-24	2013-01-23	2013-02-04
151301	13.50	52.49	12.71	2013-09-12	2013-09-13	2013-10-09
158953	17.47	51.96	12.58	2013-12-23	2013-12-28	2014-01-23
150864	7.07	16.46	4.06	2013-06-15	2013-06-18	2013-06-28
157536	16.82	35.84	8.84	2013-11-30	2013-12-08	2013-12-26
149818	0.81	1.47	0.36	2012-09-22	2012-09-22	2012-09-24
149669	6.23	20.04	5.04	2013-02-25	2013-02-26	2013-03-10
149298	6.00	22.79	5.73	2013-02-02	2013-02-02	2013-02-14
150718	1.08	3.40	0.85	2012-09-27	2012-09-28	2012-09-30
149424	13.92	60.92	15.61	2013-08-13	2013-08-11	2013-09-12
150933	18.17	43.32	11.33	2013-11-20	2013-11-27	2013-12-21
146690	6.09	23.37	6.11	2013-01-24	2013-01-24	2013-02-07
149317	6.05	21.32	5.70	2013-02-02	2013-02-03	2013-02-15
151741	7.87	31.70	8.47	2013-02-27	2013-02-26	2013-03-16
150003	1.32	2.12	0.58	2012-10-09	2012-10-10	2012-10-12
149130	19.09	40.86	11.16	2013-10-25	2013-11-02	2013-11-26
146691	1.27	1.86	0.51	2012-09-27	2012-09-28	2012-09-30
148376	1.27	1.86	0.51	2012-09-27	2012-09-28	2012-09-30
150838	18.62	41.33	11.56	2014-01-13	2014-01-20	2014-02-13
167624	19.19	15.16	4.24	2013-12-30	2014-01-14	2014-01-24
159047	22.67	26.07	7.29	2013-12-30	2014-01-14	2014-01-30
151311	7.05	26.90	7.52	2013-01-26	2013-01-26	2013-02-11
151583	24.20	81.72	23.42	2014-01-13	2014-01-14	2014-03-03

GRAF 1: *nakupi uporabnika*



ANALIZA MOŽNOSTI ZAZNAVANJA PODOBNOSTI MED UPORABNIKI

Božidara Cvetković, Mitja Luštrek
Department of Intelligent Systems
Jožef Stefan Institute
Jamova cesta 39, 1000 Ljubljana, Slovenia
e-mail: {boza.cvetkovic, mitja.lustrek}@ijs.si

POVZETEK

Prispevek predstavlja preliminarne rezultate analize možnosti zaznavanja podobnosti med uporabniki. Cilj analize je izbrati najboljši pristop, ki bo uporabljen v metodi za prilagajanje modela uporabniku MCAT.

1 UVOD IN SORODNO DELO

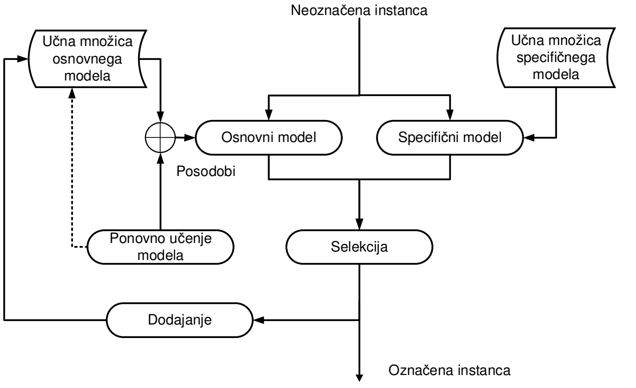
V aplikacijah, kjer se uporabljajo modeli strojnega učenja za napovedovanje človeškega obnašanja, se pogosto dogaja, da točnost delovanja v realnem okolju ni primerljiva točnosti delovanja v laboratorijskem okolju. Razlog je tako omejena količina učnih podatkov, kot tudi fizična razlika ter razlika v navadah med ljudmi. Fizične razlike se kažejo bodisi v drugačnosti izvajanja akcij v primeru problema prepoznavanja aktivnosti ali v drugačnem metabolnem sistemu v primeru problema ocene porabe energije.

Točnost modela za določenega uporabnika lahko zvišamo na dva načina:

- označimo dodatne učne podatke specifične za novega uporabnika in uporabimo nadzorovano učenje za nov model ali
- uporabimo katero od metod, ki nenadzorovano ali pol-nadzorovano prilagodijo model trenutnemu uporabniku.

Najboljše izboljšanje dobimo z označevanje dodatnih podatkov. Vendar je ta proces časovno zelo zahteven, duhamoren in drag, tako za označevalca kot za uporabnika. Velikokrat se zgodi, da je samo označevanje podatkov v ciljnem okolju onemogočeno, bodisi zaradi samega klasifikacijskega problema (označevanje padcev je lahko nevarno) ali pa zato, ker nam manjkajo dodatne naprave, ki niso mobilne in jih lahko uporabljamo izključno v laboratoriju (poraba človeške energije iz izdihanega zraka). V tem primeru se izkažejo rešitve, ki uporabljajo pol-nadzorovano učenje, bolj primerne. Metode pol-nadzorovanega učenja označijo neoznačene podatke in glede na določeno pravilo izberejo ali zavržejo trenutni podatek za dodajanje v učno množico. Nad učno množico, ki vsebuje nove podatke, se nato uporabi nadzorovan algoritem za strojno učene za pridobitev novega, prilagojenega modela.

Metode pol-nadzorovanega učenja lahko kategoriziramo na več načinov. Glede na število klasifikatorjev, glede na število dimenzij (ortogonalnost atributnih vektorjev), glede na način prilagajanja in glede na to ali se uporablja omejena ali neomejena količina neoznačenih podatkov. Najbolj osnovna metoda je samo-učenje (self-training [1]), ki uporablja en klasifikator za označevanje podatkov in ročno nastavljen prag za odločitev o izbiri podatka za dodajanje v učno množico. Prag je po navadi nastavljen tako, da mora biti zaupanje v napoved 100%. Nadgradnja metode z enim klasifikatorjem je dodajanje več klasifikatorjev, ki so naučeni z različnimi algoritmi in za dodajanje uporabljajo večinski glas (Democratic co-learning [2]) ali več klasifikatorjev z istim učnim algoritmom in več dimenzijami (Co-training [3]). Pomanjkljivost prvega je v ročno nastavljenem pragu (100% zaupanje v napoved), problem drugega pa kompleksnost delitve prostora na dva ortogonalna dela ali dimenziji. Več o metodah pol-nadzorovanega učenja pišemo v našem preteklem delu, kjer smo prilagajali klasifikator za prepoznavanje aktivnosti novemu uporabniku. Pokazali smo, da lahko z mehanizmom za prilagajanje novemu uporabniku (MCAT - Multi-Classifier Adaptive Training [4]) in omejeno količino na novo označenih podatkov (3 aktivnosti po 30 sekund) zvišamo prepoznavanje aktivnosti za približno 12 odstotnih točk. Ogrodje MCAT metode je okvirno predstavljena na sliki 1.



Slika 1: *Ogrodje metode MCAT*

Ogrodje MCAT pričakuje naslednje klasifikatorje:

- osnovni model: model, ki se je v laboratorijskem okolju izkazal za najboljšega,
- specifični model: model ali množica modelov, ki vsebujejo znanje o specifikah trenutnega uporabnika,

- selekcija: model, ki izbere končno oznako,
- dodajanje: model, ki se odloča, ali je trenutna instanca dovolj kvalitetna za dodajanje v učno množico osnovnega modela.

Cilje trenutne raziskave je uporabiti isto ogrodje na regresijski domeni, bolj specifično za oceno porabe človeške energije. Označevanje podatkov za novega uporabnika je v tem primeru onemogočeno, saj bi uporabnik moral v laboratorij, kjer se nahajajo potrebne naprave (Cosmed k4b2).

Ta prispevek predstavlja analizo pristopov za možnost detekcije podobnosti med uporabniki. Privzeli bi, da pristop z najboljšim delovanjem opiše trenutnega uporabnika zadosti dobro da ga lahko uporabimo kot specifični model v MCAT algoritmu.

2 NABOR PODATKOV

V raziskavi smo uporabili dva nabora podatkov in sicer podatke, ki so uporabljeni kot učna množica splošnega modela in pa nabor podatkov, ki predstavlja bio-impedanco oseb vsebovanih v učni množici splošnega modela.

Učna množica splošnega modela je bila zbrana v kontroliranem laboratorijskem okolju Fakultete za Šport in vsebuje podatke 10 ljudi, ki so izvajali vnaprej določene sklope aktivnosti. Opremljeni so bili s pospeškomeri na prsih in stegnu, prsnim pasom za merjenje srčnega utripa, napravo Senswear, ki meri oddajanje toplote človeka, galvanski odziv kože in telesno temperaturo ter oceni človekovo porabo energije in indirektnim kalorimetrom Cosmed k4b2, ki meri porabo energije na osnovi izdihanega ogljikovega dioksida in porabe kisika. Ta nabor podatkov je bil uporabljen za gradnjo in vrednotenje več regresijskih modelov za oceno porabe energije. Izbran je bil najboljši, ki vsebuje podatke pospeškomerov, blizu telesne temperature in srčnega utripa. Ta model je privzet za splošni model, deluje s povprečno absolutno napako (MAE) 0.55 MET (Metabolic Equivalent of Task).

Učna množica bio-impedance so podatki, pridobljeni iz naprave InBody [1], ki analizira sestavo telesa. Podatki vsebujejo: višino, težo, starost, količino vode v celicah, izven celic, količino proteinov, mineralov, maščobe, maso skeleta, indeks telesne teže, razmerje med pasom in boki in podatke o teži udov. Vsebuje tudi maksimalne in minimalne vrednosti za vsak tip podatkov, kar smo uporabili na normalizacijo in dodali se maksimalen in minimalen srčni utrip uporabnika. Ta je bil umerjen med 15 minutnim ležanjem (minimalen srčni utrip) in po dveh minutah intenzivnega teka (maksimalen srčni utrip). To učno množico smo uporabili za ugotavljanje podobnosti med uporabniki.

3 PRISTOP ZA UGOTAVLJANJE PODOBNOSTI MED UPORABNIKI

Podobnost med uporabniki smo analizirali z uporabo nabora podatkov bio-impedance in testirali na podatkih osebe, katere podobnost smo ugotavljali. Cilj je pridobiti množico oseb, ki so najbolj podobni novemu uporabniku in iz njihovih modelov oceniti porabo energije novega uporabnika. Točnost ocene mora biti višja od splošnega modela, ki je naše izhodišče.

3.1 Razvrščanje v skupine ali gručenje

Za razvrščanje v skupine smo uporabili algoritem k-means iz orodja za strojno učenje Weka [6]. Za idealno število gruč smo uporabili koeficient Silhouette, ki poda mero, kako dobro podatek ustreza trenutni gruči. Koeficient je definiran z naslednjo enačbo.

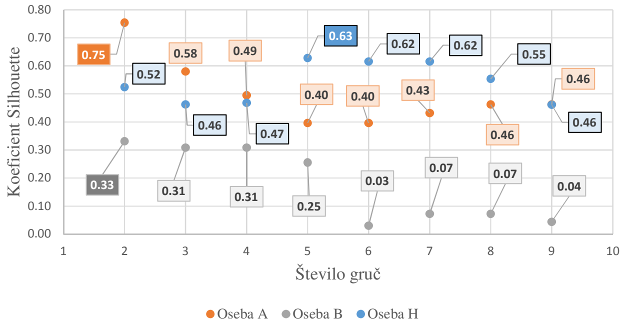
$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

$$-1 \leq s(i) \ \leq 1$$

Za izračun koeficienta uporabnika *i* uporabimo

- a(i) - povprečna razdalja vseh uporabnikov v gruči
- b(i) - najmanjša razdalja trenutnega uporabnika do sosednje gruče

Ustreznost gruče je definirana z velikostjo koeficienta. Najbolj ustrezna delitev je pri *s(i) = 1*, če je koeficient blizu *0* je na robu dveh gruč in če je *-1* verjetno bolj ustreza drugi gruči. Izračunan koeficient za tri osebe lahko vidimo na Sliki 2. Za osebi A in B je najboljša delitev na dve gruči in za osebo H na 5 gruč.



Slika 2: *Silhouette koeficient ustreznosti delitve.*

S to metodo smo dobili gruče podobnih oseb.

3.2 Meta klasifikacija

Za uteževanje ocen smo poizkusili še meta-klasifikator za vsako osebo posebej. Za meta-klasifikator smo uporabili podatke osmih oseb pri ocenjevanju devete. Za končne evaluacijo smo uporabili deseto osebo.

Začetno množico atributov meta klasifikatorja sestavljajo naslednji atributi:

- evklidske razdalje od trenutne osebe do vseh oseb v gruči,
- trenutna razpoznana aktivnost osebe,
- nivo aktivnosti (nizka, srednja, visoko),
- normaliziran srčni utrip osebe,

Tabela 1: *Rezultati glede na pristop ugotavljanja podobnosti med uporabniki. Pristopi so opisani v sekciji 3.3.*

				Pristopi					
	Splošni model (MAE)	Število gruč	Število oseb v gruči	A	B	C	D	E	F
Oseba A	0.49	2	8	0.53	0.49	0.49	0.49	0.48	0.48
Oseba B	0.69	2	3	0.77	0.69	0.70	0.69	0.73	0.69
Oseba C	0.64	3	4	0.75	0.60	0.61	0.60	0.58	0.59
Oseba D	0.55	4	1	0.93	0.54	0.54	0.54	0.48	0.49
Oseba E	0.44	2	8	0.40	0.47	0.48	0.47	0.44	0.44
Oseba F	0.55	2	8	0.68	0.60	0.60	0.60	0.55	0.55
Oseba G	0.57	2	8	0.50	0.61	0.62	0.61	0.56	0.56
Oseba H	0.46	5	2	0.42	0.51	0.51	0.51	0.46	0.46
Oseba I	0.64	2	8	0.67	0.63	0.63	0.63	0.72	0.63
Oseba J	0.50	6	1	0.65	0.47	0.47	0.47	0.53	0.50
Povprečno	**0.55**			**0.63**	0.56	0.56	0.56	0.55	**0.54**

- povprečna absolutna napaka ocene modela osebe glede na oceno splošnega modela,
- cona srčnega utripa po metodi Zoladz [8],
- procent povprečne absolutne napake ocene modela glede na oceno splošnega modela.

Delovanje meta-klasifikatorja je naslednje. Vsako instanco se oceni z modeli oseb, ki so v gruči, in vsaka ocena je ovrednotena s svojim meta-klasifikatorjem, ki vrne enega od dveh razredov: »da« ali »ne«. Da pomeni, da se ocena uporabi, in ne, da se zavrže. Poleg vsake klasifikacije klasifikator vrne stopnjo zaupanja v svojo napoved. Končna ocena se izračuna glede na število modelov, katerih rezultat je bil »da«:

- število »da« > 1; normalizira se stopnja zaupanja za vsak model, ki je klasificiral »da«. Normalizirane stopnje se uporabijo kot utež trenutne ocene in utežena vsota vseh tvori končno oceno,
- število »da« = 1; stopnja zaupanja je uporabljena kot utež ocene tega modela. Ostanek je uporabljen kot utež ocene splošnega modela. Utežena vsota obeh tvori končno oceno,
- število »da« = 0; uporabi se ocena splošnega modela

Uporabnost atributov smo ovrednotili s kombiniranjem vseh in izločili tiste atribute, ki ne pripomorejo k boljši točnosti izbire in hkrati točnosti ocene. Atributi, ki so ostali v končnem vektorju atributov, so:

- evklidske razdalje od trenutne osebe do vseh oseb v gruči,
- trenutna razpoznana aktivnost osebe,
- nivo aktivnosti (nizka, srednja, visko),
- cona srčnega utripa po Zoladz metodi [8].

3.3 Pristopi

Pristop A: Vsako instanco oceni devet modelov (posamezni model osebe) in končna ocena je povprečje ocen.

Pristop B: Vsako instanco ocenijo modeli oseb, ki so v gruči ,in končna ocena je povprečje ocen.

Pristop C: Vsako instanco ocenijo modeli oseb, ki so v gruči, in končna ocena je utežena vsota glede na evklidsko razdaljo do centroide v gruči.

Pristop D: Vsako instanco ocenijo modeli oseb, ki so v gruči, in končna ocena je utežena vsota glede na evklidsko razdaljo do nove osebe v gruči. Če je v gruči ena oseba je rezultat utežena vsota splošnega modela in modela osebe.

Pristop E: Za oceno so uporabljeni meta klasifikatorji in modeli vseh oseb.

Pristop F: Za oceno so uporabljeni meta klasifikatorji in modeli oseb v gruči.

4 REZULTATI

Rezultati predstavljajo evaluacijo vseh omenjenih pristopov. Cilj je izbrati pristop, ki vrača manjšo ali primerljivo točnost splošnemu modelu. Rezultati so predstavljeni v Tabeli 1 in sicer z povprečno absolutno napako (MAE) definirano z naslednjo enačbo:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|EE_{ocenjena} - EE_{prava}\right|$$

Končna ocena najboljšega pristopa je ocenjena z povprečno absolutno procentualno napako definirano z naslednjo enačbo (MAPE):

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{EE_{ocenjena} - EE_{prava}}{EE_{prava}}\right|$$

V obeh enačbah $EE_{ocenjena}$ predstavlja oceno porabe energije, kot jo vrne regresijski model in EE_{prava} je izmerjena poraba energije.

Točnost splošnega modela je predstavljena v drugem stolpcu Tabele 1. Povprečna napaka modela je 0.55 MET in MAPE modela je 25%. Prvi pristop (pristop A) uporabi povprečno oceno vseh oseb. Iz rezultata lahko vidimo, da se napaka poveča in da ta pristop ni pravilen, kar je tudi v skladu s hipotezo, da uporabljen model mora biti podoben modelu končne osebe. Pristop B uporabi dodatno znanje o medsebojni podobnosti oseb in za končno oceno uporabi povprečje ocen podobnih oseb (osebe v isti gruči). Rezultat je slabši od splošnega modela, tako v obliki MAPE 26% kot tudi MAE 0.56 MET. Pristop C uporabi za utež napovedi evklidsko razdaljo osebe do centroide. Končna točnost je slabša od splošnega modela in sicer 0.56 MET in 26% v obliki MAPE. Pristop D vrne primerljive rezultate kot pristopa B in C. Pristop E uporabi meta-klasifikator, vendar na vseh osebah. Iz rezultata lahko vidimo, da z vpeljavo meta klasifikatorja dosežemo primerljivo točnost, kot ga dobimo s splošnim modelom. Če uporabimo meta klasifikatorje samo na osebah ki so v gruči, pa pridobimo na točnosti in sicer 0.01 MET v obliki MAE in 3 odstotne točke v obliki MAPE.

5 ZAKLJUČEK

Ta prispevek predstavlja preliminarne rezultate analize pristopov za ugotavljanje podobnosti med uporabniki. Analiza je bila narejena na domeni ocene porabe človeške energije z namenom definirati specifični model za pol-nadzorovano metodo MCAT, katero bomo v prihodnjem delu nadgrajevali.

Pristop, ki vrača najboljšo točnost, uporablja algoritem gručenja za delitev oseb v skupine po podobnosti in meta klasifikatorje posameznih oseb v gruči za končno oceno porabe energije osebe. Z uporabo pristopa za podobnost izboljšamo rezultat najboljšega modela za 3 odstotne točke.

Prihodnje delo zajema razširitev pristopov in uporabo najboljšega pristopa v metodi MCAT.

References

[1] Frinken, V., Bunke, H.: Self-training Strategies for Handwriting Word Recognition. In: Perner P. (eds.) Advances in Data Mining. Applications and Theoretical Aspects. LNCS, vol. 5633, pp. 291--300, 2009.

[2] Zhou, Y., Goldman, S.:Democratic Co-Learning. In: 16th IEEE International Conference on Tools with Artificial Intelligence, pp. 594--602, IEEE press, 2004

[3] Blum, A. and Mitchell, T.: Combining labeled and unlabeled data with co-training. In: 11th annual conference on Computational learning theory, pp. 92--100, 1998.

[4] B. Cvetković, B. Kaluža, M. Luštrek, M. Gams, “Adapting Activity Recognition to a Person with Multi-Classifier Adaptive Training,” Journal of Ambient Intelligence and Smart Environments. Accepted for publication, 2014.

[5] InBody, http://www.e-inbody.com/

[6] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten. The WEKA Data Mining Software: An Update. SIGKDD Explorations. 11(1), pp. 10–18, 2009.

[7] Silhouette koeficient, http://en.wikipedia.org/wiki/Silhouette_(clustering)

[8] Zoladz, http://en.wikipedia.org/wiki/Heart_rate

VISUALIZATION OF EXPLANATIONS OF INCREMENTAL MODELS

Jaka Demšar, Zoran Bosnić, Igor Kononenko
University of Ljubljana, Faculty of Computer and Information Science
Večna pot 113, SI-1000 Ljubljana, Slovenia
e-mail: jaka.demsar0@gmail.com, {zoran.bosnic, igor.kononenko}@fri.uni-lj.si

ABSTRACT

The temporal dimension that is ever more prevalent in data makes the data stream mining (*incremental learning*) an important field of machine learning. In addition to accurate predictions, explanations of models and examples are a crucial component as they provide insight into model's decision and lessen its black box nature, thus increasing the user's trust. Proper visual representation of data is also very relevant to user's understanding -- visualization is often utilised in machine learning since it shifts the balance between perception and cognition to take fuller advantage of the brain's abilities. In this paper we review visualisation in incremental setting and devise an improved version of an existing visualisation of explanations of incremental models. We discuss the detection of concept drift in data streams and experiment with a novel detection method that uses the stream of model's explanations to determine the places of change in the data domain.

1 INTRODUCTION

Data streams are becoming ubiquitous. This is a consequence of the increasing number of automatic data feeds, sensoric networks and internet of things [1]. The defining characteristics of data streams are their transient dynamic nature and temporal component. In contrast with static datasets (used in batch learning), data streams (used in incremental learning) are large, changing, semi-structured and possibly unlimited. This poses a challenge for storage and processing as the data can be only read once. For incremental learning models, operations of model increment and decrement are vital. Concepts and patterns in data domain can change (*concept drift*) - we need to adapt to this phenomenon or the quality of our predictions deteriorates. According to *PAC* (*Probably approximately correct*) learning model, if the distribution, generating the instances is stationary, the error rate for sound machine learning algorithms will decline towards the Bayes error rate as the number of processed instances increases [9]. Consequently, when a statistically significant rise in error rate is detected, we can suggest that there has been a change in the generating distribution - concept drift.

The basis of *statistical process control* (*SPC*) [5] is detecting statistically significant error rate (using the central limit theorem) by monitoring the mean and standard deviation of a sequence of correct classification indicators. Another method, *Page-Hinkley test* [10] was devised to detect the change of a Gaussian signal and is commonly used in signal processing.

Bare prediction quality is not a sufficient property of a good machine learning algorithm. *Explanation* (a form of data postprocessing) of individual predictions and model as a whole is needed to increase the user's trust in the decision and provide insight in the workings of the model, which increases the models credibility. *IME* (*Interactions-based Method for* Explanation) [13] with its efficient adaptation [12] is a model independent method of explanation, which also addresses interactions of features and therefore successfully tackles the problem of redundant and disjunctive concepts in data. The explanation of the prediction for each instance is defined as a vector of contributions of individual feature values. Positive contibution implies that the particular feature value positively influenced the prediction (and vice versa) while the absolute value of a contributon is proportional to the magnitude of influence on the decision, i.e. the importance of that feature value. The sum of all contributions is equal to the difference between the prediction using all feature values and a prediction using no features (prediction difference). The explanation of a single prediction can be expanded to the whole model [12] and also to incremental setting [3]. In the latter case, drift detection (SPC) and adaptation are used to compensate for concept drift. Explanation of a data stream is therefore itself a data stream.

Related to exaplantion is *data visualisation* - a versatile tool in machine learning that serves two purposes; sense-making (data analysis) and communication as it conveys abstract concepts in a form, understandable to humans (it shifts the balance between perception and cognition to take fuller advantage of the brain's abilities [4]). The majority of published visualizations depict data that has a temporal component [8]. In this context, visualization acts as a form of summarization, since the datasets can be extremely large. The challenge lies in representing the temporal component (including concept drift), especially if we are limited to two-dimensional non-interactive visualisations.

The main goal of this paper is improving the existing methodology for visualising explanations of incremental models [3]. The feature value contributions are represented with customised bar charts. Multiple such charts are required to explain the model at diffrerent points in time. They become very difficult to read as a whole because of the large number of visual elements that we have to compare (we sacrifice macro view completely in favour of micro view). To consolidate these images and address the *change blindness* phenomenon, charts are stacked into a single plot, where the age and size of the exaplanation are represented with transparency (older and "smaller" explanations fade out). The resulting visualisation is not tainted by first impressions (as it is only one image) and is adequately dense and graphically rich. However, the major flaw of this approach lies in the situations when columns, representing newer explanations override older ones and thus obfuscate the true flow of changing explanations, for example, when the concept drift precipitates the attribute value contributions to increase in size without changing the sign. Concepts can therefore become not only hidden; what's more, the visualization can be deceiving, which we consider to be worse than just being too sparse. Therefore, we need to clarify the presentation of the concept drift along with an accurate depiction of each explanation's contributions while maintaining the macro visual value, that enables us to detect patterns and get a sense of true concepts and flow of changes behind the model.

An additional goal was to devise a method of concept drift detection which monitors the stream of explanations and detects anomalies in it; the detected anomalies are interpreted as a concept drift. We test the improved visualization and the novel concept drift detection method on two datasets and evaluate the results.

2 VISUALISATION FOR INCREMENTAL MODELS

When visualising explanations of individual predictions, horizontal bar charts are a fitting method also in the incremental setting. Individual examples are always explained according to the current model which, in our case, can change. This is not an obstacle, since the snapshot of the model is in fact the model that classified the example.

This approach fails with explanations of incremental models as we need a new figure for each local explanation. To successfully represent the temporality of incremental models, we use two variations of a line plot where the x axis contains time stamps of examples and the splines plotted are various representations of contributions (y axis).

The first type of visualization (Figures 2 and 3) has one line plot for each attribute. Contributions of values of the individual attribute are represented with line styles. The mean positive and mean negative contribution of the attribute as a whole are represented with two thick faded lines. Solid vertical lines indicate the spots where explanation of the model was triggered (and therefore become the joints for the plotted splines), while dashed vertical lines mark the places where the actual concept drift occurs in data. The second type is an aggregated version (Figure 3) where the mean positive and mean negative contributions of all attributes are visualized in one figure. In these two ways we condense the visualization of incremental models without a significant loss in information while still providing a quality insight into the model. Exact values of contributions along with timestamps of changes can be read out (micro view), while general patterns and trends can be recognised in the shapes of lines that are intuitive representations of flowing time (macro view). The resulting visualisations are dense with information, easily understandable (conventional plotting of independent variable, time, on x axis) and presented in gray-scale palette, making them more suitable for print.

3 DETECTING CONCEPT DRIFT USING THE STREAM OF EXPLANATIONS

When explaining incremental models, the resulting explanations are, in themselves, a data stream. This gives us the option to process it with all the methods used in incremental learning. In our case, we'll devise a method to detect outliers in the stream of explanations and declare such points as places of concept drift. The reasoning behind this is the notion that if the model does not change, then also the explanation of the whole model will not change. When an outlier is detected, we consider this to be an indicator of a significant change in model and thus also in the underlying data. In addition to this, the method provides us with a stream of explanations that is continuous to a certain degree of granularity and so enables us to overview the concepts behind the data at more frequent intervals than the existing explanation methodology.

We use a standard incremental learning algorithm [5] (learn by incrementally updating the model, decrement the models if it becomes too big according to the parameter, rebuild the model if we detect change [6]) and introduce some additional parameters. *Granularity* determines how often the explanation of the current model will be triggered. The generated stream of explanations (vectors of feature value contributions) will be compared using cosine distance. For each new explanation, the average cosine distance from all other explanations that are in the current model, is calculated. These values are monitored using the Page Hinkley test. When the current average cosine distance from other explanations has risen significantly, we interpret that as a change in data domain - concept drift. The last examples are then used to rebuild the model, the Page Hinkley statistic and the local explanation storage are reset (to monitor the new model).

The cosine distance is chosen because, in the case of explanations, we consider the direction of the vector of contributions to be more important than its size, which is very influential in the traditional Minkowski distances. The page Hinkley test is used in favour of SPC because of its superior drift detection times [9] and the lack of need for a buffer - examples are already buffered according to the granularity. The method is therefore model independent.

4 RESULTS

4.1 Testing methodology and datasets

We test the novel visualisation method and the concept drift detection method on two synthetic datasets, both containing multiple concepts with various degrees of drift between them. These datasets are also used in previous work [3], so a direct assessment of visualization quality and drift detection performance can be made. The naive Bayes classifier and the nearest neighbour classifier are used. Their usage yields very similar results in all tests, so only results obtained by testing with Naive Bayes are presented.

SEA concepts [11] is a data stream comprising 60000 instances with continuous numeric features $x_i \in [0,10]$, where $i \in \{1,2,3\}$. x_1 and x_2 are relevant features that determine the target concept with $x_1+x_2 \leq \beta$ where threshold $\beta \in \{7,8,9,9.5\}$. Concepts change sequentially every 15000 examples. Although the changes between the generated concepts are abrupt, class noise is inserted into each block.

The instances of second dataset, STAGGER [2], represent geometrical shapes which are in the feature space described by *size*, *color* and *shape*. The binary class variable is determined by one of the three target concepts (*small* ∧ *green, green* ∨ *square, medium* ∨ *large*). 4500 instances are divided into four blocks (concept-wise) with examples mixing near the change points according to a sigmoid function, so the dataset includes gradual concept drift.

4.2. Improved visualizations

Concept drifts in STAGGER dataset are correctly detected and adapted to as reflected in Figure 3. The defined concepts can be easily recognized from explanations triggered by the SPC algorithm - the change in explanation follows the change in concept. Windows generated by the vertical lines give us insight in local explanations of the model (where the concept is deemed to be constant). Disjunct concepts (2 and 3) and redundant feature values are all explained correctly (e.g. reduncacy of *shape* and disjunction of *size* values in concept 3). Figure 1 demonstrates how classifications of two instances with same feature values can be explained completely differently at different times - adapting to change is crucial in incremental setting. This is also evident in the aggregated visualization, which can be used to quickly determine the importance of each attribute.

For SEA dataset, explanations of instances are tightly corresponding to explanations of the model. As evident in Figure 2, the shape of contributions of features reflects the target concept; lower values increase the likelihood of positive classification and vice versa. Feature x_1 is correctly explained as irrelevant with its only contributions being the result of noise.

4.3. Concept drift detection

Evaluating the concept drift detection using the stream of explanations on the STAGGER dataset yielded positive results. As depicted in Figure 4, the method correctly detects concept drifts without false alarms and is in that regard similar to SPC method. The stream of explanations was similar to those obtained with other successful drift detection methods. Choices of larger granulations yielded similar results, but the change detection was obviously delayed. The concept drift was however never missed, provided that the granulation was smaller that the spacing between sequential changes in data. The delays of concept drift detection are correlated with the magnitude of change. For example, the last concept drift was detected with significant delay. In this regard, the proposed method is inferior to SPC algorithm - the concept drift detection in noticeably delayed and we're also dependant on two parameters – granulation and alert threshold, so the generality of the method is diminished.

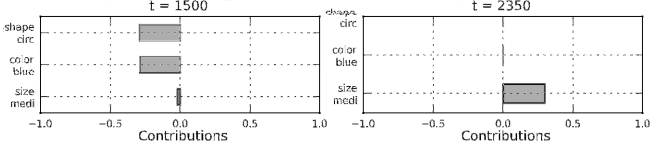


Figure 1: *Explanations of a single prediction at different times.*

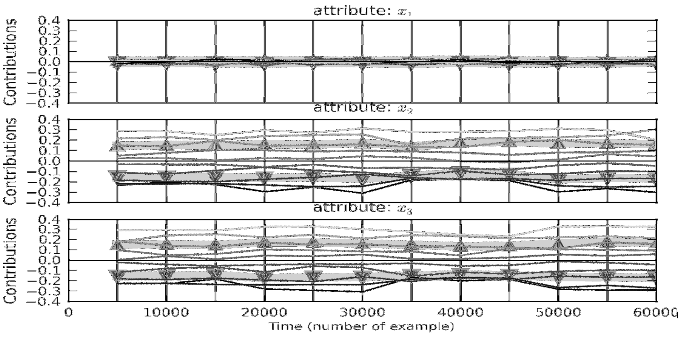


Figure 2: P*eriodically triggered explanations (SEA).*

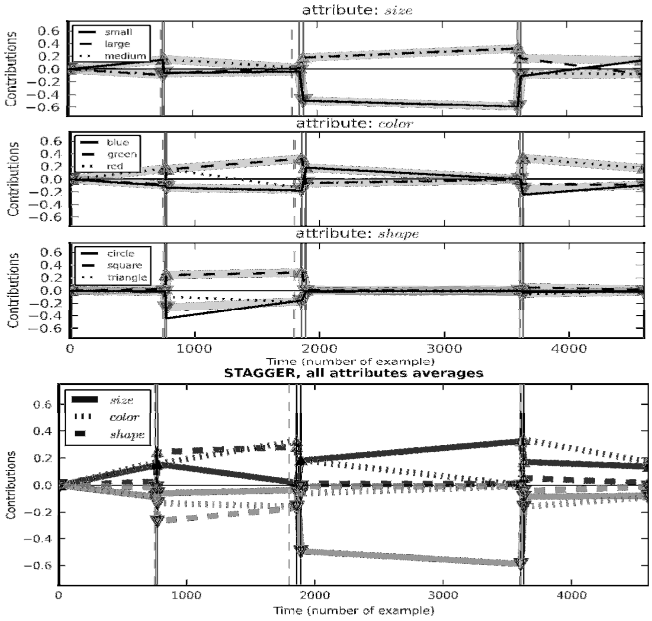


Figure 3: E*xplanations triggered at change detection (STAGGER).*

When testing with SEA datasets, the concept drift was not correctly detected. Changing the granulation and Page Hinkley alert threshold parameter resulted in varying degrees of false alarms or non reaction to change (Figure 4). This behaviour can be attributed to a small magnitude of change that occurs in data - the difference between concepts in data is quite small and continuous. However, when explaining this (incorrectly adapted) model, we still recognise true underlying concepts. This can be attributed to automatically decrementing the model when it becomes too big. It is important to note that this does not perform well in general, if the prior knowledge is insufficient for us to correctly decide on the maximum model size.

We conclude that, in this form, the presented method is not a viable alternative to the existing concept drift detection methods. Its downsides include high level of parametrization which requires a significant amount of prior knowledge and can also become improper if the model changes drastically. Consequently, another assessment of data is needed - the required manual supervision and lack of adaptability in this regard can be very costly and against the requirements of a good incremental model. The concept drift detection is also not satisfactory - it is delayed in the best case or concepts can be missed or falsely alerted in the worst case. Another downside is the time complexity - the higher the granularity the more frequent explanations will be, which will provide us with a good stream of explanations but be very costly time-wise. The method is therefore not feasible in environments where quick incremental operations are vital. However, if we can afford such delays, we get a granular stream of explanations which gives us insight into the model for roughly any given time.

A note at the end: we should always remember that we are explaining the models and not the concepts behind the model. Only if the model performs well, we can claim that our explanations truly reflect the data domain [12]. This can be tricky in incremental learning, as at the time of a concept drift, the quality of the model deteriorates.

5 CONCLUSION

The new visualization of explanation of incremental model is indeed an improvement compared to the old one. The overriding nature of the old visualisation was replaced with an easy to understand timeline, while the general concepts (macro view) can still be read out from the shape of the lines. Micro view is also improved as we can determine contributions of attribute values for any given time.

The detection of concept drift using the stream of explanations did not prove to be suitable for general use based on the initial experiments. It has shown to be hindered by delayed detection times, missed concept drift occurrences, false alarms, high level of parametrization and potential high time complexity. This provides motivation for further experiments in this field, especially because the stream of explanations provides good insight into the model with accordance to the chosen granulation.

The main goal of future research is finding a true adaptation of IME explanation methodology to incremental setting, i.e. efficient incremental updates of explanation at the arrival of each new example. Truly incremental explanation methodology would provide us with a stream of explanations of finest granularity. In addition to this, a number of new possibilities for visualisation would emerge, particularly those that rely on finely granular data, such as ThemeRiver [7].

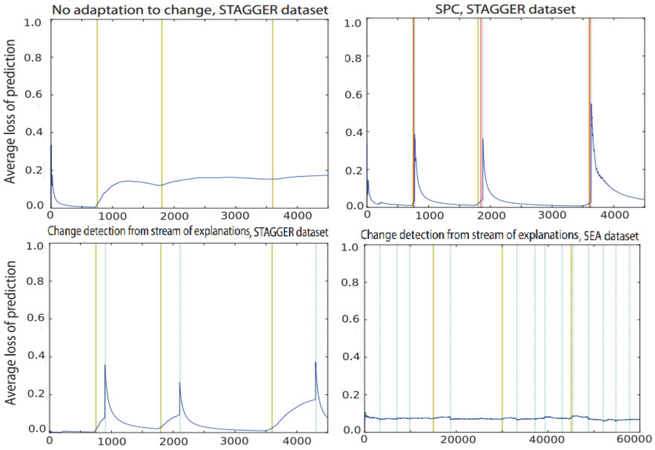


Figure 4: *Performance of various change detection methods. Yellow line indicates true change in concepts, green line indicates change detection and adaptaion).*

References

[1] C. C. Aggarwal, N. Ashish, A. P. Sheth. The internet of things: A survey from the data-centric perspective. In *Managing and Mining Sensor Data*. Springer, 2013.

[2] A. Bifet, G. Holmes, R. Kirkby, B. Pfahringer, M. Braun. Moa: Massive online analysis.

[3] Jaka Demšar. Explanation of predictive models and individual predictions in incremental learning (In Slovene). B.S. Thesis, University of Ljubljana, 2012.

[4] S. Few. *Now You See It: Simple Visualization Techniques for Quantitative Analysis*. Analytics Press, 1st edition, 2009.

[5] J. Gama. *Knowledge Discovery from Data Streams.* Chapman & Hall/CRC, 1st edition, 2010.

[6] D. Haussler. Overview of the probably approximately correct (PAC) learning framework, 1995.

[7] S. Havre, B. Hetzler, and L. Nowell. Themeriver: Visualizing theme changes over time. In *Proc. IEEE Symposium on Information Visualization*, 2000.

[8] C. Ratanamahatana, J. Lin 0001, D. Gunopulos, E. J. Keogh, M. Vlachos, and G. Das. Mining time series data. In *The Data Mining and Knowledge Discovery Handbook*. Springer, 2005.

[9] R. Sebastião and J. Gama. A study on change detection methods. In *Progress in Artificial Intelligence, 14th Portuguese Conference on Artificial Intelligence*, EPIA 2009

[10] E. S. Page. Continuous Inspection Schemes. *Biometrika*, Vol. 41:100-115, 1954.

[11] W. N. Street and Y. S. Kim. A streaming ensemble algorithm for large-scale classification. In *Proceedings of the 7th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD ’01, New York, NY, USA, 2001.

[12] E. Štrumbelj and I. Kononenko. An efficient explanation of individual classifications using game theory. *The Journal of Machine Learning Research*, 11:1–18, 2010.

[13] E. Štrumbelj, I. Kononenko, and M. Robnik Šikonja. Explaining instance classifications with interactions of subsets of feature values. *Data Knowl. Eng.*, 68(10):886–904, October 2009.

DETECTION OF IRREGULARITIES ON AUTOMOTIVE SEMIPRODUCTS

Erik Dovgan[1], Klemen Gantar[2], Valentin Koblar[3,4], Bogdan Filipič[1,4]
[1] Department of Intelligent Systems, Jožef Stefan Institute, Jamova cesta 39, SI-1000 Ljubljana, Slovenia
[2] Faculty of Computer and Information Science, University of Ljubljana, Večna pot 113, SI-1000 Ljubljana, Slovenia
[3] Kolektor Group d.o.o., Vojkova ulica 10, SI-5280 Idrija, Slovenia
[4] Jožef Stefan International Postgraduate School, Jamova cesta 39, SI-1000 Ljubljana, Slovenia
erik.dovgan@ijs.si, kg6983@student.uni-lj.si, valentin.koblar@kolektor.com, bogdan.filipic@ijs.si

ABSTRACT

The use of applications for automated inspection of semiproducts is increasing in various industries, including the automotive industry. This paper presents the development of an application for automated visual detection of irregularities on commutators that are parts of vehicle's fuel pumps. Each type of irregularity is detected on a partition of the commutator image. The initial results show that such an automated inspection is able to reliably detect irregularities on commutators. In addition, the results confirm that the set of attributes used to build the classifiers for detecting individual types of irregularities and the priority of these classifiers significantly influence the classification accuracy.

1 INTRODUCTION

Information technology (IT) is replacing human work in numerous domains. Such a technology is especially suitable for repetitive non-creative procedures where high accuracy is required. Automotive industry is introducing IT in various segments, for example in storage management and automated inspection of semiproducts.

Automated inspection of semiproducts can be done by analyzing data from several sources, such as sensors, lasers and cameras. Utilization of cameras for this purpose has several advantages, e.g., it is fast, thus not slowing down the production line, it is cheaper in comparison to highly specialized sensors, and the same hardware can be used for inspection of heterogeneous semiproducts.

This paper presents the development of an application for automated visual inspection of commutators as semiproducts for automotive industry. This application processes images of commutators with computer vision algorithms to obtain the attributes describing visual properties of the commutator. These attributes are then used by machine learning algorithms to classify the commutators, i.e., to determine whether or not they contain irregularities and, in the case they do, what is the type of irregularities. Experimental detection of irregularities was performed using various sets of attributes and various classification modes, including the detection of individual types of irregularities with binary classifiers and the classification of all types of irregularities with a single classifier.

The paper is further organized as follows. The problem of detecting irregularities on commutators is presented in Section 2. Section 3 describes the application for detecting irregularities that was designed and implemented in a prototype form for a specific production line. The experiments and results from the development process are presented and discussed in Section 4. Finally, Section 5 concludes the paper with some ideas for future work.

2 PROBLEM DESCRIPTION

Commutators are parts of electric motors that periodically reverse the current direction between the rotor and the external circuit. If the electric motor is installed in the vehicle's fuel pump, it has to withstand the chemical stress, which is usually not the case for other types of electric motors. Therefore, special graphite-copper commutators are produced for this purpose.

The production of graphite-copper commutators involves several stages. Among them the most critical one is soldering of graphite and copper parts of the commutator. The quality of the soldered joint is crucial for the quality of the commutator since even the smallest joint irregularity is unacceptable. During the soldering phase, four types of irregularities may occur:

1. metalization defect, i.e., there are visible defects on the metalization layer,
2. excess of solder, i.e., more solder is applied than feasible,
3. deficit of solder, i.e., less solder is applied than feasible, and
4. disoriented, i.e., the copper part is not appropriately oriented with respect to the graphite part.

The analysis of these irregularities showed that each type occurs only on a specific part of the commutator. Consequently, when visually inspecting the commutator, its image can be partitioned into four segments, each showing the presence or the absence of an individual type of irregularity, and into the rest of the image that can be disregarded since it contains no information about the irregularities. Currently, these irregularities are detected through manual inspection of the commutators. This approach is time-consuming and its results may be subjective. The goal of this research is to design and implement an automated visual inspection of commutators that would overcome the weaknesses of the manual inspection.

3 AUTOMATED VISUAL INSPECTION OF GRAPHITE-COPPER COMMUTATORS

The idea for the automated visual inspection of graphite-copper commutators is to consist of three phases. Firstly, a digital image of the commutator is obtained. Secondly, this image is processed using computer vision algorithms that extract informative attributes. Finally, these attributes are used by classifiers to determine whether the irregularities are present on the commutator and identify their type in the case of their presence. Before applying this inspection procedure on the production line, the classifiers need to be built with machine learning algorithms.

3.1 Processing commutator images with computer vision algorithms

Commutator images are processed in several steps. Since the commutators are not properly aligned, their rotation angle and position in the image have to be determined first. The center of the commutator is detected by matching the image with the template image of the center. Next, the position of the commutator's pin is found. The line between between the center of the commutator and the pin is used to determine the rotation angle.

The next step of image processing consists of determining four regions of interest (ROIs), one for each type of irregularities. Each ROI is obtained by applying the corresponding binary mask to the image. Before applying the binary mask, the mask has to be properly positioned and rotated. To that end, the information about the center of the commutator and its rotation angle (obtained in the previous step) is used. As a result, four ROIs are obtained. They are further processed with the same sequence of computer vision algorithms, where only the input parameter values of these algorithms are specific for each ROI.

At this stage, ROIs are in RGB format. However, preliminary tests showed that in order to reliably detect the irregularities, only one color plane should be used. Moreover, these tests also showed, that the most appropriate color plane is the red one, with the exception of the *excess of solder* irregularity for which the best color plane is the blue one. Consequently, the most appropriate color plane is extracted from each ROI with respect to the observed irregularity. This extraction results in gray-scale ROIs.

Gray-scale ROIs are then filtered with the median filter to reduce noise from the images. For this purpose a 2D median filter is used, where the size of the median window is an input parameter.

In the next step, a threshold function is used to eliminate pixels that are not relevant for detecting the irregularities. Each ROI is processed with a specific value of the binary threshold. This step results in a black (background) and white (relevant regions) image.

Connected pixels are then grouped together with the connected-component labeling algorithm [5] in order to detect the connected regions. This enables to process relevant regions, i.e., particles, rather than single pixels.

In the last image processing step, a particle filter is used to remove small particles that can be present in the image due to noise. The size of the particles to be filtered is an additional input parameter.

After the image is processed with the computer vision algorithms, the following six attributes are calculated for each ROI, i.e., for each type of irregularity:

- the number of particles,
- the cumulative size of particles in pixels,
- the maximal size of particles in pixels,
- the minimal size of particles in pixels,
- the gross/net ratio of the largest particle, and
- the gross/net ratio of all particles.

These attributes are then used to build the classifiers and classify the commutator images.

3.2 Learning classifiers with machine learning algorithms

The goal of the classifiers is to determine whether a commutator contains any irregularities. Two approaches were applied to solve this classification problem:

1. all the attributes were included in a single set of attributes and a single classifier was built to classify the commutators into one out of five possible classes (either one of the four types of irregularities or no irregularity),
2. each type of irregularity was detected with a binary classifier, where the binary classifiers were prioritized to determine the irregularity when irregularities of several types were detected.

The classification approach using four binary classifiers was further structured based on the attributes and learning instances used when building the binary classifiers. Specifically, when building a binary classifier for detecting irregularities of a particular type, four learning modes were tested:

1. only attributes of the corresponding ROI and only commutators that are either without irregularities or contain irregularities of this particular type are used,

Class	Number of images
Without irregularities	212
Metalization defect	35
Excess of solder	35
Deficit of solder	49
Disoriented	32

Table 1: Distribution of test images.

Class	Median window size	Threshold value	Particle size
Metalization defect	3	54	13
Excess of solder	3	5	2
Deficit of solder	5	78	760
Disoriented	1	81	184

Table 2: Input parameter values for the computer vision algorithms.

Learning mode	Best priority	Max. accuracy [%]
1	C_1, C_3, C_2, C_4	81.8
2	C_3, C_2, C_1, C_4	77.1
3	C_2, C_3, C_1, C_4	81.5
4	C_1, C_3, C_4, C_2	83.5

Table 3: The best binary classifier priorities and classification accuracies of learning modes.

Highest priority	Best learning mode	Max. accuracy [%]
C_1	4	83.5
C_2	4	83.2
C_3	4	83.5
C_4	4	83.5

Table 4: The best learning modes and classification accuracies of binary classifier priorities.

2. all attributes, but only commutators that are either without irregularities or contain irregularities of this particular type are used,

3. only attributes of the corresponding ROI, but all commutators including irregularities of all types are used, and

4. all attributes and all commutators including irregularities of all types are used.

4 EXPERIMENTS AND RESULTS

The proposed method for detecting irregularities was tested on a set of images of commutators without irregularities and the ones containing irregularities. The distribution of the test images among the irregularity classes is shown in Table 1.

The applied computer vision algorithms were implemented in Open Computing Language (OpenCL) [3] that is suitable for deploying on embedded many-core platforms and installing in the production environments. More precisely, we used the OCL programming package [2], which is an implementation of OpenCL functions in the Open Computer Vision (OpenCV) library [4]. The connected-component labeling algorithm was implemented based on description from [5]. The input parameter values of computer vision algorithms were determined using a tuning procedure described in [1] and are shown in Table 2. The classifiers were built using the Weka machine learning environment [7]. In particular, the J48 algorithm, the Weka's implementation of the C4.5 algorithm for building decision trees [6], was used for this purpose.

Figure 1 shows the classification accuracies obtained with the tested classifiers and learning modes. When binary classifiers are applied, all the permutations of priorities are tested, therefore a distribution of classification accuracy is shown. The results indicate that the highest classification accuracy is obtained using learning mode 4, i.e., when the attributes describing all types of irregularities and the images of all commutators are used to build the binary classifiers. This enables to, for example, correctly classify a commutator with irregularity x_1 when the binary classifier for irregularity x_2 is used. Such performance is not guaranteed when building the binary classifiers for irregularity x_i without taking into account the images of irregularities $x_j, i \neq j$ (learning modes 1 and 2). Consequently, when classifying the commutators with previously unseen irregularities (learning modes 1 and 2), the classification accuracy varies significantly with respect to the priority of classifiers as shown in Figure 1. These results also confirm that partitioning the classification problem into four subproblems, one for each irregularity type, results in higher classification accuracy, but only if all attributes and commutators with all irregularities are used when building the binary classifiers (see the classification accuracy of learning mode 4 in comparison to classification accuracy of the single classifier in Figure 1). On the other hand, when building the binary classifiers from the reduced set of attributes or the reduced set of irregularities, the obtained classification accuracy is lower than the classification accuracy of the single classifier. Finally, these results show that the priority of classifiers influences the classification accuracy. The priority is especially important when using learning modes 1 and 2.

The results were further analyzed with respect to various priorities of binary classifiers and learning modes (see Tables 3 and 4). For this purpose, the binary classifiers were abbreviated as follows:

- C_1 – the binary classifier for detecting metalization defects,
- C_2 – the binary classifier for detecting the excess of solder,
- C_3 – the binary classifier for detecting the deficit of solder, and
- C_4 – the binary classifier for detecting disoriented commutators.

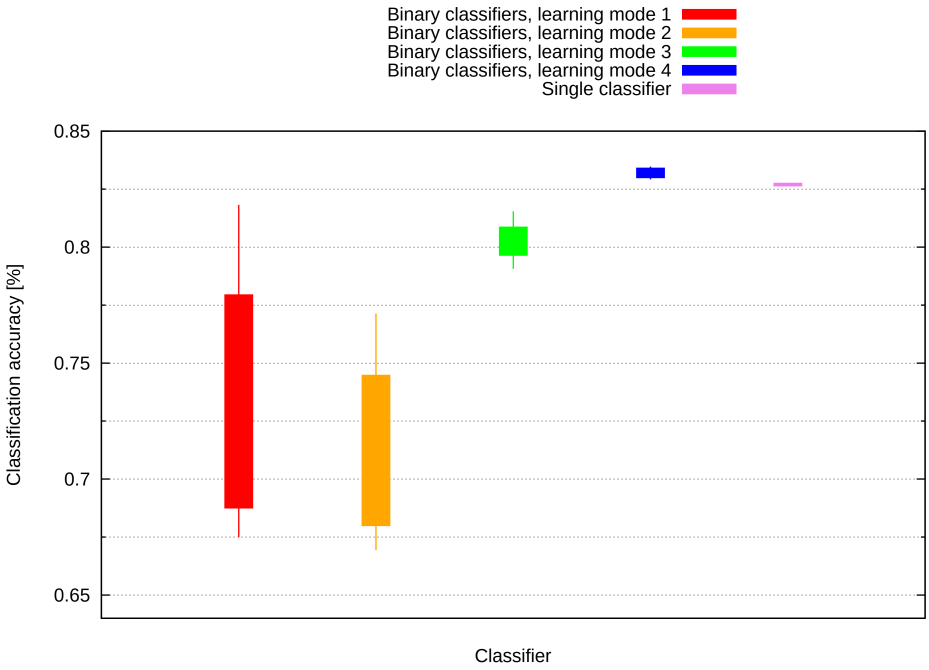


Figure 1: Classification accuracies of the tested classifiers and learning modes.

Table 3 shows the best priorities of binary classifiers and the corresponding classification accuracy for each learning mode. This table shows that the most important binary classifier is C_1 since it has the highest priority in two cases. In addition, the highest classification accuracy is obtained when this classifier has the highest priority. The second most important classifier is C_3 since it has the highest priority once and the second-highest priority three times.

Table 4 shows the best learning mode and the corresponding classification accuracy when the binary classifiers have the highest priority. These results show that the learning mode 4 is the best one irrespectively of the binary classifier that has the highest priority. Nevertheless, when classifier C_2 has the highest priority, a lower classification accuracy is achieved than in other cases.

5 CONCLUSIONS

This paper presented the development of an automated procedure for visual detection of irregularities on graphite-copper commutators after the soldering of graphite and copper in the production process. Four types of irregularities were detected a) with a single classifier and b) by partitioning the problem into four subproblems, learning the binary classifiers for each irregularity type and assigning priorities to the classifiers. The results show that the highest classification accuracy is achieved when the binary classifiers are used that are trained on the data of all types of irregularities. The results also indicate that the priority of classifiers significantly influences the classification accuracy and therefore needs to be taken into account.

In the future work we will test additional machine learning algorithms for potential improvement of the classification accuracy. Additional attributes could be extracted from the images with machine vision algorithms. It would be also interesting to compare our results with the results produced by the existing methods for detecting irregularities on semiproducts. The ultimate goal of this work is to put the automated inspection procedure into regular use on the production line.

ACKNOWLEDGEMENT

This work has been partially funded by the ARTEMIS Joint Undertaking and the Slovenian Ministry of Economic Development and Technology as part of the COPCAMS project (http://copcams.eu) under Grant Agreement number 332913, and by the Slovenian Research Agency under research program P2-0209.

References

[1] V. Koblar, E. Dovgan, and B. Filipič. Tuning of a machine-vision-based quality-control procedure for semiproducts in automotive industry. 2014. Submitted for publication.

[2] OpenCL module within OpenCV library. http://docs.opencv.org/modules/ocl/doc/introduction.html.

[3] OpenCL: The open standard for parallel programming. http://www.khronos.org/opencl/.

[4] OpenCV: Open source computer vision. http://opencv.org/.

[5] D. P. Playne, K. A. Hawick, and A. Leist. Parallel graph component labelling with GPUs and CUDA. *Parallel Computing*, 36(12):655–678, 2010.

[6] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.

[7] Weka Machine Learning Project. http://www.cs.waikato.ac.nz/ml/weka/index.html.

AN ELDERLY-CARE SYSTEM BASED ON SOUND ANALYSIS

Martin Frešer[1], Igor Košir[2], Violeta Mirchevska[1], Mitja Luštrek[1]
Department of Intelligent Systems, Jožef Stefan Institute, Slovenia[1]
Smart Com d.o.o., Slovenia[2]
e-mail: martin.freser@gmail.com, igor.kosir@smart-com.si,
{violeta.mircevska, mitja.lustrek}@ijs.si

ABSTRACT

This paper proposes an elderly-care system, which uses a single sensing device installed in the user's home, primarily based on a microphone. We present preliminary results on human activity recognition from sound data. The recognition is based on 19 types of sound features, such as spectral centroid, zero crossings, Mel-frequency cepstrum coefficients (MFCC) and linear predictive coding (LPC). We distinguished between 6 classes: sleep, exercise, work, eating, home chores and home leisure. We evaluated the recognition accuracy using 4 supervised learning algorithms. The highest accuracy, obtained using support vector machines, was 76%.

1. INTRODUCTION

Predictions made by the Statistical Office of the European Communities state that the over-65 population in EU28 expressed as a percentage of the working-age population (aged between 15 and 64) will rise from 27% in 2014 to 50% in 2060 [1]. This demographic trend puts an immense pressure to change current health and care practices, which already accounts for around 10 % of EU's GDP spending [2]. Innovative remote care systems are emerging, which motivate and assist the elderly to stay independent for longer, thus reducing the costs for elderly care and the burden put on the working-age population.

This paper presents an elderly-care system, which uses a single sensing device installed in the user's home, primarily based on a microphone. A microphone may serve both as a sensor and as a communication device. As a sensor, a microphone may be used for detecting user's activity (e.g. sleep, eating, opening a door) and consequently reasoning about potential problems related to the user (e.g. the user did not eat whole day, the user is sleeping much more than usual). As a communication device, it allows the user to initiate specific services by simply saying a keyword (e.g. call for help). It is also needed for remote user-carer communication.

Elderly care based on microphone has not received a lot of attention, although technology acceptance studies show that most users would accept to have a microphone for home care services. Ziefle et al. [3] performed a user acceptance study comparing three home-integrated sensor types: microphone, camera and positioning system. According to this study, the microphone (plus speaker) is the most accepted technology, followed by the positioning system, while the camera is ranked last.

The paper is organized as follows. In Section 2, we describe the system architecture. Activity recognition based on sound analysis is presented in Section 3. Evaluation of the presented approach on real-world recordings follows in

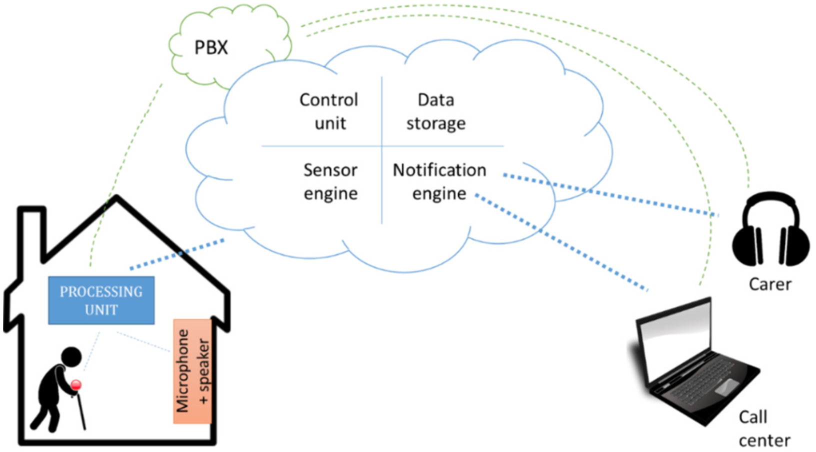


Figure 1: An elderly-care system architecture.

Section 4. Section 5 concludes the paper and presents future work.

2. SYSTEM ARCHITECTURE

Figure 1 presents the architecture of the elderly-care system.

Most of today’s commercial elderly-care systems offer a so-called emergency call functionality. The user is wearing a red button using which he/she may call for help in case of emergency. By pressing the red button, the care-system establishes a phone connection to a carer or a call center through a telephone network. We use a private branch exchange network (PBX) for establishing such calls.

We extend this functionality with sound analysis in order to provide context to the emergency call (e.g. past user activity), as well as to provide higher safety – the system establishes an emergency call when certain types of sound, such as screaming, are detected. In order to do so, a cloud-based system is established consisting of 4 main components: sensor engine, data unit, notification engine and control unit. The sensor engine analyses the sound in the apartment in order to detect user’s activity (e.g. eating, sleep) or critical sound patterns (e.g. screaming, fire alarm). The output of this engine is kept in the data unit. In case of emergency detected through sound analysis, the control unit notifies a carer or a specialized call center through the notification engine about the user who needs help and why automatic emergency call is being established. When the carer responds, the control unit establishes a telephone connection with the user’s apartment through the PBX network, enabling the carer to hear what is happening in the apartment and act accordingly.

3. ACTIVITY RECOGNITION BASED ON SOUND DATA

People can distinguish quite well between some everyday activities just by listening to them. For example, if we hear a spoon hit a plate, we can say that the person is probably eating; if we hear the sound of pressing keyboard buttons, we can say that the person is either at work or at home and is using a computer. We developed a system that automatically detects everyday home activities based on sound.

Figure 2 presents the process of activity recognition from sound data. Firstly, we gather data using a recording device, such as a microphone. When recording, some privacy protection should be taken into account (e.g. we could record short sequences of time so we could not be able to recognize spoken words). We propose recording for 5 minutes in the following way: we record 200ms in every second for 1 minute and we do not record remaining 4 minutes.

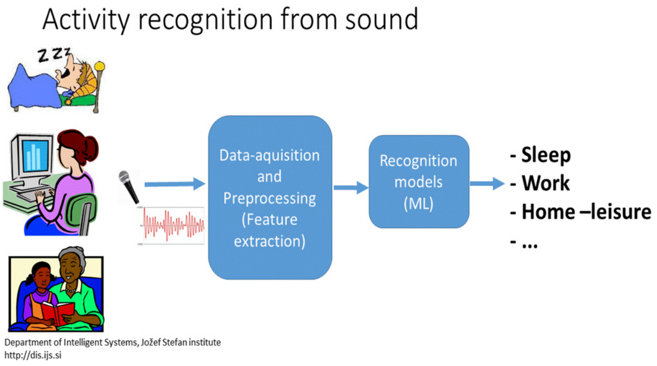

Activity recognition from sound
z z z
Data-aquisition and Preprocessing (Feature extraction)
Recognition models (ML)
- Sleep
- Work
- Home –leisure
- ...
Department of Intelligent Systems, Jožef Stefan institute
http://dis.ijs.si

Figure 2: The activity recognition process.

Then we extract sound features. Each feature is extracted in 20 ms long window. This is because sound signal is constantly changing and in such short window we assume that is not changing statistically much. We still have to have enough samples though, so shorter windows are inappropriate. We can also use window overlap so we lose less information. We used 20% of window overlap.

We extracted 19 types of features and we aggregated them in a window of one-minute length. We put together all recordings that were recorded in one minute, then we extracted each feature in 20 ms window and we aggregated it using mean and standard deviation, so we got a feature vector, which represented one minute. We also tried to aggregate for one second, but we got worse results.

Features were: Spectral centroid, Spectral rolloff point, Spectral flux, Compactness, Spectral variability, Root mean square, Fraction of low energy, Zero crossings, Strongest beat, Strength of strongest beat, Strongest frequency via FFT (Fast Fourier transform) maximum, MFCC’s (Mel frequency cepstrum coefficients) (13 coefficients), Linear predictive coding (LPC) (10 coefficients), Method of moments (5 features), Partial based spectral centroid, Partial based spectral flux, Peak based spectral smoothness, Area method of moments (10 features) and Area method of moments of MFCCs (10 features). Those features were aggregated using mean and standard deviation. We also added 10 Area moments of Area method of moments of MFCC’s. This sums up to 136 features. All features are explained in [4].

Since we have a lot of features, we use feature selection algorithms.

Finally, we use supervised machine learning techniques to build classifiers for our data.

4. EVALUATION

In this section we present an evaluation of our experiment.

We gathered recordings from 3 persons in their everyday living with smart phone's microphone. They labeled data with the following activities: "Sleep", "Exercise", "Work", "Eating", "Home - chores" and "Home - leisure".

Data was firstly intended for monitoring chronic patients.

We split data into training and test set. We were recording each person for 2 weeks and we used the first week as training set and the second week as test set.

We extracted features using open-source library jAudio [4], [5].

For feature selection and machine learning we used the open-source library Weka [6]. We used the feature selection algorithm ReliefF implemented in Weka on every person. We used 4 machine learning algorithms: SMO, J48, RandomForest and iBK, all with default parameters. We measured accuracies of all algorithms and then we used the best-performing algorithm and we measured F-measures for all the activities.

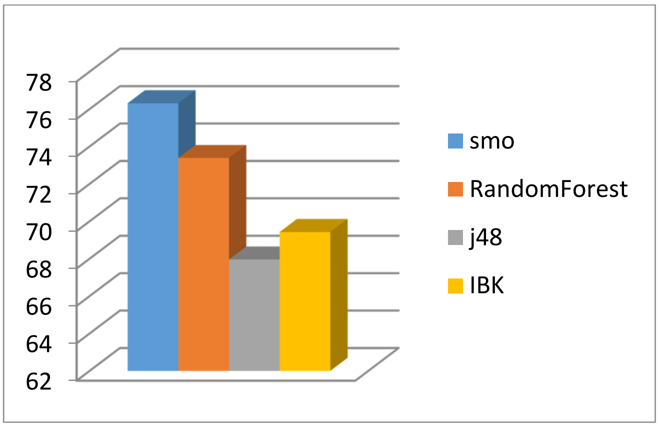


Figure 3: Average accuracy of all classifiers.

As can be seen in Figure 3, the best performing algorithm was SMO, which produced the highest accuracies on all the tested persons. The average accuracy of SMO is 76 %. In the second and third place are RandomForest and iBK with the average accuracies of 73 % and 69 % respectively. The worst was j48 with the average accuracy of 68 %.

In Figure 4 we can see the average F-measure per activity for the best performing algorithm SMO. The best recognized activity is "Sleep" with the average F-measure of 0.96, following by "Work" with 0.85. SMO detected "Eating" and "Exercise" relatively well with the average F-measure of 0.46 and 0.43, respectively. The remaining average F-measures for "Home - leisure" and "Home - chores" were 0.38 and 0.26, respectively.

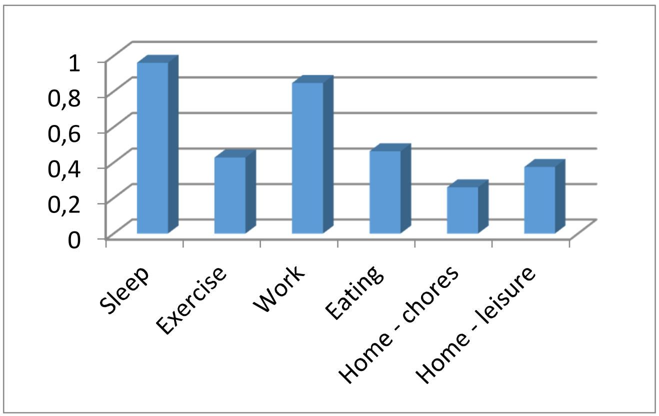


Figure 4: Average F-measure values of SMO.

Since we had 3 persons, we trained one best-performing classifier (SMO) for each person. We got a confusion matrix for each person and we summed all 3 matrices in one. It can be seen in Table 1. We can see that activity "Sleep" is almost flawless. We can also see that "Home - chores" is usually misclassified as "Home - leisure", which could be a consequence of similar sounds produced in a person's home during various activities. Due to the high number of instances labeled as “Work”, we got very good classification of "Work", but there are also many instances misclassified as "Work". We must take into account that recorded persons worked in the office, so many sounds are similar as in the home environment. We can conclude that for different activities, there can be many similar sounds, e.g. when person reads a book at home ("Home - leisure"), there can be silence as if the person took a nap ("Sleep"), so it is very challenging for classifiers to achieve high accuracies.

Table 1: Summed confusion matrix of all persons.

Sleep	**Exercise**	**Work**	**Eating**	**Home – chores**	**Home - Leisure**	⇦ **classified as**
198	0	1	0	0	5	**Sleep**
0	48	40	0	0	0	**Exercise**
0	18	1290	43	20	95	**Work**
0	2	48	50	7	10	**Eating**
10	6	92	3	66	44	**Home - chores**
7	0	61	5	31	216	**Home - leisure**

5. CONCLUSION

This paper presents a system and an approach to human activity recognition based on sound. The approach was tested on real-life recordings of three persons who annotated their activity for 2 weeks.

As outlined in Section 4 activity recognition from sound on 1 minute intervals may be challenging. There may be complete silence during different kinds of activities (e.g. sleep, work) or the recording may be dominated by speech. Therefore, it is difficult to achieve high accuracies in such settings.

Nevertheless, activity recognition from sound may be used for remote elderly-care. If we detect that the user was eating at usual times during the day, even though we do not have correct value about the eating period, we may conclude that user's state is normal. Having reliable sleep recognition, we may detect if the person is waking up during the night or if the period of sleep is lengthening, both of which may indicate a health problem. As future work, we need to record everyday living activities of the elderly, and test the system's capability to detect events that are critical for determining their health state.

References

[1] Eurostat, Retrieved September 2, 2014, from http://epp.eurostat.ec.europa.eu/tgm/table.do?tab=table&init=1&language=en&pcode=tsdde511&plugin=1.

[2] European commission, Horizon 2020, Health, demographic change and well being. Retrieved September 2, 2014, from http://ec.europa.eu/research/participants/portal/doc/call/h2020/common/1617611-part_8_health_v2.0_en.pdf

[3] M. Ziefle, S. Himmel, W. Wilkowska (2011) When your living space knows what you do: Acceptance of Medical Home Monitoring by Different Technologies, Lecture Notes in Computer Science, pp. 607-624.

[4] McEnnis, Daniel, Ichiro Fujinaga, Cory McKay, Philippe DePalle. 2005. "JAudio: A feature extraction library". ISMIR.

[5] McEnnis, Daniel, Ichiro Fujinaga. 2006. "jAudio: Improvements and additions". ISMIR.

[6] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

ARE HUMANS GETTING SMARTER DUE TO AI?

Matjaž Gams
Department of Intelligent Systems, Jožef Stefan Institute
Jamova cesta 39, 1000 Ljubljana, Slovenia
e-mail: matjaz.gams@ijs.si

ABSTRACT

Humans are getting smarter due to use of tools, in history because of mechanical tools and in recent decades due to information tools. The hypothesis in this paper goes a step further: that we are getting smarter due to use of AI. The thesis is indicated by solutions to three well-known logical paradoxes that have been recently resolved by the author of this paper: the unexpected hanging paradox, the Pinocchio paradox and the blue-eyes paradox. This paper is a bit shorter version of the Informatica paper on the same issue [19].

1. INTRODUCTION

Systematic measurements of the standard broad-spectrum IQ tests improve each decade. According to the Flynn effect [1] that humans are getting smarter and smarter. One theory claims that the increase of human intelligence is related to the use of information tools [2], which often progress exponentially over time [3]. In this paper we go a step further - that artificial intelligence (AI) influences human intelligence in a positive way as other influencing factors. We illustrate the hypothesis in Figure 1. The *y* axis is logarithmic in the scale. Therefore, the linear growth of computer skills on the graph corresponds to the exponential nature of Moore's law [4]. Basic human physical and mental characteristics, such as speed of movement, coordination or speed of human computing, have remained nearly constant in recent decades, as represented by the horizontal line in Figure 1.

Our first thesis is that the ability of humans to solve problems increases due to information tools such as computers, mobile devices with advanced software, and AI in particular (the bold top line in the Figure 1). Programs such as the Google browser may provide the greatest knowledge source available to humans, thereby representing an extension of our brains as do calculators in the field of arithmetic.

The stronger and more provocative hypothesis that humans are getting smarter on their own due to the AI comprehensions. After all, AI is about intelligence. In the AI community [5], it is generally accepted that AI progress is increasing and might even enable human civilization to take a quantitative leap [6].

Several opposing theories claim that humans actually perform worse on their own, since machines and tools have replaced humans' need to think on their own. We argue that while this effect may be valid for human physical properties related for example to obesity due to lack of physical activity, it is not the case in mental tasks. Another pessimistic viewpoint suggests that intelligent civilizations decline after reaching a certain development level (see Figure 1), possibly because of overpopulation, self-destruction or depletion of natural resources. This would explain why we have not yet detected alien civilizations, though the Drake's equation [7] indicates that many such civilizations should exist. This remains an open question.

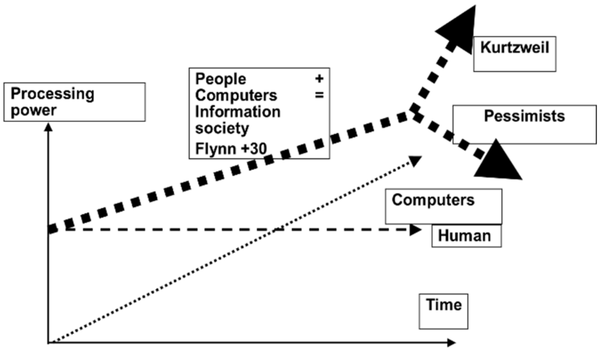


Figure 1: Current and predicted growth of computer and human computing capabilities to solve problems.

How to indicate that AI helps improve humans think better? If we can show that humans can solve logical puzzles that they were not able to solve until recently without computers, that would be a good indication of humans getting smarter on their own. An objection might be that just one solution of one puzzle is far too little to show anything. However, an indication it might be – at least to start a debate.

To demonstrate the idea, we analyze the unexpected hanging paradox [8, 9, 10, 11] and shortly mention a couple of other logical paradoxes.

2. THE LIAR PARADOX

For an introduction to logical paradoxes we quickly investigate the liar paradox, first published in [12]. According to [13] it was first formulated by the Greek philosopher Eubulides: "A man says that he is lying. Is what he says true or false?" This sentence is false when it is true. These days, the paradox is usually presented in the form "This sentence is false."

But today it is generally accepted that there is no true paradox, since the statement is simply false [14]. The contradiction is of the form “A and not A,” or “It is true and false.” In other words, **if a person always lies by definition, then that person is allowed to say only lies**. Therefore, such statements are simply not allowed, which means they are false.

We presented the liar paradox to analyze why humans had troubles with it before and why now it is seen as a trivial case. When faced with the liar paradox for the first time, humans fall into a loop of true/untrue derivations without observing that their thinking was already falsified by the declaration of the problem. It seems a valid logical problem, so humans apply logical reasoning. However, the declaration of the logical paradox was illogical at the start rendering logical reasoning meaningless.

In analogy, 1 + 1 = 2, and we all accept this as a true sentence without any hesitation. Yet, one liter of water and one liter of sugar do not combine to form two liters of sugar water. Therefore, using common logic/arithmetic in such a task is inappropriate from the start.

Which AI methods help us better understand such paradoxes? The principle and paradox of multiple-knowledge [15] tentatively explain why humans easily resolve such problems as the liar paradox. We use multiple ways of thinking not only in parallel, but also with several mental processes interacting together during problem-solving. Different processes propose different solutions, and the best one is selected. The basic difference in multiple-knowledge viewpoint compared to the classical ones occurs already at the level of neurons. The classical analogy of a neuron is a simple computing mechanism that produces 0/1 as output. In the multiple viewpoint, each neuron outputs 2^N possible outcomes, which can be demonstrated if N outputs from a single neuron are all connected to N inputs of another neuron. In summary, the multiple-knowledge principle claims that the human computing mechanism at the level of a neuron is already much more complex than commonly described, and even more so at the level of higher mental processes. Therefore, humans have no problems computing that one apple and one apple are two apples, and one liter of water and one liter of sugar is 1.6 liters of liquid and a mass of 2.25 kilograms, since they use multiple thinking. It is only that a person who logically encounters the sugar-water merge for the first time may claim that it will result in 2 liters of sugar water. However, after an explanation or experiment, humans comprehend the problem and have no future problems of this kind.

Another AI solution at hand uses contexts. In arithmetic, $1 + 1 = 2$. In merging liquids and solid materials, $1 + 1 \neq 2$. In the first case, the context was arithmetic and in the second case, merging liquids and solid materials. The contexts enable an important insight into the paradoxes such as the unexpected handing paradox.

3. THE UNEXPECTED HANGING PARADOX

The unexpected hanging paradox, also known as the hangman paradox, the unexpected exam paradox, the surprise test paradox, or the prediction paradox, yields no consensus on its precise nature, so a final correct solution has not yet been established [9]. This is a paradox about a person’s expectations about the timing of a future event that they are told will occur at some unexpected time [16]. The paradox has been described as follows [9]:

A judge tells a condemned prisoner that he will be hanged at noon on one weekday in the following week but that the execution will be a surprise to the prisoner. He will not know the day of the hanging until the executioner knocks on his cell door at noon that day. Having reflected on his sentence, the prisoner draws the conclusion that he will escape from the hanging. His reasoning is in several parts. He begins by concluding that the "surprise hanging" can't be on Friday, as if he hasn't been hanged by Thursday, there is only one day left - and so it won't be a surprise if he's hanged on Friday. Since the judge's sentence stipulated that the hanging would be a surprise to him, he concludes it cannot occur on Friday. He then reasons that the surprise hanging cannot be on Thursday either, because Friday has already been eliminated and if he hasn't been hanged by Wednesday night, the hanging must occur on Thursday, making a Thursday hanging not a surprise either. By similar reasoning he concludes that the hanging can also not occur on Wednesday, Tuesday or Monday. Joyfully he retires to his cell confident that the hanging will not occur at all. The next week, the executioner knocks on the prisoner's door at noon on Wednesday — which, despite all the above, was an utter surprise to him. Everything the judge said came true.

Evidently, the prisoner miscalculated, but how? Logically, the reasoning seems correct. While there have been many analyses and interpretations of the unexpected hanging paradox, there is no generally accepted solution. The paradox is interesting to study because it arouses interest in both laymen and scientists. Here, we provide a different analysis based on the viewpoint of cooperating AI agents [5, 16], contexts and multiple knowledge [15]. It might be the case that similar solutions were presented before, but it seems that AI knowledge disregards potential complications and provides a simple solution.

First we examine, which events are repeatable and which irreversible. The prediction of hanging on one out of five possible days is well defined through a real-life empirical fact of a human life being irreversibly terminated. However, the surprise is less clearly defined. If it denotes cognitive surprise, then the prisoner can be sure that the hanging will take place on the current day. No surprise is assured each new day, even on the first day, so hanging under the given conditions is not possible. Such an interpretation makes no sense. To avoid the prisoner being cognitively certain, the following modifications are often proposed [9]:

The prisoner will be hanged next week, and the date (of the hanging) will not be deductible in advance from the assumption that the hanging will occur during the week (A). The prisoner will be hanged next week and its date will not be deducible in advance using this statement as an axiom (B).

Logicians are able to show that statement (B) is self-contradictory, indicating that in this interpretation, the judge uttered a self-contradicting statement leading to a paradox.

Chow [10] presents a potential explanation through epistemological formulations suggesting that the unexpected hanging paradox is a more intricate version of Moore's paradox [9]:

A suitable analogy can be reached by reducing the length of the week to just one day. Then the judge's sentence becomes: "You will be hanged tomorrow, but you do not know that."

Now we can apply AI methods to analyze the paradox. First, the judge's statement is a one-sided contract from an AI agent viewpoint, defining a way of interacting and cooperating. As with any agreement/contract, it also has some mechanisms defining the consequences if one side violates the agreement. Since the judge unilaterally proclaimed the agreement, he can even violate it without any harm to him, whereas the prisoner's violations are punished according to the judge's will and corresponding regulations. For example, if the prisoner harms a warden, the deal is probably off, and the hanging can occur at the first opportunity, regardless of whether it is a surprise. This is an introductory indication that the hanging paradox is from the real world and that it matters, and is not just logical thinking. Even more important, it enables a valid conclusion that **any error in prisoner's actions releases the judge from his promise**.

Since the judge is the interpreter of the agreement, he can accept the weird viewpoint that it suffices that the prisoners claims a surprise to be released. However, the judge is supposed to be a smart person and there is no sense in such a viewpoint. The judge is also supposed to be an honest person and as long as the prisoner abides to the appropriate behavior, the judge will keep his word and presumably postpone the execution if the prisoner predicts the exact day of the hanging. Now, we come to the crucial reasonable definition of ambiguity, defined by the smart and honest judge. The term *deducible* now means that the prediction will be 100 percent guaranteed accurate about a one-time event (that is, hanging), so such **a prediction can be uttered only once a week, not each day anew**. Therefore, the prisoner has exactly one chance of not only predicting, but also **explaining with certainty to the judge**, why the hanging will occur on that particular day. The judge will have to be persuaded; that is, he will have to understand and accept the prisoner's line of reasoning. If not, the deal is off and the judge can choose any day while still keeping his word.

For easier understanding, consider that the prisoner is given a life-saving coupon on which he writes the predicted day and stores it in the judge's safe on Monday morning with the explanation attached. Obviously, the prisoner stands no chance if the judge orders handing on Monday. Namely, if the prisoner proposes Monday, he cannot provide a deducible explanation why the handing will happen on Monday. Yes, he will not be surprised in cognitive terms, but both a correct prediction and a deducible explanation are required in order to avoid hanging. The only chance to avoid hanging is to predict Friday and hope that he will not be hanged till Friday. (In this case, the judge could still object that, on Monday for example, the prisoner could not provide a plausible explanation for Friday. Yet, that would not be fair since, on Friday, the prisoner would indeed be sure of the judge coming into contradiction.) Even if the prisoner is allowed to deposit the one and only coupon on any day in the week, there is no major difference in terms of explanation in this paper. Again, if the prisoner is allowed to deposit the coupon each day anew, this formulation makes no sense.

We can further explain the error in the prisoner's line of reasoning by assuming that instead of giving his ruling five days in advance, he gave it on Thursday morning, leaving a two-day opportunity. Since the prisoner could use the single pardon (remember: *deducible* for a one-time event means one prediction once) and save himself on Friday, he concludes that Thursday is the only day left and cashes in his only coupon with a 100 percent certain logical explanation on Thursday. However, in this case the judge could carry out the hanging on Friday. Why? Because the prisoner provided the only 100 percent certain prediction in the form of a single life-saving coupon on Thursday, which means that on Friday he could not deliver the coupon. In other words, the prisoner wrongly predicted the hanging day and therefore violated the agreement.

The situation on Thursday is similar to the situation on Monday. Even if the judge knocks on the door on Thursday, and the prisoner correctly predicted Thursday, he still could not provide a 100 percent certain explanation why the hanging would occur on Thursday since the judge could come back on Friday as described in the above text; therefore, the judge can proceed also on Thursday or Friday without violating his proclamation.

4. DISCUSSION

Wikipedia offers the following statement regarding the unexpected hanging paradox [9]:

There has been considerable debate between the logical school, which uses mathematical language, and the epistemological school, which employs concepts such as knowledge, belief and memory, over which formulation is correct.

According to other publications [8], this statement correctly describes the current state of scientific literature and the human mind.

To some degree, solutions similar to the one presented in this paper have already been published [8, 9]. However, they have not been generally accepted and, in particular, have not been presented through AI means. Namely, AI enables clearer explanation such as:

The error in the prisoner's line of reasoning occurs when extending his induction from Friday to Thursday, as noted earlier, but the explanation in this paper differs. The correct conclusion about Friday is not:

"Hanging on Friday is not possible" (C), but:

"If not hanged till Friday and the single prediction with explanation was not applied for any other day before, then hanging on Friday is not possible." (D)

The first condition in (D) is part of common knowledge. The second condition in (D) comes from common sense about one-sided agreements: every breach of the agreement can cause termination of it. The two conditions reveal why humans have a much harder time understanding the hanging paradox, compared to the liar paradox. The conditions are related to the concepts and **interpretation** of ***time* and *deducibility*** and should be applied **simultaneously**, whereas only one insight is needed in the liar paradox. In AI, this phenomenon is well known as the **context-sensitive reasoning** in agents, which was first presented in [18] and has been used extensively in recent years. Here, as in real life, under one context the same line of reasoning can lead to a different conclusion compared to the conclusion under another context (remember the sugar water). But one can also treat the conditions in statement (D) as logical conditions, in which case the context can serve for easier understanding. The same applies to the author of this paper: Although he has been familiar with the hanging paradox for decades, the solution at hand emerged only when the insight related to the contexts appeared.

Returning to the motivation for analysis of the unexpected hanging paradox, the example was intended to show that humans have mentally progressed to see the trick in the hanging paradox, similar to how people became too smart to be deceived by the liar paradox.

This new approach has also been used to solve several other paradoxes, such as the blue-eyes paradox and the Pinocchio paradox. Analyses of these paradoxes are being submitted to other journals.

In summary, the explanation of the hanging paradox and the difficulty for human paradox solvers resembles those of the liar paradox before solving it beyond doubt. It turns out that **both paradoxes are not truly paradoxical**; instead, they describe a logical problem in a way that a human using logical methods cannot resolve the problem. Similar to the untrue assumption that a liar can utter a true statement, the unexpected hanging paradox in the prisoner's line of reasoning exploits **two misconceptions**. The first is that a 100 percent accurate prediction for a single event can be uttered more than once (through a vague definition of "surprise") and the second that a conclusion that is valid at one time is also valid during another time span. Due to the simplicity of the AI-based explanation in this paper, there is no need to provide additional logical, epistemological, or philosophical mechanisms to explain the failure of the prisoner's line of reasoning.

This paper provides an AI-based explanation of the hanging paradox for humans in natural language, while formal explanations remain a research challenge. The formal analysis have already been designed for the Pinocchio paradox whereas the blue-eyed paradox has not yet been formally explained, only in a way similar to this paper.

Acknowledgements

The author wishes to thank several members of the Department of Intelligent Systems, particularly Boštjan Kaluža, Mitja Luštrek, and Tone Gradišek for their valuable remarks. Special thanks are also due to Angelo Montanari, Stephen Muggleton, and Eva Černčic for contributions on this and other logic problems.

References

[1] Teasdale, T W, Owen, D. R. (2005). A long-term rise and recent decline in intelligence test performance: The Flynn Effect in reverse. Personality and Individual Differences, Volume 39, Issue 4, September 2005, Pages 837‑843, Elsevier.

[2] Flynn, J. R. (2009). What Is Intelligence: Beyond the Flynn Effect. Cambridge, UK: CambridgeUniversity Press.

[3] Computing laws revisited (2013). Computer 46/12. [4] Moore, G.E. (1965). Cramming more components onto integrated circuits. Electronics Magazine, 4.

[5] Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI'13) (2013). Beijing, China.

[6] Kurzweil, R. (2005). The Singularity is Near. New York: Viking Books.

[7] Dean, T. (2009). A review of the Drake equation. Cosmos Magazine.

[8] Wolfram A. (2014). http://mathworld. wolfram.com/UnexpectedHangingParadox.html

[9] Unexpected hanging paradox, Wikipedia (2014). https://en.wikipedia.org/w/index.php?title=Unexpected_hanging_paradox&oldid=611543144, June 2014

[10] Chow, T.Y. (1998). The surprise examination or unexpected hanging paradox. *American Mathematical Monthly* 105:41–51.

[11] Sober, E. (1998). To give a surprise exam, use game theory. *Synthese* 115:355–73.

[12] O'Connor, D.J. (1948). Pragmatic paradoxes. *Mind* 57: 358–9.

[13] Beall, J.C., Glanzberg, M. (2013). In Edward N. Zalta, E.N. (eds.), *The Stanford Encyclopedia of Philosophy*.

[14] Prior, A.N. (1976). *Papers in Logic and Ethics*. Duckworth.

[15] Gams, M. (2001). *Weak Intelligence: Through the Principle and Paradox of Multiple Knowledge.* New York: Nova Science Publishers, Inc.

[16] Sorensen, R. A. (1988). *Blindspots*. Oxford, UK: Clarendon Press.

[17] Young, H.P. (2007). The possible and the impossible in multi-agent learning. *Artificial Intelligence* 171/7.

[18] Turner, R.M. (1993). Context-sensitive Reasoning for Autonomous Agents and Cooperative Distributed Problem Solving, In *Proceedings of the IJCAI Workshop on Using Knowledge in its Context,* Chambery, France.

[19] Gams, M (2014). The Unexpected Hanging Paradox from an AI Viewpoint, Informatica 38, 181–185.

DEVELOPING A SENSOR FIRMWARE APPLICATION FOR REAL-LIFE USAGE

Hristijan Gjoreski, Mitja Luštrek, Matjaž Gams
Department of Intelligent Systems, Jožef Stefan Institute,
Jožef Stefan International Postgraduate School,
e-mail: {hristijan.gjoreski, mitja.lustrek, matjaz.gams}@ijs.si

ABSTRACT

In recent years the demand for intelligent systems that support the life of the elderly is increasing. In order to provide an appropriate support, these systems should constantly monitor the user with sensors. However, using sensors in real-life situations is a challenging task, mainly because of the constraints in the sensor energy consumption (battery life) and memory capacity for storing the sensors data. In this paper we present an example of a sensor communication protocol developed for the Shimmer accelerometers, so that they can be used in in real-life situations, i.e., constantly monitoring the user during a normal day. A custom firmware application is developed, which has several functionalities: real-time data streaming through Bluetooth, data logging into internal microSD card, sending the stored data to a Bluetooth-enabled device and detecting when the sensor is put on and off a charging dock.

1 INTRODUCTION

The world's population is aging rapidly, threatening to overwhelm the society's capacity to take care of its elderly members. The percentage of persons aged 65 or over in developed countries is projected to rise from 7.5% in 2009 to 16% in 2050 [1]. This is driving the development of innovative ambient assisted living (AAL) technologies to help the elderly live independently for longer and with minimal support from the working-age population [2][3]. To provide timely and appropriate assistance, AAL systems must monitor the user by using ambient (environmental) and/or wearable sensors. With the recent development of the sensors technology, the wearable sensors are gaining attraction and can measure lot of different user-related parameters: location, activity, physiological, etc. Examples of them include: GPS, accelerometers, gyroscopes, heart-rate sensors, breath-rate sensors, etc.

In order to be used in everyday life situations, these sensors have to able to constantly monitor the user during the day. However, often this is a challenging task mostly due to battery consumption constraints and memory storage capacity. In this study we present a sensing protocol for the Shimmer accelerometer sensors, so they can be used to constantly monitor the user during the day. A custom firmware TinyOS application was developed in order to satisfy the user's requirements and the sensors limitations. It implements two modes of operation: real-time data sending and logging the data in the internal memory and sending it offline in batches.

The motivation and the context of the study is the CHIRON project (Cyclic and person-centric health management: Integrated approach for home, mobile and clinical environments) [4]. It is a European research project of the ARTEMIS JU Program with 27 project partners. It includes industry partners (large companies and SMEs), research and the academic institutions, and also medical institutions. The project addresses one of the today's societal challenges i.e., "effective and affordable healthcare and wellbeing". CHIRON combines state-of-the art technologies and innovative solutions into an integrated framework of embedded systems for effective and person-centered health management throughout the complete healthcare cycle, from primary prevention (person still healthy) to secondary prevention (initial symptoms or discomfort) and tertiary prevention (disease diagnosis, treatment and rehabilitation) in various domains: at home, in nomadic environments and in clinical settings.

2 SENSORS

In the CHIRON project, two Shimmer accelerometers are used to monitor the user's activities. The sensor platform is based on the Shimmer Wireless Sensor Network (WSN) module. It is based on a T.I. MSP430F1611 microcontroller, which operates at a maximum frequency of 8 MHz and is equipped with 10Kb RAM and 48 Kb of Flash. Wireless communication is achieved either with Bluetooth v2 (BT – RN-42 module) or through IEEE 802.15.4 (T.I. CC2420 module.). In our study we used the standard BT v2 in order to easily connect it with a smartphone. For storage purposes, the Shimmer platform is equipped with an integrated 2GB microSD card, which is used in normal operation mode to store sensor readings [5]. The power supply is comprised of a 450mAh rechargeable Li-ion battery.

The firmware of the Shimmer platform is based on the open-source TinyOS operating system [6]. It uses the NesC programming language, which is a light-weight version of C. TinyOS/NesC is dedicated for low-power wireless sensors and allows many sensor platforms with a heterogeneous set of hardware devices to be programmed and controlled (microcontroller, sensors, SD cards, etc.).

The TinyOS in the Shimmer sensors follows three-layer abstraction architecture. At the bottom is the Hardware Presentation Layer (HPL) which allows access to input/output pins or registers of the hardware devices. Next, the Hardware Abstraction Layer (HAL) allows configuring more complex functionality in order to communicate with external sensors or resources implemented in the platform. The top layer is the Hardware Independent Layer (HIL), which permits to read the sensor data independently of the digital communication bus.

Each layer communicates with the adjacent ones trough interfaces, either generic or hardware specific. As the TinyOS is an event-driven operating system, the interface call commands that are addressed to the lower layer. These commands are answered from the lower layers by signaling events. In our case, the Shimmer sensor platform, the HPL and the HAL layers are already available and for its internal resources, as the accelerometer, the SD card or the Bluetooth radio, the HIL layer is also implemented.

Once the layers are implemented, a firmware application is developed. In our case, the application is based on the specifications (sensing protocol) provided by doctors in the CHIRON project. The firmware application and the sensing protocol are discussed in the next section.

3 SENSING PROTOCOL AND FIRMWARE APPLICATION

In order to explain the sensing protocol (shown in Figure 1), let us consider the following scenario. The user wakes up and takes the two accelerometers from the charging dock. Once they are taken out from the dock, the sensors have to start sensing. The user attaches the sensors in the wearable garment (e.g., chest and thigh elastic straps) and performs an initialization activity sequence. This sequence is performed in order to ensure if the sensors have the right orientation (important for the post-processing of the data). The orientation checking lasts for a few minutes, during which the sensor data is streamed in real-time to a smartphone application. Once the smartphone confirms that the setup is all right, the user continues with his everyday activities. During this period the sensors log the data locally to a microSD card. At the end of the day, the user takes out the sensors, puts them to the charging docks, and goes to sleep. During the sleep, the sensors are charged and all the data is transferred to the processing unit.

The scenario shows that the battery life of the sensors should last at least 16 hours (the active period of a normal day) and the sensors should be able to receive commands from a smartphone through Bluetooth. Our tests showed that if the standard firmware application is used (real-time data sending using BT), the battery will last around 6 hours, which is not sufficient for the whole day. Furthermore, with this approach there should be a constant BT connection between the smartphone and the sensors, which is highly unlikely in real-life scenario. In order to achieve these functionalities we created a custom firmware application which has two modes of operations: real-time data sending and logging the data in microSD card and sending it for offline analysis. The application is based on the two standard Shimmer firmware applications, which are publicly available: real-time data acquisition ("BoilerPlate.ihex") and data logging ("JustFATLogging.ihex").

The original logging application (JustFATLogging.ihex) has one main function, to log the acceleration data on the microSD card. The start of the logging is triggered when the sensor is removed from a dock station and the end of logging is triggered when the sensor is put back on the dock station. The data can be accessed only through the USB port of the docking station. We used this firmware application as a base for further development.

First, we added the Bluetooth functionality in order to allow wireless communication between the sensor and the smartphone. However, the activation of the Bluetooth significantly decreased the sensor's battery life. Therefore, we modified the application so the Bluetooth is activated only when the sensor is put back on the charger. During the charging time, the smartphone sends a command and collects the logged data. When the user decides to mount the sensors he/she gives a command to start logging and to turns off the Bluetooth. Thus, during the logging process the Bluetooth is off and there is no communication between the smartphone and the sensor.

For the acceleration data, it is really important how the sensor is mounted, i.e., the sensor orientation must be the same for every recording. In order to check the orientation of the sensors, an algorithm analyzes the data during some predefined activities, e.g., standing and lying, and accordingly gives a notification to the user if the sensors are mounted in the correct way. To allow this data analysis, the data has to be analyzed in real-time, therefore we added also this functionality, i.e., *real-time transmission*. The real-time data acquisition is performed before the *start logging* command is sent.

In addition, two more functionalities were implemented: deleting a log file (*delete log*), and checking the availability of a log file (*is log available*).

The final modification is related to the timestamps of the data samples and data synchronization between different sensors. In order to synchronize the data between the sensors, one must know the absolute timestamp of the data samples. In our case, we used the timestamp of the start of the logging and the time difference between consecutive data samples. The sensor's internal crystal clock is used for estimating the time difference between consecutive data samples. Thus, each data sample is labeled with a timestamp provided by the clock. The starting timestamp is sent by the smartphone with each start logging command. Using the starting timestamp and the internal counter's timestamps, the smartphone was able to reconstruct an absolute timestamp for each data sample.

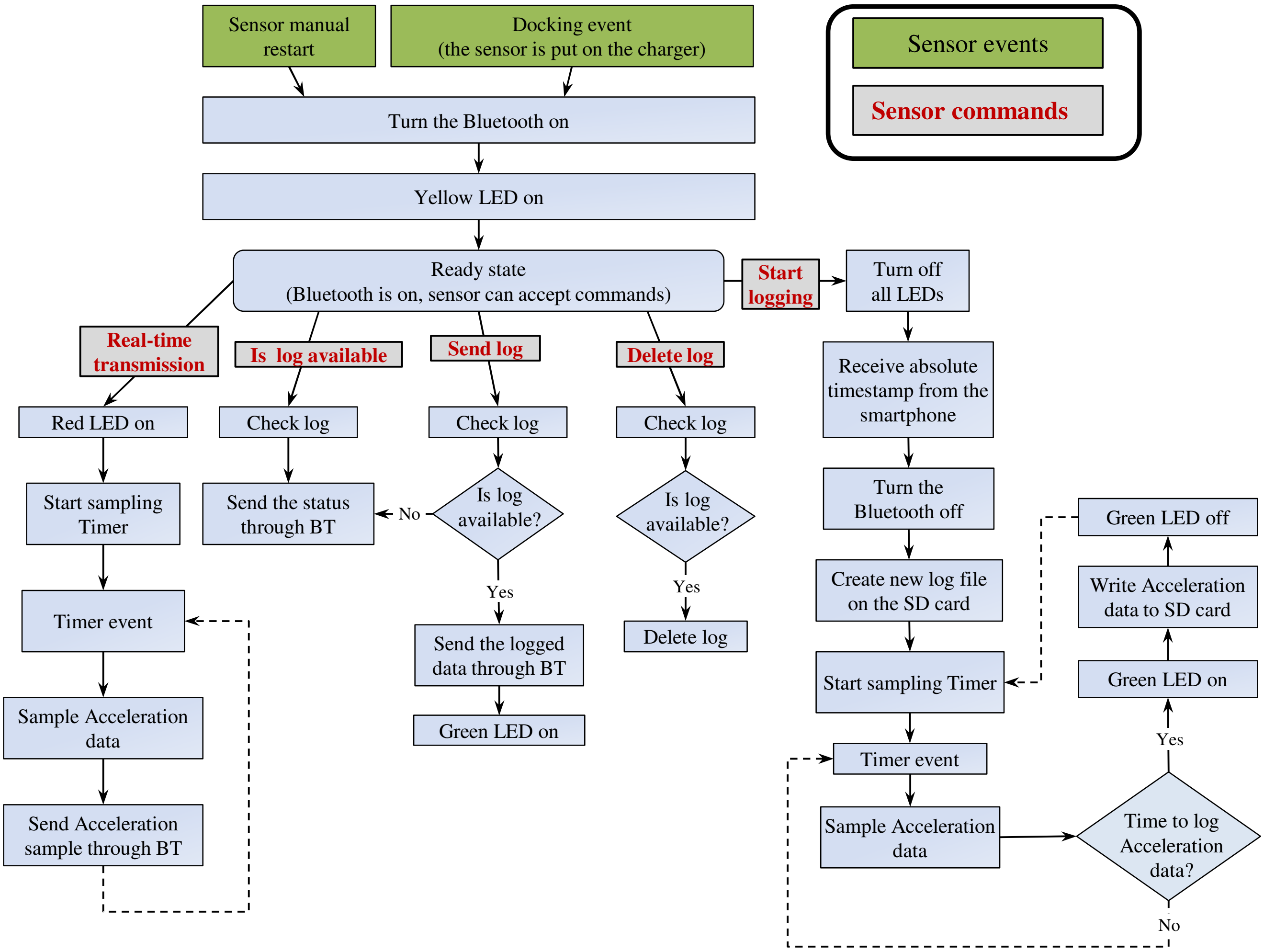


Figure 1. Sensor Firmware Flowchart.

In theory, the internal Shimmer crystal clock (Epson FC-135 32.7680KA-A3) tolerance is ±20 ppm, which results in 1.8 seconds maximal drift in 24 hours. In the worst case scenario, when two sensors have different drift direction (+ or -), the time difference is 3.6 seconds, which is acceptable for the project's requirements. Several practical tests were performed and confirmed the theoretical analysis, i.e., the measured drift was in the range of 1 second for a whole day recording.

4 SENSOR COMMANDS AND EVENTS

Table 1 describes the commands that can be received by the sensor application firmware. These commands can be sent by any BT-equipped device.

Table 1. Sensor commands.

Real-time transmission	The sensor samples accelerometer data at 50Hz and sends each sample to the smartphone.
Start logging	The sensor stops the Real-time sending, and waits for the absolute real timestamp from the smartphone. This timestamp is written at the beginning of each log file and is used as a reference point for reconstructing the timestamp for each data sample. After that, the sensor starts logging the accelerometer data. Additionally, the Bluetooth is turned off; there is no communication between the sensor and the smartphone during the logging process.
Is log available	The sensors checks if the log file is available for sending and sends the status.
Send log	The sensor sends the logged file. First, the absolute timestamp is sent, and then the accelerometer data samples are sent.
Delete log	The sensor deletes the log file.

Table 2 describes the events that are detected by the application firmware.

Table 2. Events that can be detected by the sensor's application firmware

Docking event (the sensor is put on the charging dock)	The sensor stops the logging and turns on the Bluetooth. Yellow LED is turned on, representing that the log file is ready to be sent.
Sensor manual restart	The sensor restarts to the initial state. That is, stops the logging and turns on the Bluetooth. Yellow LED is turned on, representing that the log file is ready to be sent.

5 THEORETICAL AND EMPIRICAL TESTS

After developing the firmware application we performed several theoretical and empirical tests. First, we analyzed the amount of data expected to be generated on a daily basis. The MSP430 A/D channels perform 12-bit (2 bytes of storage) digitization and that a 16 bit (2 bytes) timestamp is stored for each sample. Table 2 summarizes our projections based on the sampling frequency of every sensor. Based on these calculations the total amount of data for 12-16h of daily use should be 114Mb – 152.4Mb. This amount of data does not pose any issue in any operational mode, since in the real-time scenario BT can achieve data rates up to 300kbps which is more than adequate for the amount of data generated per second and in the logging operating mode the microSD cards on the modules have more than enough capacity to store the generated data.

Table 3. The amount of data generated by the accelerometer.

Sampling Frequency (Hz)	**Data per second (KB/s)**	**Data per hour (MB/h)**
50	0.78	2.74

The energy consumption analysis of the Shimmer platform, presented by Burns et. al. [5], designates that the accelerometer draws 1.6mA when sampled at 50Hz. When the sensor streams accelerometer data in real-time through the BT the consumption increases to 5.2 mA. From this analysis, it is safe to assume, that since the same hardware equipped with a 450mAh battery is used the clinical requirement of 6-8h data logging (in the storing mode) or an adequate amount of time (around 1h) for live streaming (streaming mode) can be easily met, provided that the module's batteries are fully charged at the beginning of sensing. Table 4 lists the average battery lifetime (full battery drainage period) obtained for the two modes of operation from a series of experiments.

Table 4. Average working time for the real-time (Bluetooth is active) and the logging mode (Bluetooth is not active; the sensor is logging in the SD card).

Sampling Frequency (Hz)	**Real-time mode**	**SD-logging mode**
50	6h 30m	14 days

6 CONCLUSION

In this paper we showed how one can overcome the sensor limitations (battery life and memory storage) by creating a custom firmware application and adjusting it to real-life situations. We presented a sensing protocol and sensor firmware application developed for the Shimmer accelerometers. The protocol was created so that the sensors can be used in in real-life situations, i.e., constantly monitoring the user during a normal day. The developed custom firmware application has several functionalities: real-time data streaming through Bluetooth, data logging into internal microSD card, sending the stored data to a Bluetooth-enabled device, and detecting when the sensor is put on and off a charging dock.

Acknowledgement

This work was partly supported by the Slovene Human Resources Development and Scholarship funds and partly by the CHIRON project - ARTEMIS Joint Undertaking, under grant agreement No. 2009-1-100228.

References

[1] United Nations 2009, World population ageing, Report

[2] A. Bourouis, M. Feham, A. Bouchachia, "A new architecture of a ubiquitous health monitoring system: a prototype of cloud mobile health monitoring system," The Computing Research Repository, 2012.

[3] M. Luštrek, B. Kaluža, B. Cvetković, E. Dovgan, H. Gjoreski, V. Mirchevska, M. Gams, "Confidence: ubiquitous care system to support independent living" DEMO at European Conference on Artificial Intelligence, pp. 1013-1014, 2012.

[4] The CHIRON project: http://www.chiron-project.eu/

[5] A. Burns, B. R. Greene, M. J. McGrath, T. J. O'Shea, B. Kuris, S. M. Ayer, F. Stroiescu, V. Cionca, "SHIMMER™ – A Wireless Sensor Platform for Noninvasive Biomedical Research," IEEE Sens. J., vol.10, no.9, pp.1527- 1534, 2010.

[6] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, D. Culler. TinyOS: An operating system for sensor networks. In Werner Weber, JanM Rabaey, and Emile Aarts, editors, Ambient Intelligence, chapter 7, pp. 115–148, 2005.

AUTOMATIC RECOGNITION OF EMOTIONS FROM SPEECH

Martin Gjoreski[1], *Hristijan Gjoreski*[2], *Andrea Kulakov*[1]
[1]Faculty of Computer Science and Engineering, Rugjer Boshkovikj 16, 1000 Skopje, Macedonia;
[2]Department of Intelligent Systems, Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia
e-mail: martin.gjoreski@gmail.com, hristijan.gjoreski@ijs.si, andrea.kulakov@finki.ukim.mk

ABSTRACT

This paper presents an approach to recognition of human emotions from speech. Seven emotions are recognized: anger, fear, sadness, happiness, boredom, disgust and neutral. The approach is applied on a speech database, which consists of simulated and annotated utterances. First, numerical features are extracted from the sound database by using audio feature extractor. Next, the extracted features are standardized. Then, feature selection methods are used to select the most relevant features. Finally, a classification model is trained to recognize the emotions. Three classification algorithms are tested, with SVM yielding the highest accuracy of 89% and 82% using the 10 fold cross-validation and Leave-One-Speaker-Out techniques, respectively. "Sadness" is the emotion which is recognized with highest accuracy.

1 INTRODUCTION

Human capabilities for perception, adaptation and learning about the surroundings are often three main compounds of the definition about what intelligent behavior is. In the last few decades there are many studies suggesting that one very important compound is left out of this definition about intelligent behavior. That compound is emotional intelligence. Emotional intelligence is the ability of one to feel, express, regulate his own, to recognize and handle the emotional state of others. In psychology the emotional state is defined as complex state that results in psychological and physiological changes that influence our behaving and thinking [1].

With the recent advancements of the technology and the growing research areas like machine learning, audio processing and speech processing, the emotional states will be inevitable part of the human-computer interaction. There are more and more studies that are working on providing the computers with abilities like recognizing, interpretation and simulation of emotional states.

In this research we present an approach for automatic recognition of emotions from speech. The goal is to recognize the emotional state that is experiencing the speaker. Furthermore, the focus is on how something is said, and not what is said. Besides this approach where only the speaker's voice is analyzed, there are several different approaches for recognizing the emotional state. In some approaches the voice and the spoken words are analyzed [2]. Some are focused only on the facial expressions [3]. Some are analyzing the reactions in the human brain for different emotional states [4]. Also there are combined approaches where combination of the mentioned approaches is used [5].

In studies where human emotions are analyzed mainly two methodologies are used. In the first methodology the emotions are viewed as discrete and completely distinct classes that are universally recognized [6]. In the second methodology the emotional states are represented in 2D or 3D space where parameters like emotional distance, level of activeness, level of dominance and level of pleasure can be observed [7]. In this research the discrete methodology will be used, so the emotional states will be represented as 7 classes: anger, fear, sadness, happiness, boredom, disgust and neutral.

The remainder of this paper is organized as follows. Next section is a brief overview of speech emotion analysis. Then, the methodology used for the process of emotion classification is presented. In the next section, the experimental setup and the results are presented. Finally, the conclusion and a brief discussion about the results is given.

2 SPEECH EMOTION ANALYSIS

Speech emotion analysis refers to usage of methods to extract vocal cues from speech as a marker for emotional state, mood or stress. The main assumption is that there are objectively measureable cues that can be used for predicting the emotional state of the speaker. This assumption is quite reasonable since the emotional states arouse physiological reactions that affect the process of speech production. For example, the emotional state of fear usually initiates rapid heartbeat, rapid breathing, sweating and muscle tension. As a result of these physiological activities there are changes in the vibration of the vocal folds and the shape of the vocal tract. All of this affects the vocal characteristics of the speech which allows to the listener to recognize the emotional state that the speaker is experiencing [8]. The basic speech audio features that are used for speech emotion recognition are: fundamental frequency (human perception for fundamental frequency is pith), power, intensity (human perception for intensity is loudness), duration features (ex. rate of speaking) and vocal perturbations. The main question is: Are there any objective voice feature profiles that can be used for speaker emotion recognition? A lot studies are done for the sake of providing such feature profiles that can be used for representation of the emotions, but not always the results are consistent. For some basic problems like distinguishing normal speech from angry speech or distinguishing normal speech from bored speech the experimental results converge [9]. The problem arises when we have to distinguish emotional states like anger from happiness or fear from happiness. By using the basic speech audio features for describing these emotional states, the feature profiles will be quite similar so distinguishing them is hard.

In the last few years, new method is introduced where static feature vectors are obtained by using so called acoustic Low-Level Descriptors (LLDs) and descriptive statistical functionals [10]. By using this approach a big number of large feature vectors is obtained. The downside is that not all of the feature vectors are of good value, especially not for emotion recognition. For that reason a feature selection method is often used.

3 THE APPROACH

Figure 1 shows the whole process of the speech emotion classification used in this research. An emotional speech database is used, which consists of simulated and annotated utterances. Next, feature extraction is performed by using open source feature extractor. Then, the extracted features are standardized. After standardization, feature selection methods are used for decreasing the number of features and selecting only the most relevant ones. Finally, the emotion recognition is performed by a classification model.

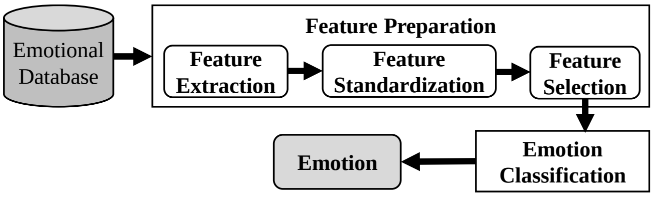


Figure 1: *Scheme for speech emotion classification.*

3.1 Emotional Database

There are several emotional speech databases that are extensively used in the literature [11]: German, English, Japanese, Spanish, Chinese, Russian, Dutch etc. One of the main characteristics of an emotional speech database is the type of the speech: whether it is simulated or it is extracted from real life situations. The advantage of having a simulated speech is that the researcher has a complete control over the emotion that it is expressed and complete control over the quality of the audio. However, the disadvantage is that there is loss in the level of naturalness and spontaneity. On the other hand, the non-simulated emotional databases consist of a speech that is extracted from real life scenarios like call-centers, interviews, meetings, movies, short videos and similar situations where the naturalness and spontaneity is kept. The disadvantage is that in these databases there is not a complete control over the expressed emotions. Also the low quality of the audio can be problem.

For this research the Berlin emotional speech database [12] is used. It consists of 535 audio files, where 10 actors (5 male and 5 female) are pronouncing 10 sentences (5 short and 5 long). The sentences are chosen so that all 7 emotions that we are analyzing can be expressed. The database is additionally checked for naturalness by a human expert. The utterances that were rated with more than 60% naturalness and from which the expressed emotion was recognized with more than 80%, were included in the final database.

3.2 Feature Preparation

The feature extractor tool used in this research is openSMILE (Open Speech and Music Interpretation by Large Space Extraction) [13]. It is a tool for signal processing and machine learning. We extracted 1582 features in total [14]. The LLDs that openSMILE is using are computed from basic features (pitch, loudness, voice quality) or representations of the audio signal (cepstrum, linear predictive coding).

On these LLDs functionals are applied and static feature vectors are produced, therefore static classifiers can be used. The functionals that are applied are: extremes (position of mix/min value), statistical moments (first to forth), percentiles (ex. the first quartile), duration (ex. percentage of time the signal is above threshold) and regression (ex. the offset of a linear approximation of the contour).

After the feature extraction the feature vectors are standardized so the distribution of the values of the feature vectors is with mean equal to 0 and standard deviation equal to 1. Next, a method for feature selection is used. Features are ranked with algorithms for feature ranking and experiments are done with varying number of top ranked features. For ranking the features two different algorithms are used, gain ratio [15] and ReliefF [16]. Both algorithms are used as they are implemented in Orange software packet for machine learning and data mining [17].

3.3 Emotion Classification

Once the features are extracted, selected and standardized, they are used to form the feature vector database. That is a database in which each data sample is an instance, i.e., feature vector. Additionally, each instance is labeled with the emotion. After this the instances are used to train a classification model in order to recognize emotions out of a speech data.

4 EXPERIMENTS

Three types of experiments are performed. In the first type, tests for comparison of three classification algorithms are done. The algorithm with the highest accuracy is further evaluated with 2 evaluation techniques: 10 fold cross-validation and Leave-One-Speaker-Out (LOSO) cross-validation.

4.1 Comparison of Classification Algorithms

Three classification algorithms are compared: KNN [18], SVM [19] and Naïve Bayes [20]. They are used as implemented in the Orange machine learning toolkit. The data is split 70-30, i.e., 70% of the data is used as training, and the remaining 30% is used for testing. Tests are performed with varying number (50, 100, 200, 300, 400, 500, 750, 1000 and 1582) of top ranked features by gain ratio. The results (shown in Figure 2) show that the SVM has the highest accuracy, i.e., 91% when the top ranked 500 features are used. By using the top ranked 300 features the drops to 88%.

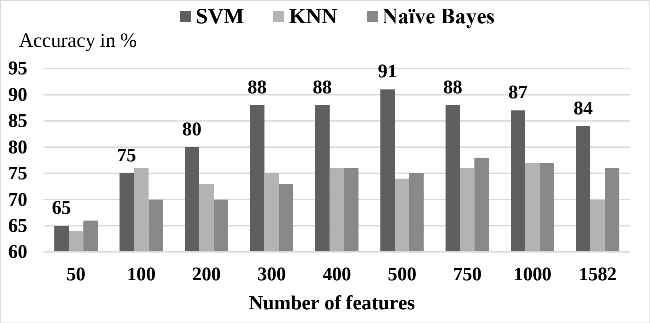


Figure 2: *SVM, KNN and Naïve Bayes classification accuracy for varying number of features.*

4.2 10 Fold Cross-Validation

We further evaluated the SVM with the 10 fold cross-validation technique. The results are shown in Figure 3. The highest accuracy of 89% is obtained by using top ranked 750 features. By using the top ranked 300 features the average accuracy is 87%, which is significantly high performance with such a low number of features.

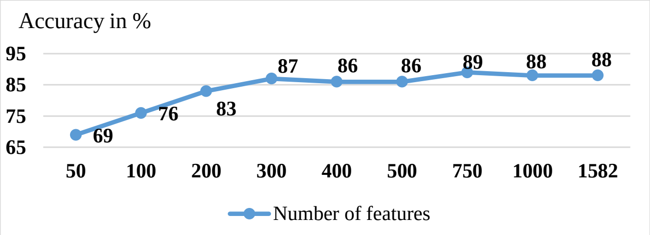


Figure 3: *SVM classification accuracy for 10 fold cross-validation with varying number of features.*

Additional analysis of the performance is performed by analyzing the recognition results for each emotion individually. The results achieved for the top ranked 750 features are shown in Figure 4. The highest accuracy per class is achieved for the class “sadness” (97%). On the contrary, the lowest accuracy per class is achieved for the class “happiness” (68%).

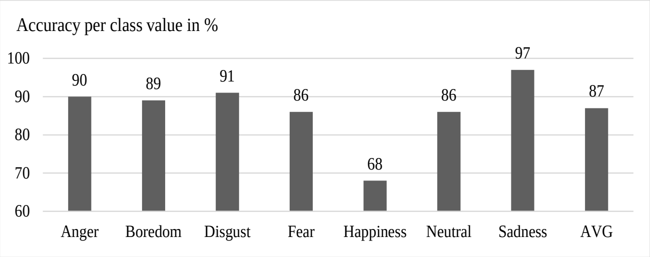


Figure 4: *SVM accuracy per class for 10 fold cross-validation with top ranked 750 features.*

4.3 Leave-One-Speaker-Out Cross-Validation

If the system for speech emotion recognition is supposed to work in an environment where it does not have any information about the speaker, LOSO is the best approach for testing the accuracy of the system.

The LOSO validation approach means that the train data consists of 9 speakers and the remaining one is used for testing. This is repeated 10 times, each time using different speaker’s data for testing. Figure 5 shows the results that are obtained with the LOSO technique. The testing speaker is represented on the x-axis. The varying color represents the number of top ranked features (by ReliefF) used. The highest average accuracy of 82% is obtained by using top ranked 1000 features. Also we can see that the accuracy depends mainly from the speaker that is used as test data.

For the experiments about the accuracy per class for each of the 7 emotional states, top ranked 1000 features (by ReliefF) are used. The results are shown in Figure 6. The highest accuracy per class of 94% was achieved for the class “sadness” and the lowest accuracy per class of 70% was achieved for the class “fear”.

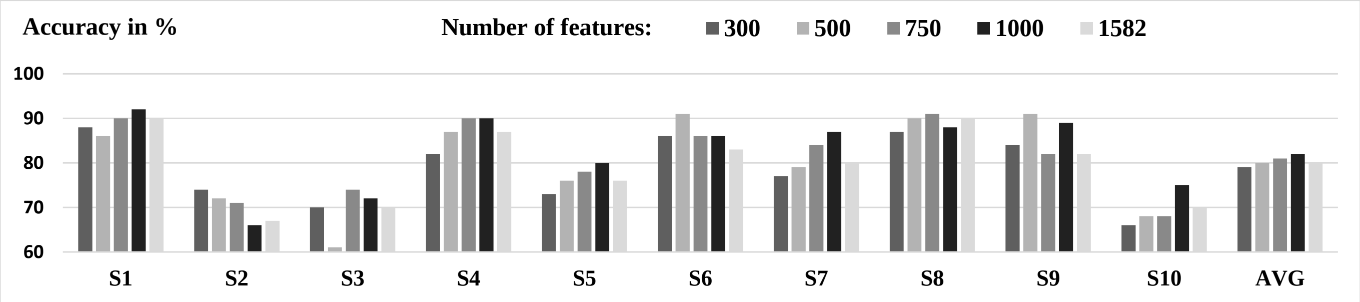


Figure 5: *SVM classification accuracy for LOSO with varying number of features*

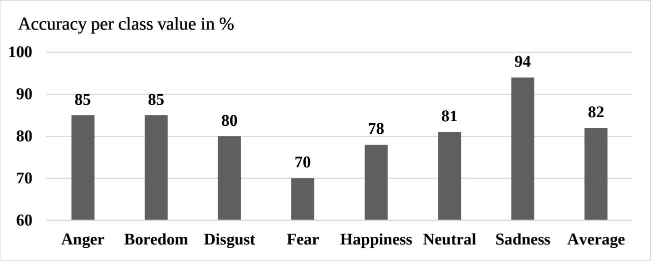


Figure 6: *SVM accuracy per class value for Leave-One-Speaker-Out cross-validation with top ranked 1000 features.*

5 CONCLUSION

The results showed that SVM outperforms the KNN and Naïve Bayes. By using the top ranked 500 features by gain ratio, SVM achieved the highest accuracy of 91%.

In addition, the 10 fold cross-validation of the SVM showed that highest accuracy of 89% was achieved by using the top 750 ranked features. By using the top 300 ranked features the accuracy was 87%. This is the so-called “knee” on the graph, which represents the best tradeoff between the number of features and the achieved performance.

Regarding the accuracy for each of the 7 emotions, experiments were performed with the top ranked 750 features by gain ratio. The best recognized emotion was the “sadness”, with 97%; and the worst recognized emotion was the “happiness” with 68% accuracy.

With LOSO cross-validation, the SVM achieved highest accuracy of 82% by using the top 1000 ranked features. By using the top 500 ranked features the accuracy was 80%. Regarding the accuracy per emotion, experiments were performed with the top ranked 1000 features. The highest accuracy per class (emotion) of 94% was achieved for the class “sadness” and the lowest for the class “fear” 70%.

The results showed that the classifier achieves better accuracy with the 10 fold cross-validation technique compared to the LOSO validation technique. The reason for this is that with the 10 fold cross-validation the training and the testing data usually contain data samples of the same speaker. This is not the case if the system is intended to be used in real life for users not known in advance. However, a hybrid approach that includes a calibration phase at the beginning (for example asking the user to record several data samples) is considered for future work.

References

[1] D. G. Myers. Theories of Emotion. Psychology: Seventh Edition. New York NY: Worth Publishers. 2004.

[2] V. Perez-Rosas, R. Mihalcea. Sentiment Analysis of Online Spoken Reviews. Interspeech, 2013.

[3] A. Halder, A. Konar, R. Mandal, A. Chakraborty. General and Interval Type-2 Fuzzy Face-Space Approach to Emotion Recognition. IEEE Transactions on Systems, Man, and Cybernetics,43 (3), 2013.

[4] R. Horlings, D. Datcu, L. J. M. Rothkrantz. Emotion recognition using brain activity. Proceeding CompSysTech '08 Proceedings of the 9th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing, 2008.

[5] A. Metallinou, S. Lee, S. Narayanan. Audio-Visual Emotion Recognition Using Gaussian Mixture Models for Face and Voice. Multimedia. 2008. ISM 2008. IEEE International Symposium on Multimedia, 2008.

[6] P. Ekman. Emotions in the Human Faces. 1982.

[7] James A. Russell. A circumplex model of affect. 1980.

[8] P. N. Juslin, K. R. Scherer. Vocal expression of affect. In J. A. Harrigan, R. Rosenthal, & K. R. Scherer (Eds.). The new handbook of methods in nonverbal behavior research, pp. 65-135, 2004.

[9] K. R. Scherer. Vocal communication of emotion: A review of research paradigms. Speech Communication 40: 227–256. 2003

[10] M. E. Mena. Emotion Recognition From Speech Signals, 2012.

[11] D. Ververidis, C. Kotropoulos. A review of emotional speech databases. In: PCI 2003. 9th Panhellenic Conference on Informatics., pp. 560–574, 2003.

[12] F. Burkhardt, A. Paeschke, M. Rolfes, W. Sendlmeier, B. Weiss. A Database of German Emotional Speech. 2005. In: Proc. Interspeech. pp. 1517–1520.

[13] F. Eyben, M. Wöllmer, B. Schuller. openSMILE - The Munich Versatile and Fast Open-Source Audio Feature Extractor. 2010.

[14] F. Eyben, F. Weninger, M. Wollmer, Bjorn Schuller. openSmile Documentation. Version 2.0.0., 2013.

[15] H. Deng, G. Runger, E. Tuv. Bias of importance measures for multi-valued attributes and solutions. Proceedings of the 21st International Conference on Artificial Neural Networks (ICANN2011). 2011

[16] I. Kononenko, E. Simec, M. Robnik-Sikonja. Overcoming the myopia of inductive learning algorithms with RELIEFF. Applied Intelligence, Forthcoming.

[17] J. Demšar, B. Zupan. Orange: From experimental machine learning to interactive data mining. White Paper (http://www.ailab.si/orange). Faculty of Computer and Information Science. University of Ljubljana.

[18] D. Aha, D. Kibler. Instance-based learning algorithms. 1991, Machine Learning. 6:37-66.

[19] N. Cristianini, J. Shawe-Taylor. An Introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press, 2000.

[20] R. Stuart, N. Peter. Artificial Intelligence: A Modern Approach. Second Edition, Prentice Hall.

QUALCOMM TRICORDER XPRIZE FINAL ROUND: A REVIEW

Anton Gradišek, Maja Somrak, Mitja Luštrek, Matjaž Gams
Department of Intelligent Systems and Solid State Physics Department
Jožef Stefan Institute
Jamova cesta 39, 1000 Ljubljana, Slovenia
Tel: +386 1 4773967
e-mail: anton.gradisek@ijs.si

ABSTRACT

The Qualcomm Tricorder XPRIZE competition began in January 2012, with the goal of developing a mobile device to monitor health parameters and quickly diagnose several common medical conditions. In August 2014, a list of ten finalists was announced, including a Slovenian team MESI Simplifying diagnostics that brings together companies MESI, D·Labs, and Gigodesign, and partners from academia, Jožef Stefan Institute and Faculties of Electrotechnics and Medicine of the University of Ljubljana. In this review, we present the XPRIZE competition, we briefly look at the ten finalists and more closely at the MESI Simplifying diagnostics approach. Special attention is given to the diagnostic algorithm that was developed in order to facilitate the diagnostic process.

1 INTRODUCTION

XPRIZE, formerly known as the X Prize Foundation, is a non-profit organization that was established in order to stimulate innovation for the benefit of humanity through incentivized competition. The challenges are “audacious, but achievable, tied to objective, measurable goals” [1]. The first prize from the foundation was the Ansari XPRIZE that offered a US$10 million prize for the first non-government organization to launch a reusable manned spacecraft into space twice within two weeks. The prize was won by an aerospace company Scaled Composites with their SpaceShipOne [2]. In the following years, other XPRIZEs were announced, such as Google Lunar XPRIZE that focuses on launching and landing a robotic spacecraft on the Moon, with sending data back to Earth.

The Qualcomm Tricorder XPRIZE [3] was launched in January 2012. The name was inspired by the science-fiction TV series *Star Trek*, where “tricorder” was a device that immediately diagnosed medical conditions of the patients. The sponsor of the prize is Qualcomm, an American semiconductor company that focuses on wireless telecommunications technologies. The aim of the competition is to revolutionize the healthcare system by developing an instrument capable of measuring some key health parameters and diagnosing a set of common medical conditions, independent of professional health care personnel. Being able to diagnose common medical conditions at home benefits both the users by directing them to see the doctor if needed and the healthcare system itself, by reducing the costs and waiting times at medical centers. To be precise, there are two criteria in the competition: i) to continuously monitor key health metrics (blood pressure, respiratory rate, heart rate, temperature and the oxygen saturation – SpO2), and ii) to diagnose a set of 13 pre-selected (core set) health conditions (Anemia, Atrial Fibrillation, Chronic Obstructive Pulmonary Disease (COPD), Diabetes, Hepatitis A, Leukocytosis, Pneumonia, Otitis Media, Sleep Apnea, Stroke, Tuberculosis, Urinary Tract Infection, Absence of condition) and three other conditions from an additional set (Airborne Allergens, Cholesterol Screen, Food-borne Illness, HIV, Hypertension, Hypo- and Hyperthyroidism, Melanoma, Mononucleosis, Osteoporosis, Pertussis, Shingles, Strep Throat). Furthermore, the consumer experience represented an important component of the qualifying round evaluation criteria.

Around 300 teams from all over the world entered the competition, with 34 teams reaching the qualifying round. In August 2014, ten teams were chosen for the final round of the competition that will include testing the products on real patients during the summer of 2015. The winners of the competition will be announced in January 2016.

In this review paper, we present the ten finalists and their approaches, based on the information made public so far. Some teams unveiled several details about their products while others are more secretive. We pay special attention to the MESI Simplifying diagnostics team approach and the diagnostic algorithm.

2 FINALIST TEAMS

Among ten finalists, there are four teams from the United States, two from the United Kingdom, and one from each Taiwan, Canada, India, and Slovenia. The teams are presented as on the XPRIZE website, except for MESI Simplifying diagnostics that is presented separately later on.

Aezon [4] is a team of student engineers from Johns Hopkins University in Baltimore, Maryland (US), with several partners from the industry. Their solution consists of four components, each being developed by or in partnership with a specialized company. The vital signs monitoring unit is designed to wrap around the neck, like a neck support pillow, and is being designed by a startup company Aegle. The diagnostic module is exploiting microfluidic chip technology and qPCR to test for the presence of pathogens and is being developed in partnership with Biomeme. The data is processed by a smartphone app that also uses algorithms to direct users towards relevant tests. In addition, the phone uses software for spirometry, developed by SpiroSmart. The data is stored on a cloud where an API uses big data to help turn user reported symptoms into diagnostic solutions. The team also participated in an Indiegogo campaign where they raised around 5000 US$.

CloudDX [5] is a Canadian team, associated with the company Biosign Technologies, a manufacturer of medical devices. The vital signs unit is placed around the neck; it uses two electrodes at upper chest area to monitor ECG and an ear bud with an infrared temperature sensor to measure body temperature. An ear clip uses photoplethysmograph to monitor breathing and heart rate. Blood pressure is measured by a wrist monitor with the pulse transit time approach. The diagnostic module is designed to analyze saliva, blood, and urine. The team is working with industrial partners to consolidate multiple tests onto one multi-strip cassette. In addition, an application was developed to accept data from fitness devices and to integrate them into the system.

Danvantri [6] is a team from Chennai, India, associated with American Megatrends. The main component of their product is a handheld health monitor that features a 3-Lead ECG electrode to measure ECG signals from the finger, pulse oximeter, an infrared temperature sensor, camera, 3-axis accelerometer for monitoring physical activity and a glucometer strip attachment node. Additional devices include a wireless spirometer, neckband ECG/EEG meter, otoscope, and urine sample analyzer. The data is processed and visualized either on a smartphone or on a tablet.

DMI [7] is a team from Boston-Cambridge, Massachusetts (US), connected to the DNA Medicine Institute. They developed the rHEALTH Sensor which is a device that employs fluorescence detection optics, microfluidics, and nanostrip reagents to perform a suite of hematology, chemistry, and biomarker assays from blood. The device was developed in collaboration with NASA to monitor astronaut health.

Dynamical Biomarkers Group [8] is a team from Taiwan. Their system consists of five components: Smart Vital-Sense-Patch and Smart Vital-Sense-Wrist module; Smart Blood Sense module; Smart Scope module; Smart Exhaler module; and Smart Urine Sense module. The modules are connected to a smartphone app that runs algorithms based on proprietary algorithms to conduct a diagnosis.

Final Frontier Medical Devices [9] is a team from Paoli, Pennsylvania (US), connected to Basil Leaf Technologies. They are developing a device called DxtER, which relies on algorithms developed by medical experience as well as on actual patient charts. Concept art for the product indicates the device is roughly spherically shaped with integrated sensors.

Scanadu [10] is a team from Moffett Field, California (US). The team's product is called Scanadu Scout, which is a disk-shaped device that contains sensors for temperature, hearth rate, and blood pressure. The disk is to be held between the thumb and index finger and placed on the forehead. The data is transferred to a smartphone and processed there. No technical specifications are known yet, neither is the approach for the diagnostic module. Scanadu ran an Indiegogo campaign from May to July 2013 and managed to raise over US$ 1.6 million. The campaign has also received considerable media coverage.

SCANurse [11] is a team from London, UK. Their system consists of blood, vitals, breath, and image units. No specific information was provided on their website at the time of writing.

Zensor [12] is a team from Belfast, UK, connected to Intelesens Responsive Healthcare, a company working on non-invasive vital signs monitoring. Their prototype can detect 3-lead ECG, respiration rate, temperature, and motion. SpO2 sensor is being developed. To diagnose medical conditions, urine and blood analysis is included, although the details have not been made public yet.

3 MESI SIMPLIFYING DIAGNOSTICS

MESI Simplifying diagnostics [13] is a team from Ljubljana, Slovenia. The team consists of partners from the industry and the academia. The team is led by MESI, a start-up company that specializes in development of medical devices. Their flagship product is an ankle-brachial index measuring device (ABPI MD) for the detection of peripheral arterial disease. Company D·Labs is responsible for a mobile app and API while Gigodesign focuses on improving the user experience and industrial design. Partners from academia come from Jožef Stefan Institute (Department of Intelligent Systems) and two faculties of University of Ljubljana, Faculty of Electrotechnics and Faculty of Medicine. Academic partners are responsible for the development of algorithms and for expert medical knowledge. The system consists of several modules [14]. A bracelet monitors activity and three vital signs, ECG, SpO2, and temperature. A "shield" module is placed on the upper arm and consists of a wireless cuff for blood pressure measurements. It also contains a patch located on the chest to measure SpO2, temperature, ECG, respiratory rate, and activity tracking. Data obtained from the bracelet and the shield module fulfill the vital signs monitoring requirement of the competition.

The diagnosis of medical conditions is performed with the help of a smartphone app and aims to recognize all conditions from the core set, together with Hypertension, Melanoma, and Strep Throat. The user, that has already performed the vital signs measurements, indicates his concern: "I feel pain" or "I feel unwell". If "pain" is chosen, the user specifies the type of the pain on a schematic human figure (such as "chest pain"). Based on vital signs data and the type of pain/feeling unwell, the algorithm generates a list of possible symptoms that the user may experience. This list is generated to include both the symptoms that the user most probably experiences at the time and would probably want to report, and also the most relevant symptoms that would help the physician or the diagnostic method set a reliable diagnosis. Based on the chosen symptoms, the algorithm then asks for a couple of additional symptoms in order to narrow down the diagnosis and direct the user to one or more specialized modules that confirm or reject the suggested diagnosis. There are four specialized modules. A module "To see" includes a camera which is used to diagnose Melanoma and Strep Throat. Using a special camera is advantageous to using the integrated camera in a smartphone since the specifications of phone cameras may vary from a model to a model. In addition, light conditions are easier to control with a dedicated module. A module "To hear" includes microphones that are used to monitor breathing – in order to detect pulmonary diseases. This module also allows user to perform a spirometry which is used to diagnose COPD. The urine module, "Pee", performs urine analysis using test strips and a camera that reads the test results. The fourth module is called "Blood" and is intended for blood tests. In order to achieve best user experience, this module should rely on non-invasive methods, such as spectroscopy, although it is more likely that a drop of blood will be required for analysis in the final version. This module is intended for detection of diabetes and anemia.

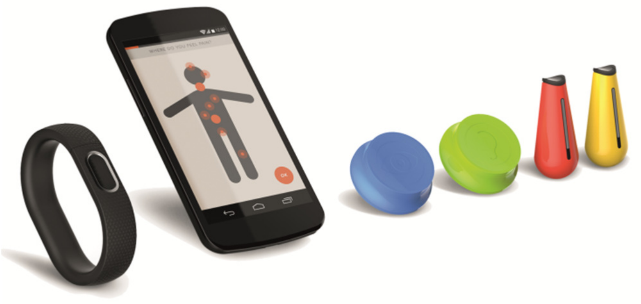


Figure 1: *MESI Simplifying diagnostics system: a bracelet, smartphone app, and four diagnostic modules – To see, To hear, Blood, and Pee. The Shield module is not shown here, it comes in form of a sleeve with attachable electrodes.*

4 DIAGNOSTIC ALGORITHM

The diagnostic algorithm was developed at the Department of Intelligent Systems of Jožef Stefan Institute by Maja Somrak, Mitja Luštrek, Matjaž Gams, and the author of this review [15]. The aim of the algorithm is to predict the medical condition of the patient, based on the symptoms that he or she experiences. Around 60 different symptoms are taken into account. The problem is highly non-trivial. There is no simple function that would map the domain of a group of symptoms to a codomain containing a single disease. People with the same medical condition may experience different symptoms, for example, people with Otitis Media may or may not experience a headache or a discharge from the ear. An individual symptom is usually typical for several different diseases. For example, elevated temperature is typically exhibited in cases of Tuberculosis, Pneumonia, Strep Throat, Otitis Media, and others. On the other hand, even healthy people ("absence of conditions") often experience some symptoms due to reasons that are not connected to diseases. Fatigue may be related to a lack of sleep while high blood pressure may be a consequence of drinking caffeinated drinks. In addition, asking the patient for all symptoms on the list is not considered user-friendly, therefore the goal is to diagnose the medical condition as accurately as possible using as small number of questions as possible. In order to achieve the best performance, the algorithm combines expert medical knowledge and methods of artificial intelligence. At this point, we only aim to diagnose the diseases of patients with a single medical condition. Diagnosing a combination of more than one disease for a single patient is a next-level problem.

As discussed above, the initial input for the algorithm comes from the vital signs measurements (symptoms such as elevated temperature or high blood pressure) and from the pain symptoms that the user chooses. Additionally, for personalized tests, the algorithm may also include identified risk factors for a particular user (from the algorithm point of view, we also treat the risk factors as "symptoms"). For example, smokers and older people are more likely to develop COPD than non-smokers, people with a high BMI have higher risks for diabetes, etc. All these are called the "initial symptoms". The additional list of suggested symptoms is generated using association rules (ARs) and the minimum-Redundancy-Maximum-Relevance (mRMR) method. The ARs (symptom A –> symptom B) are used to produce a set of probable additional symptoms. The goal of the mRMR method is to select symptoms that are as mutually dissimilar as possible and at the same time as indicative of the medical condition as possible. In other words, the algorithm tries to avoid asking the user about several similar symptoms and at the same time ask about symptoms that cover all spectrum of probable medical conditions.

In the following step, the information gathered up to this point is used for actual disease prediction. The probabilities for the 15 medical conditions are evaluated using a set of J48 classifiers, one for each of the conditions. There are two probability thresholds: conditions above the high threshold are considered very probable and conditions below the low threshold are considered unlikely. The area between the two thresholds is a so-called "gray zone" where we do not have enough information to make a reliable claim whether the medical condition is present or not. The diagnostic procedure terminates when all conditions from the list are either above the high or below the low threshold. If one or more condition remain in the gray zone, at least one additional question (symptom) is required for a confident prediction. The additional symptom is chosen according to the highest information gain (IG) that an individual symptom would bring.

Calculation of the IG and mRMR values, searching for ARs, and building the J48 classifiers is based on two types of data – real and simulated patient data. Real patient data was collected either with both patients with medical conditions and healthy individuals filling in a questionnaire about the symptoms they experience (a complete set of symptoms), or by medical doctors retroactively filling in the symptom tables for real patients. The simulated dataset was build using expert medical knowledge. Physicians prepared a table of probabilities for patients suffering from each of the medical conditions to exhibit each of the symptoms from the list, based on their professional experiences. Using this table, it is possible to generate millions of distinct "virtual patients". Initial tests using only simulated data showed high sensitivity and specificity for disease diagnostics [15]. Tests using a combination of real and simulated data are currently underway.

5 CONCLUSIONS

We present an overview of the Qualcomm Tricorder XPRIZE competition and the teams that reached the final round, with a special focus on the Slovenian team entry. The approaches of many teams are similar to some degree. The most common approach is to use of single a device with a number of integrated sensors to monitor vital signs (the first competition task). The second task, the diagnosis of medical conditions, is typically achieved using a series of dedicated additional modules. Some teams rely strongly on detection of biomarkers in body fluids while others also incorporate technologies such as spirometry and image-processing algorithms. Several teams mention they use algorithms for diagnostics, although not much has been revealed to the public so far.

The MESI Simplifying diagnostics approach consists of a bracelet and a "Shield" module to monitor vital signs. The diagnosis of medical conditions is obtained using an algorithm that runs on a mobile device. The algorithm uses the vital signs data and the symptoms entered by the patient to predict a possible medical condition and to direct the patient to use a specialized module which confirms or rejects the prediction.

An overview of the algorithm, developed at Jožef Stefan Institute, is presented. The algorithm combines expert medical knowledge with methods of artificial intelligence and machine-learning. The aim of the algorithm is to make an accurate prediction of diagnosis with a small number of questions, to improve the user experience. We outline the challenges of the task. Testing of the algorithm on real patient data is currently underway and the results will be published later.

Reference

[1] http://www.xprize.org/

[2] http://space.xprize.org/ansari-x-prize

[3] http://www.qualcommtricorderxprize.org/

[4] http://www.aezonhealth.com/

[5] http://www.clouddx.com/

[6] http://www.vitalsplus.com/

[7] http://www.dnamedinstitute.com/

[8] http://dbg.ncu.edu.tw/

[9] http://www.basilleaftech.com/

[10] https://www.scanadu.com/

[11] http://www.scanurse.com

[12] http://www.intelesens.com

[13] http://www.simplifyingdiagnostics.com/

[14] M. Somrak, M. Luštrek, J. Šušterič, T. Krivc, A. Mlinar, T. Travnik, L. Stepan, M. Mavsar, M. Gams: Tricorder: Consumer Medical Device for Discovering Common Medical Conditions, Informatica 38 (2014) 81–88.

[15] M. Somrak, A. Gradišek, M. Luštrek, A. Mlinar, M. Sok, M. Gams: Medical diagnostics based on combination of sensor and user-provided data. AI-AM/NetMed 2014, Artificial Intelligence and Assistive Medicine: proceedings of the 3rd International Workshop on Artificial Intelligence and Assistive Medicine co-located with the 21st European Conference on Artificial Intelligence (ECAI 2014), Prague, Czech Republic, pp. 36-40

AVTOMATIZACIJA IZGRADNJE BAZE ODGOVOROV VIRTUALNEGA ASISTENTA ZA OBČINE

Leon Noe Jovan, Svetlana Nikić, Damjan Kužnar, Matjaž Gams

Odsek za inteligentne sisteme, Institut "Jožef Stefan", Jamova cesta 39, 1000 Ljubljana
e-mail: leon.jovan@gmail.com

POVZETEK

Asistent je inteligentni virtualni pomočnik, ki odgovarja na vprašanja, postavljena v naravnem jeziku in je sposoben poiskati odgovore na spletnih straneh. Cilj projekta Asistent je vzpostavitev spletne storitve za izdelavo in urejanje prilagojenega virtualnega asistenta, ki si ga bodo lahko občine namestile na svoje spletne strani in tako obiskovalcem olajšale iskanje informacij, ki jih stran ponuja. Ta prispevek opisuje postopke avtomatizacije izgradnje baze asistentovih odgovorov z uporabo različnih pristopov strojnega učenja ter ekstrakcije informacij iz spletnih strani. Ta postopek bo olajšal delo občin pri uvajanju asistenta, kar predvsem vpliva na razširjenost uporabe na spletnih straneh občin. Opisana je osnovna ideja arhiterkture sistema za avtomatizacijo gradnje odgovorov virtualnega asistenta, predvsem pa so predstavljeni pristopi za generiranje odgovorov in ekstrakcijo podatkov iz spletnih strani.

1 UVOD

Večina slovenskih spletnih strani je omejena z uporabo starejših spletnih tehnologij, kar povzroča oteženo iskanje po njih. Splošni iskalniki v povprečju najdejo le med 10% do 30% ustreznih odgovorov [13]. Ena izmed možnih rešitev je inteligentni virtualni pomočnik oz. asistent, ki zna odgovoriti na vprašanja v naravnem jeziku. Asistenti se pojavljajo kot pomoč pri iskanju po spletnih straneh, pametnih telefonih, itd. Ena najbolj poznanih asistentk na svetu je npr. Siri, ki jo najdemo na novejših sistemih iOS podjetja Apple Inc. in se jo lahko uporablja v več svetovnih jezikih. Prva virtualna asistentka v slovenščini pa je Vida, ki je nastala kot pomoč pri iskanju po straneh DURSa.

Cilj celotnega projekta [12] je ustvariti virtualne asistente za slovenske občine, ki bi bili potem lahko dostopni preko njihovih spletnih strani.

Vsaka baza znanja za neko občino je sestavljena iz vnosov, vsak vnos pa vsebuje vprašanje, imenovano pravilo, in odgovor, pri čemer so pravila ključne besede vprašanj, ki jih zastavljajo uporabniki. Vsak asistent občine ima približno 500 pravil, ki so enaka za vse občine, odgovore pa je potrebno kreirati za vsako posamezno občino. Tem pravilom pravimo "zlata osnova". Zlata osnova je bila oblikovana iz pravil in odgovorov, ki so jih določene občine ročno vnesle v svoje asistente na začetku projekta. Ročno vnašanje odgovorov na vprašanja je zelo zamudno, kar lahko vpliva na to, koliko občin bo sodelovalo pri projektu. Zato se pojavlja potreba po avtomatizirani rešitvi, ki bi ustvarila odgovore za posamezno občino za celotno zlato osnovo.

Osnovna ideja naše rešitve je, da poskušamo čimbolj avtomatizirati vnašanje podatkov v bazo asistenta. S klasifikacijo želimo določiti, na kateri strani spletni strani občine se nahaja podatek, ki ga zahteva posamezno pravilo. S kratkimi skriptami pa želimo nato iz spletne strani pridobiti podatek ter ga prikazati v kratkem, uporabniku prijaznem odgovoru.

Na koncu projekta bi tako vsaka občina lahko imela sebi prilagojenega asistenta, s čimer bi vsi občani dobili možnost naravnega poizvedovanja in komuniciranja z občinami.

Problem je soroden ekstrakciji informacij (angl. Information extraction) iz HTML dokumentov [4]. Svetovni splet je zbirka velike količine dokumentov, vendar pa podatki niso najbolje strukturirani. Naša naloga je, da iz takšnih nestrukturiranih podatkov najdemo podatke, ki so za nas uporabni.

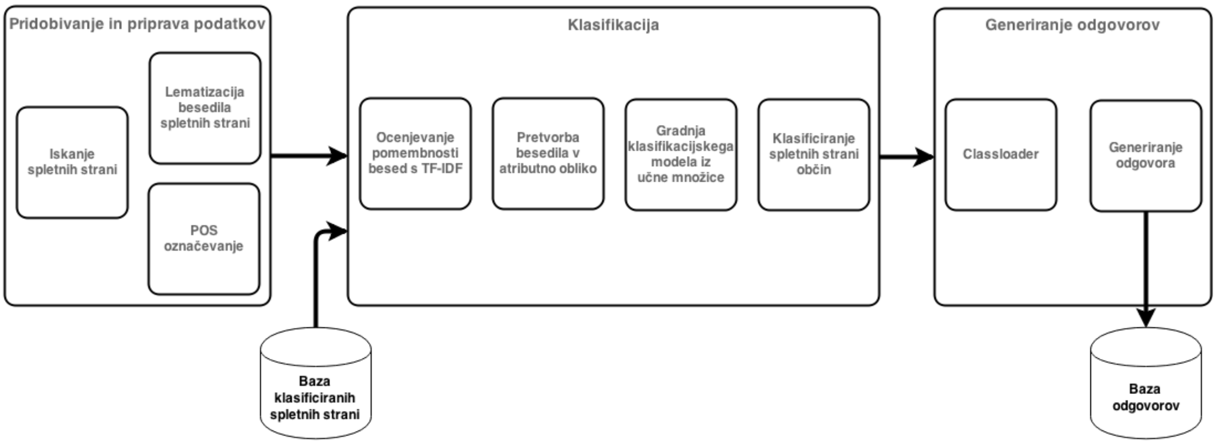
V zadnjem času je bilo predlaganih več različnih pristopov za ekstraktcijo informacij iz spleta. Pristopi vključujejo uporabo strojnega učenja, iskanja vzorcev in podobno z različnimi stopnjami avtomatizacije [5].

2 PROJEKT ASISTENT

Celoten proces izgradnje baze odgovorov je sestavljen iz treh korakov, in sicer pridobivanje in priprava podatkov, klasifikacija in generiranje odgovorov.

Najprej moramo podatke pridobiti in jih pripraviti za nadaljno obdelavo. Podatke pridobimo s spletnim pajkom, ki obišče vse spletne strani občine in jih v obliki HTML dokumenta shrani v interno bazo podatkov. Iz teh datotek nato izluščimo celotno besedilo, ga lematiziramo in označimo besede z označevalnikom, saj bomo te podatke uporabili v naslednjih korakih. Ta dva postopka naredimo z lematizatorjem LemmaGen [9] in Oblikoslovni označevalnik za slovenski jezik [7]. Spletnega pajka za shranjevanje spletnih strani smo izdelali z uporabo Java knjižnjice Jsoup [6].

Naslednji korak je klasifikacija, kjer moramo pridobljene spletne strani razvrstiti med skoraj 500 vnosov. Za klasifikacijo uporabimo lematizirana besedila spletnih strani, ki smo jih predstavili kot vrečo besed [2], z mero TF-IDF [3] pa izberemo le najbolj pomembne besede, saj je preveč različnih besed, da bi obravnavali vse.



Pridobivanje in priprava podatkov
Iskanje spletnih strani
Lematizacija besedila spletnih strani
POS označevanje
Klasifikacija
Ocenjevanje pomembnosti besed s TF-IDF
Pretvorba besedila v atributno obliko
Gradnja klasifikacijskega modela iz učne množice
Klasificiranje spletnih strani občin
Generiranje odgovorov
Classloader
Generiranje odgovora
Baza klasificiranih spletnih strani
Baza odgovorov

Figure 1: Arhitektura sistema

Za generiranje odgovora za določen vnos imamo torej na voljo spletno stran, ki smo jo uvrstili, da vsebuje podatke primerne temu pravilu, poleg pa tudi lematizirano in označeno besedilo te strani, včasih pa tudi katere od drugih virov (Wikipedia, Uradni list, ...). Iz teh podatkov torej generiramo kratek odgovor, ki ga prikaže virtualni asistent.

3 AVTOMATIZACIJA IZGRADNJE BAZE ODGOVOROV

Izdelali smo program, ki je zmožen zgraditi bazo odgovorov 471 vprašanj, ki so skupna vsem občinam ("zlata osnova"). Sistem za generiranje odgovorov je sestavljen iz 471 razredov, izdelanih v programskem jeziku Java, ki so poimenovani po ključu $Rule < IDpravila >$ glede na pravilo, za katerega generirajo odgovor. Razred, ki generira odgovor, potrebuje le spletno stran, iz katere črpa informacije, v nekaterih primerih pa tudi lematizirano in označeno besedilo te spletne strani. Osnoven razred je RuleClass, ki omogoča branje vseh potrebnih podatkov ter vračanje odgovora. Izdelali smo hierarhijo razredov, ki dedujejo in so nadgradnja RuleClass razreda, omogočajo pa generiranje odgovorov, ki so bolj specifični in rešujejo določen problem, ki se pogosto pojavlja. Takšni problemi so pridobivanje obrazcev, kontaktov, imen oseb, povzetkov iz daljšega besedila in povezav. Pristopi za reševanje takšnih problemov so opisani v nadaljevanju.

3.1 Vloge in obrazci

Za pridobitev obrazcev, ki so specifični za vsako občino, smo uporabili spletno stran Uradni list [11], ki objavlja zakone, predpise in druge javne objave v Republiki Sloveniji. S pomočjo iskalnika na spletni strani računalnik pridobi določen obrazec, ki ga potem posreduje uporabniku v obliki odgovora.

Do dokumentov na spletni strani Uradni list smo dostopali tako, da smo naredili zahtevek z ustreznimi GET parametri, ki od strani zahtevajo, da nam poišče določene dokumente.

```
http://www.uradni-list.si/1/search?smode=ul&
    cmd=search&q=iskalni_niz&rubm=s&
    selectItem=id_obcine&rublist%5B%5D=
    id_obcine
```

Iskalni niz smo določili glede na pravilo, ID številko občine pa smo prebrali iz spletne strani tako, da smo poiskali ustrezen HTML element, ki se je nahajal med možnostmi za filtriranje in je vseboval ime občine ter ID za filtriranje po tej občini. Za pravilno delovanje smo vse šumnike pretvorili s Percent-encoding [8], ki se uporablja za kodiranje nestandardnih znakov v URL naslovih. Rezultate iskanja smo nato prebrali iz vrnjene spletne strani. Če je vrnjen rezultat samo eden, smo povezavo do tega elementa dodali v odgovor, če pa je rezultatov iskanja več, se v odgovor dodajo prvi trije dokumenti s pripisom, da se več dokumentov nahaja na strani v ozadju.

3.2 Povzetki

Vprašanja so v nekaterih primerih precej splošna in običajno zahtevajo daljši odgovor, ki ni primeren za asistentov odgovor. Zato asistent odgovori s krajšim povzetkom, več o tem pa si uporabnik lahko prebere na strani, ki se mu prikaže v ozadju. Takšna vprašanja so na primer opisi kulturnih znamenitosti, kmetijstva, predstavitev grba ter zastave in podobno.

Algoritem deluje na principu ključnih besed in regularnih izrazov, ki jih predhodno podamo za posamezno pravilo. Podamo lahko seznam ključnih besed, ki jih algoritem pretvori v regularne izraze. Potem iz celotnega HTML dokumenta rekurzivno odstranimo vse vrstične elemente, njihovo vsebino pa dodamo staršu tega elementa. Ta postopek nam omogoča lažje določanje besedilnih enot, saj nam vmesni vrstični elementi ne delijo enega odstavka na več delov. Po končanem postopku odstranjevanja vrstičnih elementov, algoritem pregleda vsa vozlišča z besedilom (TextNode) in prešteje, kolikokrat se elementi iz seznama regularnih izrazov pojavljajo v besedilu. Na podlagi te ocene algoritem izbere najboljši odstavek in ga prikaže kot odgovor.

3.3 Kontakti

Nekateri kontakti so na voljo samo na straneh občine in jih je potrebno pridobiti neodvisno od oblike strani. Sem spadajo na primer kontakt direktorja občine, svetovalcev in podobno.

Kontakt je v osnovi sestavljen iz imena, naziva, telefonske številke ter elektronskega poštnega naslova oziroma kombinacije le-teh. Najprej je bilo tako potrebno iz besedila prepoznati te elemente.

Za prepoznavanje imen smo naredili bazo imen, pri čemer smo za osnovo vzeli podatke z Wikipedie [1]. V besedilu smo nato lahko imena preprosto našli s pregledom baze.

Za iskanje telefonskih številk smo si pomagali s knjižnico Libphonenumber [10], ki najde v podanem besedilu vse telefonske številke neke države.

Elektronske naslove smo iskali s pomočjo regularnega izraza:

```
[_A-Za-z0-9-\\+]+(\\.[_A-Za-z0-9-]+)*@[A-Za-
    z0-9-]+(\\.[A-Za-z0-9]+)*(\\.[A-Za-z
    ]{2,})
```

Če si spletno stran predstavljamo kot drevesno strukturo HTML elementov, je ta algoritem iskal v tem drevesu najnižje ležeče vozlišče, ki vsebuje vse te elemente - ime, naziv (podane ključne besede), telefonsko številko in/ali poštni naslov. Da bi bil odgovor v uporabniku prijazni obliki, smo odstranili še razne stile in vrnili HTML iz vozlišča kot odgovor.

3.4 Osebe

Nekatera pravila zahtevajo prepoznavanje oseb v besedilu in njihove vloge. Takšno pravilo je na primer ugotavljanje župana ali podžupanov občine. Za ta postopek smo uporabili Oblikoslovni označevalnik za slovenski jezik [7], ki nam določi sklon, spol, število, besedno vrsto ter ugotovi ali gre za obče ali lastno ime.

Župana na spletni strani smo prepoznali tako, da iščemo vsa zaporedja besed, dolga vsaj dve besedi, ki so sestavljena samo iz samostalnikov v ednini in so lastna imena. Ta zaporedja besed se lahko začnejo tudi z besedami mag., dr., ali pa župan v našem primeru. Vsa ta zaporedja besed še preverimo, če vsebujejo eno od osebnih imen, ki smo jih pridobili na wikipediji. Tako predpostavimo, da gre res za ime osebe. Na koncu izberemo osebo, ki se največkrat pojavlja v besedilu na spletni strani.

4 OBJEKTIVNA EVALVACIJA REZULTATOV

Za evalvacijo rezultatov smo kot učno množico uporabili podatke 16 občin, kjer so bile spletne strani ročno klasificirane. Klasifikacije za ostale občine nismo testirali, saj še ni popolnoma narejena, a to ne vpliva na testiranje generatorjev, na kar smo se mi osredotočili, saj smo najprej želeli preveriti kako dobro deluje generiranje odgovorov.

Za učni primer smo vzeli občino Pivka in njene spletne strani ter vnose, saj ima najbolj popolne odgovore. Kvaliteto ustvarjenih odgovorov smo preizkusili na podlagi 16 drugih občin. Ročno smo pregledali vse vnose in strani, katere so vpisale občine, in izločili tiste, ki niso bile primerne. Tako smo izločili vpliv napačne klasifikacije na kvaliteto ustvarjenih odgovorov. S tem smo pri nekaterih občinah močno zmanjšali število pravil, saj je bilo veliko podanih spletnih strani oziroma odgovorov napačnih ali pa so bili prepisani od neke druge občine. Rezultat generiranja odgovorov smo pregledali ročno, jih primerjali z ročno vpisanimi odgovori, ki so jih izdelale občine, ter izračunali delež ustreznih odgovorov. Odgovor je bil ocenjen kot ustrezen, če je vseboval iskane podatke v uporabniku prijazni obliki. Sem torej niso šteti odgovori, ki pozovejo uporabnika, naj si ogleda stran v ozadju.

Rezultate ocenjevanja ustreznosti ustvarjenih odgovorov prikazuje spodnja tabela 4. Prvi stolpec predstavlja občino, za katero smo preverjali rezultate, drugi število vseh pravil, ki jih je občina vpisala in tretji število pravil, ki imajo pravilne ali vsaj delno pravilne odgovore. Zadnja dva stolpca predstavljata rezultate generiranja odgovorov.

Občina	Vnešena pravila	Ustrezna pravila	Ustrezni odgovori	Neustrezni odgovori
Koper	343	311 (91%)	235 (76%)	76 (24%)
Velenje	388	347 (89%)	269 (78%)	78 (22%)
Hrastnik	399	363 (91%)	268 (74%)	95 (26%)
Hrpelje - Kozina	457	395 (86%)	314 (79%)	81 (21%)
Kidričevo	209	187 (89%)	165 (88%)	22 (12%)
Krško	456	375 (82%)	297 (79%)	78 (21%)
Log - Dragomer	294	244 (83%)	206 (84%)	38 (16%)
Mengeš	338	270 (80%)	230 (85%)	40 (15%)
Podvelka	202	193 (96%)	170 (88%)	23 (12%)
Vodice	132	128 (97%)	113 (88%)	15 (12%)
Vransko	257	235 (91%)	202 (86%)	33 (14%)
Celje	452	431 (95%)	350 (81%)	81 (19%)
Litija	456	264 (58%)	228 (86%)	36 (14%)
Šentjur	373	330 (88%)	247 (75%)	83 (25%)
Žalec	448	425 (95%)	346 (81%)	79 (19%)
Trbovlje	286	258 (90%)	212 (82%)	46 (18%)
Povprečje	343	88%	82%	18%

Vsi rezultati so bili narejeni na podlagi ročne klasifikacije. Nadaljnje delo tako vsebuje še klasifikacijo spletnih strani ostalih občin. Ker klasifikacija ne bo delovala popolno, lahko pričakujemo nekoliko slabše rezultate.

Ustreznih generiranih odgovorov je 82% odstotkov, kar ocenjujemo za zadovoljiv rezultat, ki olajša človekovo delo. Višji odstotek so vrnili odgovori tistih občin, ki so imele manj pravilno podanih povezav, saj so večinoma ostale splošne povezave, ki so za vse občine podobne in jih je lažje generirati.

5 Zaključek

V tem dokumentu smo opisali postopke avtomatizacije izgradnje baze asistentovih odgovorov z uporabo različnih pristopov strojnega učenja ter ekstrakcije informacij iz spletnih strani. Opisana je osnovna ideja arhiterkture sistema za avtomatizacijo gradnje odgovorov virtualnega asistenta, predvsem pa so predstavljeni pristopi za generiranje odgovorov in ekstrakcijo podatkov iz spletnih strani.

Preizkusili smo del sistema, ki iz spletnih strani pridobiva informacije in generira kratke odgovore na določena vprašanja. Kvaliteta ustvarjenih odgovorov je zadovoljiva, 82% odstotkov odgovorov je bilo ustreznih, kar ocenjujejmo, da je dovolj, da olajša človeško delo.

Za nadaljnje delo načrtujemo izdelavo klasifikacijskega modela, ki bo za vsako pravilo poiskal spletno stran občine, na kateri se nahaja ustrezen podatek. Od klasifikacije je zelo odvisen tudi generator odgovorov, ki smo ga izdelali, saj lahko deluje precej slabše ob slabši klasifikaciji.

References

[1] Seznam osebnih imen. URL http://sl.wikipedia.org/wiki/Seznam_osebnih_imen. Pridobljeno 1. 9. 2014.

[2] Bag-of-words representation of text. URL https://inst.eecs.berkeley.edu/~ee127a/book/login/exa_bag_of_words_rep.html Pridobljeno 5. 9. 2014.

[3] Tf-idf weighting. URL http://nlp.stanford.edu/IR-book/html/htmledition/tf-idf-weighting-1.html Pridobljeno 5. 9. 2014.

[4] InformationExtraction URL http://en.wikipedia.org/wiki/Information_extraction Pridobljeno 3. 9. 2014.

[5] C. Chang, M. Kayed, M. R. Girgis in K. F. Shaalan. A Survey of Web Information Extraction Systems. *Knowledge and Data Engineering*, let. 18 št. 10, str. 1411-1428, 2006.

[6] Jsoup, Java HTML Parser. URL http://jsoup.org/. Pridobljeno 16. 9. 2014.

[7] Oblikoslovni označevalnik za slovenski jezik. URL http://www.w3schools.com/tags/ref_urlencode.asp. Pridobljeno 16. 9. 2014.

[8] HTML URL Encoding Reference URL http://oznacevalnik.slovenscina.eu/Vsebine/Sl/ProgramskaOprema/Navodila.aspx. Pridobljeno 16. 9. 2014.

[9] LemmaGen, Multilangual Open Source Lemmatisation. URL http://lemmatise.ijs.si/. Pridobljeno 16. 9. 2014.

[10] libphonenumber, Google's phone number handling library URL https://code.google.com/p/libphonenumber/. Pridobljeno 16. 9. 2014.

[11] Uradni list Republike Slovenije URL http://www.uradni-list.si/. Pridobljeno 18. 9. 2014.

[12] Projekt Asistent, Virtualni asistent za občine in društva. URL http://www.projekt-asistent.si/wp/. Pridobljeno 18. 9. 2014.

[13] Projekt Asistent, Virtualni asistent za občine in društva, opis projekta. URL http://www.projekt-asistent.si/wp/?page_id=100. Pridobljeno 18. 9. 2014.

INFERRING MODELS FOR SUBSYSTEMS BASED ON REAL WORLD TRACES

Rutger Kerkhoff, Aleš Tavčar[1]*,Boštjan Kaluža*[1]
Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenija[1]
e-mail: rutger.kerkhoff@gmail.com, {ales.tavcar, bostjan.kaluza}@ijs.si

ABSTRACT

Creating simulations for smart cities is a complex and time consuming task. In this paper we show that using traditional Bayesian networks and real world data traces it is possible to infer models that can simulate the original domain. The created model can provide great insight into the actual subsystems that are considered. We show that given a set of observed values we can successfully use the created model to simulate data and show trends present in the original system.

1 INTRODUCTION

With modern cities becoming more complex and ever increasing in size it is of vital importance to control and optimize the different systems present in a city. While the different systems already available in a city, e.g. the power grid, waste management, bus scheduling etc., can and are being optimized, the bigger picture has not been explored yet. Connecting them allows further optimisation and realises emergent behaviour; something that cannot be observed when looking at a subsystem alone.

An important aspect of realising such a system is understanding the relations between the subsystems and even within the subsystems themselves. A concrete example of where this knowledge is needed is simulation. It is of vital importance to thoroughly test such a city-controlling system before applying it, and for that simulations are required. A simulation will also allow faster development of new systems and applications within the smart city. However, simulating a smart city is an immensely complex task, and in all but the simplest cases it is infeasible to specify all the variables and relations concerned in such a simulation. We therefore propose to learn a model, representing all the variables and relations in the smart city, from real-world data traces.

To allow for a better understanding of the system and the ability to simulate new traces we use a graphical model, a Bayesian belief network [1] to be exact. In this article we will focus on different ways of creating the network and inferring the probabilities. The ability to graphically represent the network makes it an ideal tool for understanding the modelled system. Moreover, we can input a set of observed variables and update, according to those values, the probabilities of the unobserved which makes it a suitable tool for creating simulations.

As a proof of concept this article will focus on simulating a single smart house. While this model will be significantly less complex, and even allow for exact simulation, it is ideal for showing the power of this approach. In this article we will show that using a manually defined Bayesian network already allows for quite accurate predictions. Expanding to an automatically learned network improves these even more, showing great promise for modelling a complete smart city.

2 RELATED WORK

Using a Bayesian network to model a system is not a new idea, take for example a water supply network [5], where the authors try to predict when a pipe will burst. It has also been used to model mobility within a city [3] or to predict when to replace parts of the New York power grid [2]. While Bayesian networks have not yet been used to model a complete city, there are approaches which try to tackle large or distributed models. For example a hierarchical object oriented approach [7] or building local networks and identifying which variables likely link the systems [6]. While not needed for the model of the smart house discussed in this paper these techniques will likely be necessary when considering a smart city. As an example of what possible data streams and variables can be found in a smart city one can look at sensor data streams in London [8].

While other papers focus mainly on classification or decision support systems [4], we focus on recreating the original data and trends for a smart environment. We are not aware of any complete simulation and creation of a trace using a Bayesian network.

3 DOMAIN

The domain explored in this paper is that of a smart house, which is less complex than a city, but it is well suited to show the ideas and explore the methodology. The traces used to model, learn and simulate the smart house are obtained from the EnergyPlus simulator [14]. The simulator was developed within the OpUS system [13] and it is based on a real world building. While simulating a simulator seems pointless, it allows for testing of the model in varying situations without having to gather real world data for an extended amount of time. For simulating the city this will be necessary.

The simulation has 6 input variables, all concerning the environment. One should think of values for outside temperature, wind etc. The simulator also supports setting if the occupant is present, the house was set to be empty from 7:00 to 17:00 every day. Redundant output variables were filtered out and we are left with 9 variables, ranging from inside temperature, heating coil power consumption to electricity produced by a solar panel. All variables are recorded in an interval of 15 minutes and are continuous. The subsystems that we want to simulate in the city can be seen as the different devices in the house, e.g. the solar power generator reacting to an increase in solar radiation.

4 METHOD

To learn a model of the house we use Bayesian networks. We decided on using Bayesian networks as they do not only perform well when predicting, they also have an understandable structure that can give insight to the situation being modelled. We assume the reader to be familiar with the subject as described by Heckerman et al. [1]. In this section we will explain the specific parameters and choices made for our implementation.

Creating a Bayesian network can be seen as a two stage process. First, we define a network structure, nodes and how variables are related as a directed acyclic graph. Second, we must set the probabilities distributions of all the different nodes depending on their ancestors. We define our network as G, our nodes as V and the arcs between the nodes as A.

$$G = (V,A) \qquad (1)$$

During the first stage the structure of the network can be defined either manually or automatic. First, we define it manually, leveraging our knowledge of the system to define which nodes should be related. An advantage of this is that we will not get an over-fitted network based on the coincidences on our training data. A drawback is that we might miss relations we did not know beforehand, and that for larger networks defining a structure quickly becomes a complex task.

Because the drawbacks will become more significant in the smart city we also implement an automated approach to learn the structure from the data. We use a local search based metric, the “Look Ahead in Good Directions, LAGD, hill-climbing algorithm [9]. It looks at a sequence of best future moves, instead of a single move as is usually done for hill climbing. The algorithm to calculate a sequence has exponential time complexity, and therefore it first computes the best moves and then decides on a sequence. It scoring function considers conditional mutual information between the nodes. For a formal definition please see Abramovici et al. [9].

The second problem, learning the probability distributions, is solved by calculating direct estimates of the conditional probabilities using a set of training data. For a single node $X \in V$ we define the probability that $X = x$ as:

$$P(x) = P(X = x|Xi, ..., Xj) \qquad (2)$$

Here x are the different possible outcomes for that node. Simply counting the occurrences in our training data gives us a probability table for every possible outcome of X depending on every possible outcome of Xi, ...Xj . Because not every possible combination is observed we use a prior $\alpha = 0.5$ as an initial count on each variable.

As the observant reader might have noticed the equation 2 requires discrete variables, and thus we discretized all our variables into 10 equal sized buckets. The partition intervals where chosen based on a minimum description length principle. For more information and a formal definition we refer the reader to Fayyad et al. [12].

We input the values of our observed testing variables and use this knowledge to calculate the remaining variables. This is done by the junction tree algorithm [10], it calculates the marginal distribution for each unobserved node based on the values of the observed nodes. We take the most likely value for each node, set it as evidence and update the margins, till all the variables are set.

5 RESULTS

In this section we present the experimental results for the models presented in the previous section. In section 5.1 we will look at the structure of the networks and in section 5.2 we will analyse the performance of the complete models.

The data used was obtained from running the EnergyPlus simulator. We used two days of simulated data for training the model and two days for testing. The variables were all recorded at 15 minute intervals. The concrete implementation was done in Java using the WEKA library [11].

5.1 Network Structure

We first created a manual structure of the network. Since the selected domain is relatively simple with only 17 variables it could be largely comprehended by the analysis of the data. We looked at correlations between the variables, used our knowledge of the physical processes and experimented with a few possible networks to get a network that covered all the variables. Because of computational challenges some relations were simplified to restrict the number of parents of a node. Precautions were taken to make sure no relations were lost, for example, the solar radiation influencing the solar panel which in turn influences the battery. The arc from solar panel to battery was removed and instead it was linked directly to the radiation. This goes at the cost of a little accuracy but does not lose the relation. In Figure 1, a network created using the LAGD hill climber algorithm with at maximum 1 parent is shown. The figure does not show all the variables, as the nodes that do not have any arcs are left out for the sake of readability. The reasons why certain variables do not have arcs will be discussed in the next subsection. It is interesting that a lot of arcs seem to be reversed. Note; however, that setting evidence for any variable will update

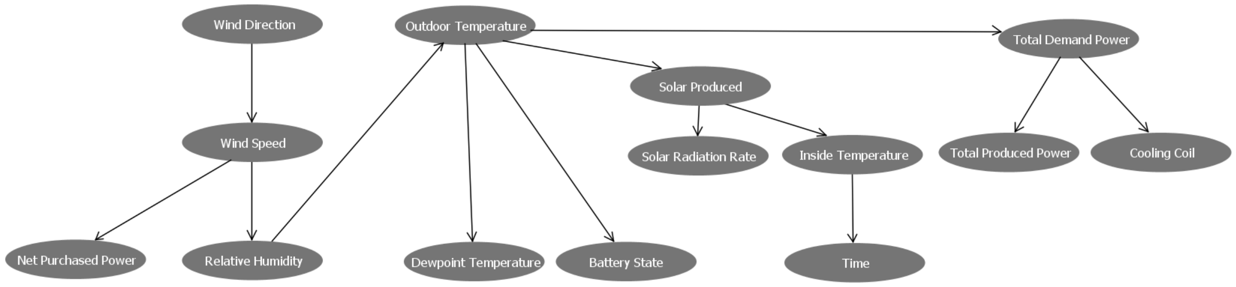


Figure 1: Network generated by LAGD Hill climber

margins throughout the network, and thus even a reverse relation is still captured. Automatic network generation can model inconsistent relations, for example, *Net Purchased Power* related to *Wind speed* seems to be over fitted on a coincidence in the data. The other curious relations, like *Solar Produced* and *Inside temperature* can be explained by the fact that all the input variables are closely related, on a warm day there will be more solar radiation etc.

5.2 Performance

To test the performance of the networks we computed the percentage of correctly predicted values per variable and the root mean-square-error (RMSE). However, even more important than predicting the right values is to predict the trends in the data. A few erroneous spikes are of lesser importance then missing a trend. Therefore we also plotted the data and did a visual analysis of the results.

	Baseline		Manual		Automatic	
Variable	%	E	%	E	%	E
Battery charge [J]	8	85e5	82	21e4	74	47e4
Heating coil	99	1.15	99	1.15	99	1.15
Washing machine	98	4.27	98	4.27	98	4.27
Total demand	51	87.54	51	87.54	60	15.17
Total produced	67	84.32	67	84.32	76	28.52
Cooling coil	69	46.49	35	49.45	77	22.25
Net purchased	65	35.61	65	35.61	65	35.61
Solar produced	60	88.31	81	34.58	82	35.37
Lights	72	35.53	88	12.46	72	35.53
Inside temp [C]	34	2.62	30	0.6	72	0.24
Overall	62		70		77	

Table 1: All variables are power in watt [W] if not stated otherwise. The columns labelled % depicts the percentage correctly classified. E is the RMSE, lower is better.

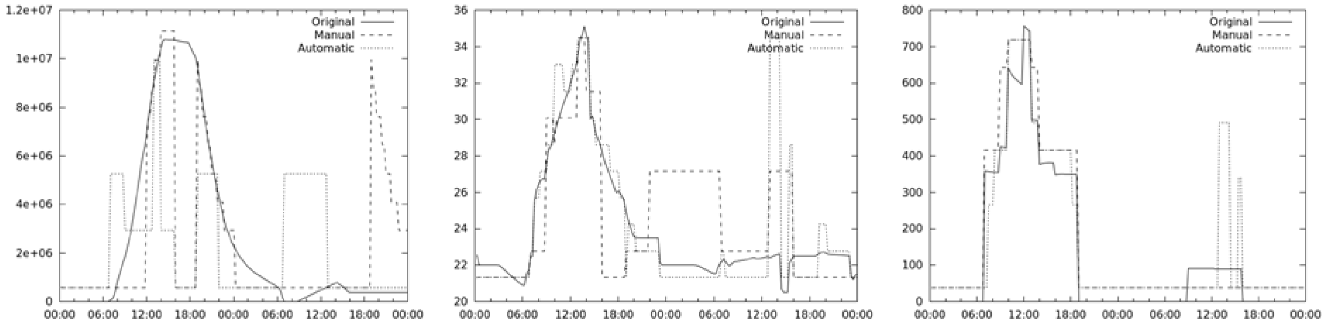
In Table 1 the percentage of correctly predicted values and the RMSE for each variable for three separate classifications is shown. First a baseline where the most probable bucket was chosen based on just priors. Second the manually constructed network and last the automatically learned network. For the heating coil and the washing machine even the baseline has an exceptionally good prediction power. This is because they are off in all but a few cases, showing the importance of complete training data. The manual network performed better on a subset of the variables, for example power used by lighting. Lighting can be clearly linked to time, this was not found by the LAGD algorithm as it does not occur often in the training data. However, in most cases automatic structure generation did perform better. Some of the interesting variables are the predicted values of inside temperature (Fig. 2b), solar power produced (Fig. 2c) and battery charged state (Fig. 2a). Note that the predicted data was discretized into buckets and for each datapoint the average of the bucket was used for the calculation of the RMSE and as data points in the graphs.

The predicted inside temperature over time shown in Figure 2b follows the general trends for both models, though the manual network is over-fitted on outside temperature and clearly performs worse. Another problem is that because of the discretisation some small changes are amplified. In Figure 2c the prediction of power produced by the solar panel is shown. It is closely correlated to one of the input variables, solar radiation, and is therefore quite accurately predicted. The second production increase is not seen in the graph for the manual network as the values are still in the range of the first bucket.

Predicted battery charge state over time can be seen in figure 2a. This is difficult to predict as it depends on many variables. As can be seen the manual network is over-fitted on a wrong variable; time. The first charge peak happens to coincide in the training and testing data and therefore the percentage of correct prediction is still high. The automatic network is not conclusive in predicting a trend. For variables like a battery, which cannot drop quickly and get back up again, it will be a valuable extension to consider the previous state as well.

In general, the automatically constructed network model performed better than the manually constructed one and possibly correct but yet unknown relations were found. The variables could be predicted relatively accurately and most trends present in the original data could also be found in the generated data.

Figure 2: Predictions of system parameters



(a) Amount of joules in the battery (b) Inside temperature (c) Solar panel production

4 CONCLUSION

We have shown that it is possible to build complex models from real world traces that model relations between subsystems. These models can then be used to simulate the system and generate more data based on a set of input variables. We have seen that trends in the data can be modelled and even single predictions can be used as an indication of expected data.

While automatic generation of a Bayesian network for a domain is possible, some expert knowledge will still be required to reduce over-fitting due to coincidences in the data, and to improve the network. Due to the nature of the Bayesian networks the cooperation with a domain expert can easily be established.

The biggest drawback of this method is that it largely depends on the available data. A lack of data will lead to incomplete or incorrect models.

The most important direction for future work will focus on taking the temporal nature of the network into account. Expanding to dynamic Bayesian networks or Hidden Markov Models will allow for an even more accurate prediction of trends. The challenge will be to cover the unknown parameter space that is not directly present in the training data. Introducing Gaussian probabilities to closer model the values produced by different sensors is another possible direction for future work. A third extension will handle tackling computational challenges that arise when the network sizes increases, those may be solved by creating a more hierarchical network structure.

References

[1] Heckerman D., Bayesian networks for data mining, Data mining and knowledge discovery, vol. 1, no. 1, 1997.

[2] Rudin C., Waltz D., Anderson N. R., Boulanger A., Chow M., Dutta H. Machine learning for the New York City power grid, IEEE Transactions Pattern Analysis and Machine Intelligence, vol. 34, no. 2, 2012.

[3] Fusco G., Looking for Sustainable Urban Mobility through Bayesian Networks, Cybergeo: European Journal of Geography, 2003.

[4] Lanini S., Water management impact assessment using a Bayesian network model, Proceedings of the 7th Conference on Hydroinformatics, 2006.

[5] Babovic V., Drcourt J., Keijzer M., and Hansen P. F., A data mining approach to modelling of water supply assets Urban Water, vol. 4, no. 4, 2002.

[6] Rong C., Sivakumar K. and Kargupta H., Collective mining of Bayesian networks from distributed heterogeneous data, Knowledge and Information Systems, vol 6, no. 2, 2004.

[7] Molina, J. L., John Bromley, J. L. Garca-Arstegui, C. Sullivan, and J. Benavente, Integrated water resources management of overexploited hydrogeological systems using Object-Oriented Bayesian Networks, Environmental Modelling & Software, vol. 25, no. 4, 2010.

[8] Boyle D., Yates D. and Yeatman E, Urban Sensor Data Streams: London 2013, IEEE Internet Computing, vol. 17, no. 6, 2013.

[9] Abramovici M., Neubach M., Fathi M. and Holland A., Competing fusion for bayesian applications, Proceedings of International Conference on Information Processing, vol. 8, p. 379, 2008.

[10] Lauritzen S. L., and Spiegelhalter D. J., Local computations with probabilities on graphical structures and their application to expert systems, Journal of the Royal Statistical Society. Series B, 1988.

[11] Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., Witte I. H., The WEKA Data Mining Software: An Update, SIGKDD Explorations, vol. 11, no 1, 2009.

[12] Fayyad U. and Irani K., Multi-interval discretization of continuous-valued attributes for classification learning, Chambery, France, 1993.

[13] Tavčar A., Piltaver R., Zupančič D., Šef T. and Gams M., Modeliranje Navad Uporabnikov Pri Vodenju Pametnih Hiš, Proceedings of Information Society 2013, p 114-117, 2013.

[14] US Department of Energy, EnergyPlus Energy Simulation Software, eere.energy.gov/buildings/energyplus/, accessed 08-2014.

INCLUSION OF VISUALLY IMPAIRED IN GRAPHICAL USER INTERFACE DESIGN

Mario Konecki
Faculty of Organization and Informatics
University of Zagreb
Pavlinska 2, 42000 Varaždin, Croatia
Tel: +385 42 390834
e-mail: mario.konecki@foi.hr

ABSTRACT

Visually impaired programmers have been included into programming industry since its very beginning and they were able to perform their jobs without difficulties. Graphical user interfaces and point and click method of instructing computers have created many difficulties for visually impaired programming professionals. Visually impaired have interest in programming just as everyone else and the means of their inclusion in overall software development process are important issues that need to be resolved. One of disadvantages for visually impaired is the lack of assistive technology that would enable them to design and create graphical user interfaces. In this paper the GUIDL (Graphical User Interface Description Language) system that is aimed to resolve the mentioned issues is presented and discussed.

1 INTRODUCTION

Inclusion of visually impaired in the world of computers and programming has been present from the beginning of computer mass usage [5]. Visually impaired have been able to use computers and perform various programming tasks by using assistive technology in the form of various text-to-speech synthesizers of which the most well-known are JAWS, HAL Screen Reader, COBRA, Window Eyes and Easy Web Browsing [7]. However, the graphical revolution in the world of computers has made using computers for visually impaired much more difficult since text-to-speech synthesizers were not able to represent the context and organization of graphical screens. Some problems that existing screen reading technologies came across are [7]:

- Interpretation of images – Screen readers are not able to adequately interpret images. Only properties and descriptions of images can be presented.
- Graphical layout and context – Screen readers read information in linear way that is not sufficient to interpret complex graphical user interfaces and screen organization.
- Reading of data tables – Because of linear way of reading information small tables are suitable and readable in this kind of approach but even moderately large tables are almost impossible to present.
- Graphical charts interpretation – Graphical charts are usually made of several different sub-elements and shapes which cannot be adequately interpreted.
- Robustness issue – Inability to cope with new technology and constant software development.

The same problems have emerged in programming domain where various graphical environments have appeared as well as the need to create graphical user interfaces by point and click method since textual description of graphical elements for visually impaired was too complicated and virtually impossible in practice [7]. And although this problem was not so prominent in the area of web programming since its textual coding nature, it was very real in the domain of classic desktop programming [6].

Inclusion of visually impaired as equals into all aspects of social and business life remains important issue and enabling visually impaired to design graphical user interfaces is one of its aspects. The interest of visually impaired for programming today is present and actual [1, 12]. There are over 130 blind programmers registered at the American Foundation of the blind programmers and programming is stated as potentially promising carrier opportunity for visually impaired in Europe [2, 11]. Inclusion of visually impaired into overall software development process which includes the design of graphical user interfaces is an actual and important issue. Its solution in a form of GUIDL (Graphical User Interface Description Language) system is proposed and described in the rest of this paper.

2 POSSIBLE APPROACHES TOWARDS SOLUTION

In order to solve the problem of inclusion of visually impaired into the process of overall software development and to enable them to design and create graphical user interfaces several possible approaches can be taken [7]:

- Interpreters of specific graphical elements and attributes of every development environment could be created

- Audio support for creation of graphical elements could be incorporated into programming environments
- A specific scripting language for every programming technology and environment could be developed

All mentioned approaches are time consuming and specific to particular programming language and environment. In order to provide a more universal solution several requirements must be satisfied [8]:

- Easy usage: system has to be simple and easy to use so it can be used by programmers but also by designers and other interested computer users.
- Intuitive, simple and understandable syntax: system's language that will be used to describe the graphical elements has to be intuitive, simple to use and easy to understand.
- Technology independence: system and its language for description of graphical user interfaces has to be applicable to various programming languages and development environments, not to be technology specific.
- Extensibility: system and its language have to be to be extensible so they can include support for new programming languages and corresponding development environments as well as new graphical element and attributes.

Mentioned requirements have been evaluated through conducted research that confirmed stated requirements and added some new requirements with the requirement for proper documentation being the most frequently mentioned.

3 GUIDL SYSTEM

In order to provide a more general solution a GUIDL (Graphical User Interface Description Language) system has been developed with GUIDL language as its core part. GUIDL language enables visually impaired to define all graphical elements in one place using language that is simple and has several assistive concepts such as:

- Predefined gradual sizes of forms
- Predefined gradual sizes of graphical elements
- Predefined width/height attribute values
- Division of forms into quadrants
- Possibility to position graphical elements into one of form's quadrants
- Possibility to define the position offset of forms
- Possibility to define the position offset of graphical elements
- Detection of problems with position of graphical elements (graphical element out of form boundaries)
- Automatic correction of problems with form's dimension and position (form out of screen boundaries)

In this way visually impaired are able to define entire user interface in just one place and one technology and GUIDL system then enables them to translate that interface into desired programming technology format which can then be included into native programming environment of chosen programming language. Conceptual model of GUIDL system is shown in Figure 1 [8].

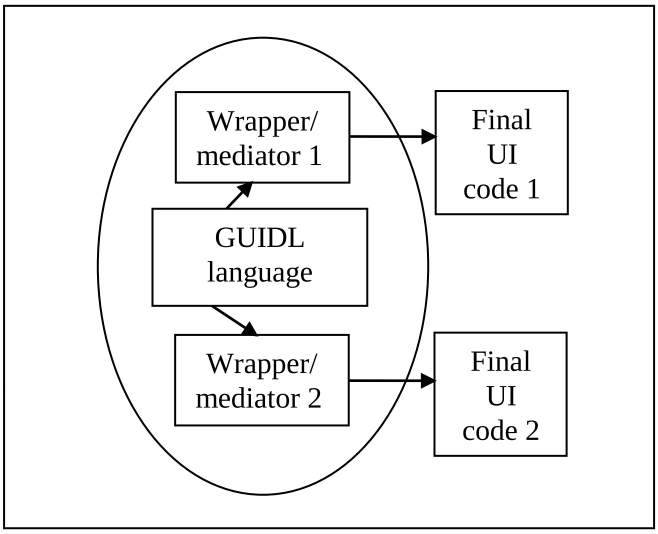


Figure 1: *Conceptual model of GUIDL system*

In GUIDL system visually impaired programmers start the development of programming language specific graphical user interface by defining the interface layout in GUIDL using all assistive built-in concepts. After the GUIDL code has been written, the GUIDL lexer performs lexical analysis. Lexical analysis uses scanner that reads the code character by character and transfers it to lexer which checks whether received characters form an array that can be identified as an acceptable string or token. The set of acceptable strings or tokens are defined in GUIDL context-free grammar [9] $G = (V, \Sigma, R, S)$ where V is the finite set of nonterminal symbols, Σ is the finite set of terminal symbols, R is the set of substitutions or production rules of from $A \rightarrow \alpha$ where A is some nonterminal symbol and α is a string over $V \cup \Sigma$.

S represents the start symbol which is nonterminal. Every string $x \in (V \cup \Sigma)^*$ which has the form of yAz can be turned into yαz by using production rule $A \rightarrow \alpha$ that substitutes A with α. Set of terminals defines the language's alphabet and $\Sigma \subset V$. The set of possible tokens or the language itself is defined as $L(G) = \{q \in \Sigma^* : S \rightarrow_G^* q\}$ or all strings of finite length that are composed of zero or more symbols from Σ in a way that particular string can be generated from start symbol by using zero or more steps which are defined by production rules.

The set of tokens produced by lexer is processed by GUIDL parser that compares the order of the tokens against defined grammar rules and in this way conducts the syntax validation of written code. After the syntax is validated the parser creates corresponding syntax tree from which the GUIDL generator generates the final graphical user interface code for specific programming language and environment. Described process is shown in Figure 2 [8].

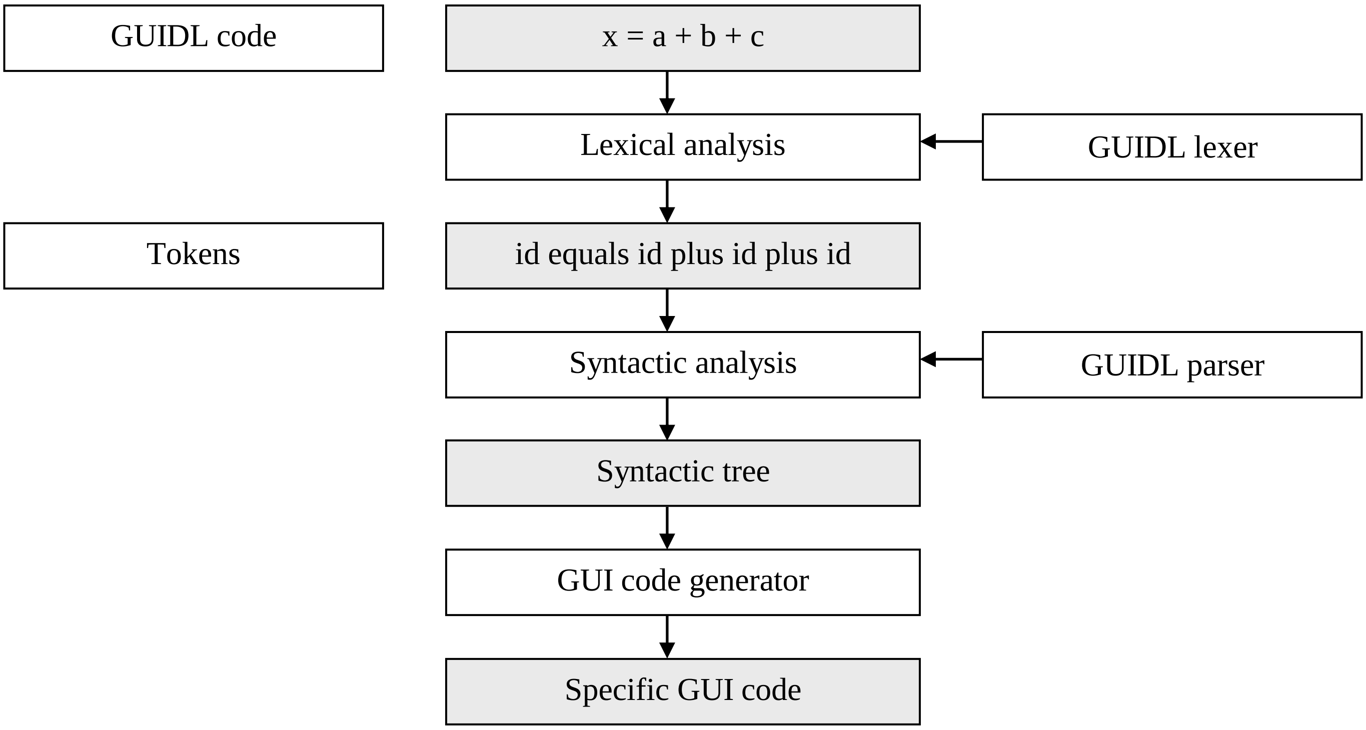


Figure 2: *Steps in using GUIDL system*

3.1 GUIDL syntax

GUIDL syntax has been designed to be simple and easy to use in order to enable visually impaired to perform their design tasks in quick and efficient manner. Its form is inspired by the simplicity of BASIC (Beginner's All-purpose Symbolic Instruction Code) syntax which has been created for beginners and for learning purposes. Partial GUIDL grammar in EBNF [10] form is given below.

project = projectcode, controlname, form;
projectcode = 'Project ' | 'project ';
form = formcode, controlname, formattributes, [controldeclarations], formend;
formcode = 'Frm ' | 'frm ';
formend = ('End' | 'end'), [eol];
controlname = qoute, word, qoute, eol;
formattributes = frmcommonattributes, windowstateattribute, {colorattribute};
frmcommonattributes = textattribute, frmrestcommonattributes;
frmrestcommonattributes = frmsizeattribute, locationattribute;
frmsizeattribute = sizecode, (frmsize | frmwidth, ws ,frmheight), eol;
locationattribute = locationcode, xposition, ws, yposition, eol;
frmsize = 'frmsize1' | 'frmsize2' | 'frmsize3'
locationcode = 'Location = ' | 'Location=' | 'location = ' | 'location=';
xposition = 'left' | 'center' | 'right';
yposition = 'top' | 'middle' | 'bottom';

4 GUIDL AS ASSISTIVE TECHNOLOGY

GUIDL system and its language are designed to be simple and understandable. Its main purpose is to provide visually impaired an assistive technology that would enable them easier creation of designed graphical user interfaces. There are several models that support development of assistive technology and one of them is CAT (Comprehensive Assistive Technology) [4] that was used in development of GUIDL system. CAT model is hierarchically structured with four aspects as is shown in Figure 3 [4].

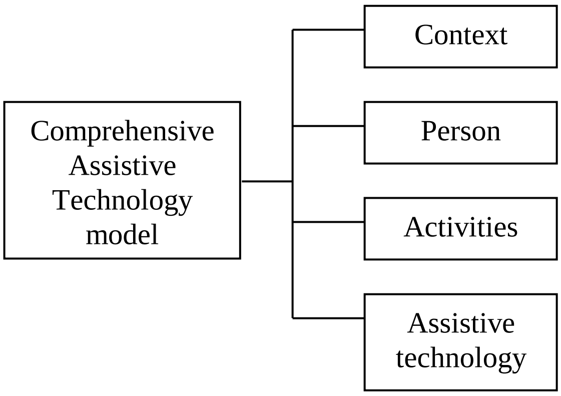


Figure 3: *CAT model*

In the case of GUIDL system development, the context is composed of several aspects such as: teams and working environment in which visually impaired programmers work, overall labor market, the attitude towards visually impaired as programmers, government measures to aid visually impaired in getting jobs, etc. Person that uses GUIDL is visually impaired person that wants to be a part of overall software development process and wants to work as a programmer of equal opportunities. Activities that are supported by GUIDL system are activities of quicker and easier development of graphical user interfaces that will enable visually impaired to be included as equals in the work of the development team that they belong to.

GUIDL system has a meaning of assistive technology [3] in a way that it is designed to overcome the obstacle of designing and creating graphical user interfaces in graphical point and click development environments. GUIDL system isn't build to replace development environments and to provide a specific isolated technology, it is designed to include visually impaired in actual programming technologies' parts in which they had the most difficulties. Visually impaired programmers start their development in a native programming environment where they set a project. Then they use GUIDL to design graphical user interfaces which are then included in already made project where visually impaired programmers continue with development and writing of program code which is something that they were always been able to do well by using another assistive technology in a form of text-to-speech synthesizers.

In this way visually impaired become included in the design process and creation of graphical user interfaces as well as in other segments of overall software development.

4.1 GUIDL system in practice

GUIDL system has been tested on 47 participants that were given the GUIDL prototype along with instructions and examples of its use. All participants were given several practical programming tasks through which they had to evaluate whether the GUIDL system will provide an efficient assistive role in their programs development activities. The evaluation of GUIDL system has shown that it indeed serves well as assistive technology. All performed tasks have been reported as easier and quicker when using GUIDL system than when using purely native technology. GUIDL system has also been reported as a suitable mean of inclusion of visually impaired into activities of overall software development that includes design of graphical user interfaces. Another important aspect of GUIDL as assistive technology is that it enables visually impaired to work in actual technologies rather than having isolated and specialized system.

5 CONCLUSION

Visually impaired programmers have been a part of computer revolution since its very beginning. Graphical user interfaces and occurrence of point and click development environments have left visually impaired in difficult position since existing assistive technology in the form of text-to-speech synthesizers could not cope well enough with rapid development and new graphical elements. Possible approaches towards solution of mentioned problems have been presented in this paper along with GUIDL system as assistive technology that is aimed at including visually impaired in graphical user interface design as an integral part of overall software development process.

Evaluation of GUIDL system has shown that it is suitable as an assistive technology and that it enables visually impaired to perform the actions of graphical interface design in an easier and more suitable manner. Adding new features and concepts to GUIDL system will be a part of future work.

References

[1] Alexander, S. Blind programmers face an uncertain future, available at http://www.cnn.com/TECH/computing/9811/06/blindprog.idg/index.html, accessed 14th August 2014, 1998.

[2] Cattani, R. The employment of blind and partially-sighted persons in Italy: A challenging issue in a changing economy and society, available at http://www.euroblind.org/media/employment/employment_Italy.doc, accessed: 25th March 2011.

[3] Francioni, J. M.; Smith, A. C. Computer Science Accessibility for Students with Visual Disabilities. *ACM SIGCSE Bulletin*, 34(1):91-95, 2002.

[4] Hersh, M. A.; Johnson, M. A. *Assistive Technology for Visually Impaired and Blind People*. Springer, 2008.

[5] Hodson, B. Sixties Ushers in Program To Train Blind Programmers. *Computer World Canada* 11, 2004.

[6] Konecki, M.; Kudelić, R.; Radošević, D. Challenges of the blind programmers. In *Proceedings of the 21st Central European Conference on Information and Intelligent Systems*, pages 473–476, 2010.

[7] Konecki, M.; Lovrenčić, A.; Kudelić, R. Making Programming Accessible to the Blinds. In *Proceedings of the 34th International Convention. Croatian Society for Information and Communication Technology, Electronics and Microelectronics – MIPRO*, pp. 820–824, 2011.

[8] Konecki, M.: A New Approach Towards Visual Programming for the Blinds. In *Proceedings of the 35th International Convention on Information and Communication Technology, Electronics and Microelectronics - MIPRO*, pages 935–940, 2012.

[9] Lewis, H. R.; Papadimitriou, C. *Elements of the Theory of Computation*. Prentice Hall, Inc., 1981.

[10] Information technology – Syntactic metalanguage – Extended BNF. ISO/IEC 14977. Geneva, Switzerland: ISO, 1996.

[11] bfi Steiermark, European Labour Market Report, available at http://eurochance.brailcom.org/download/labour-market-report.pdf, accessed: 25th March 2011.

[12] Blind Programming Project in NetBeans IDE, available at http://netbeans.dzone.com/videos/netbeans-ide-for-the-blind/, 2010, accessed: 20th August 2014.

MINING TELEMONITORING DATA FROM CONGESTIVE-HEART-FAILURE PATIENTS

Mitja Luštrek[1,2], *Maja Somrak*[1,2]
[1]Jožef Stefan Institute, Department of Intelligent Systems
[2]Jožef Stefan International Postgraduate School
e-mail: {mitja.lustrek, maja.somrak}@ijs.si

ABSTRACT

The Chiron project carried out an observational study in which congestive-heart-failure patients were telemonitored in two countries. Data from 1,068 recording days of 25 patients were gathered, consisting of 15 dynamic parameters (measured daily or continuously) and 49 static parameters (measured once or a few times during the study). The features derived from these parameters were mined for their association with the feeling of good/bad health. The findings mostly correspond to the current medical knowledge, although some may represent new insights.

1 INTRODUCTION

Telemonitoring of patients with chronic diseases is becoming technically increasingly feasible, but benefits for the patients are not always apparent, nor is it clear how to make the most of the data obtained this way. In the case of heart failure, two systematic literature reviews showed lower mortality resulting from telemonitoring [1][2], but in the trials they reviewed, telemonitoring was mostly compared with conventional care worse than what is offered today. Conversely, two large recent trials showed no benefit from telemonitoring [3]. However, the telemonitoring in these two trials was not very advanced – the monitored parameters were limited and no intelligent computer analysis was involved. We can conclude from this that as the conventional care improved, so should telemonitoring. One way to do so is by using intelligent computer methods on the gathered data, both to save the time of the medical personnel who would otherwise have to look at all the data themselves, and to uncover previously unknown relations in the data.

This paper describes the mining of telemonitoring data from congestive-heart-failure (CHF) patients gathered in the Chiron project [4]. The objective of this project was to develop a framework for personalized health management with a focus on telemonitoring. The Chiron patients were equipped with a wearable ECG, activity, body-temperature, sweating and sensors. In addition, their blood pressure, blood oxygen saturation, weight, and ambient temperature and humidity were measured [5]. The data gathered this way was fed into a decision-support system, whose objective was to estimate the health risk of the patients [6]. However, since there is not enough knowledge on how to associate the values of the various measured parameters with the risk, an observational study was carried out in the project with the intention to generate such knowledge. This paper presents an initial analysis of the data gathered in this study.

2 DATA FROM THE CHIRON STUDY

2.1 Data gathering and description

The data analyzed in this paper were gathered in the period from May 2013 to May 2014. The whole study included 38 CHF patients: 19 from the United Kingdom and 19 from Italy. However, some of the data were incomplete, so only the data of 12 patients from the UK and 13 patients from Italy were included in the analysis. These 25 patients together provided a total of 1,068 usable recording days. The data consists of 64 parameters carefully selected based on their relevance to CHF [7].

The initial measurements of 49 *static* parameters were taken for each of the patients at the beginning of the study. This data includes general patient information (age, gender, BMI, waist-to-hip ratio, smoking, etc.), their current medical treatments (beta blockers, anti-coagulants, ACE inhibitors, etc.), related health conditions (arrhythmias, hypertension, diabetes, etc.) and the results of a blood analysis (hemoglobin, lymphocytes, LDL/HDL cholesterol, blood glucose, Na and K levels, etc.). Some of these measurements were repeated periodically every few weeks to provide up-to-date information. However, the exact period varied from patient to patient and roughly half of the patients only had the measurements taken at the beginning of the study.

During the study, the patients were wearing vital-signs monitoring equipment [5] for several hours each day. The equipment consisted of an ECG device, two accelerometers places on the chest and thigh, a body-temperature and a humidity sensor. The ECG recordings were subsequently analyzed to extract the physiological parameters related to the heart rhythm: heart rate, QRS interval, QT interval, PR interval, T wave amplitude and R wave amplitude. The accelerometers continuously provided the patient's activity and energy-expenditure estimation. The temperature and humidity sensors provided the measurements of the skin temperature and sweating index in five-minute intervals.

The patients were also provided with a mobile application for generating weekly and daily reports. The patients reported their overall feeling of health with respect to the previous day on a daily basis (feeling much worse than yesterday, worse, the same, better or much better), and answered 13 questions about their health and well-being on a weekly basis. In addition, they reported measurements of systolic and diastolic blood pressure, body mass, blood oxygen saturation, and ambient temperature and humidity. These – together with the continuously monitored parameters – are labeled *dynamic* in Section 3.

The study also intended to gather data about hospital admissions and deaths, but no such events occurred during the study period. Therefore we decided to use the patients' self-reports of health instead. The analysis in this paper is based on the daily questions about the feeling of health.

2.2 Data preprocessing

The ECG and accelerometer data recordings required the most attention when preprocessing the data prior to the data mining. These two types of recordings also generated the vast majority of all the gathered data.

The ECG signal was already processed with the Falcon algorithm [5], producing an output where each heart beat is described with an 11-tuple. Because the tuples were not explicitly separated and some of them are incomplete, it was important to distinguish between them in order to extract the specified parameters. We used R-peaks in the ECG signal to identify distinct tuples. Additionally, a lot of the data was corrupt or missing, so those parts had to be removed.

Similar problems occurred when processing the accelerometer data. It was not possible to extract the information about the activity and energy expenditure if a recording of any one of the axes of either of the two sensors was missing. If a patient forgot to wear both sensors, or one had an empty battery, the data thus had to be discarded.

Finally, some data was not uploaded successfully to the servers due do connection problems, and some data are missing as a result of inconsistent patients' behavior.

All of the parameters that were measured continuously were further separated by the main activities of the day: during lying, sitting and moving separately (resulting in features labeled *per_act* in Section 3) or during all the activities together (*all_act*). The ratios of the durations of these three activities were calculated for each day. For every parameter that was measured continuously or multiple times per day, the average value (*avg*) and standard deviation (*sd*) were calculated; the calculations were done for separate activities and for the whole day.

The key value whose association with the other monitored parameters we study in this paper – the overall feeling of health – was reported by the patients relatively to the previous day. Since the value is not absolute (e.g., feeling well) but relative (e.g., feeling better or worse than yesterday), it is associated with the measurements of both the current and the previous day. Because of that we introduced features that represent changes of the parameters' values with respect to the previous day (*chg*). Again, the calculations were done for separate activities and for the whole day.

For the purpose of data mining, classes were assigned to the data. If each of the five distinct feelings of health corresponds to one class, the differences between them are too small. Therefore we decided to have only two classes:

- Much worse vs. much better (*MW-MB*)
- Much worse or worse three times in a row vs. much better or better three times in a row (*MW3-MB3*)
- Much worse or worse vs. much better or better (*MWW-MBB*)
- Much worse vs. everything else (*MW-E*)
- Much worse or worse three times in a row vs. everything else (*MW3-E*)
- Much worse or worse vs. everything else (*MWW-E*)

The majority of the data instances have the class 'feeling the same as yesterday', while very few instances have 'feeling much better' or 'feeling much worse'. Because of this, the first three classes result in discarding the majority of the instances (only 69, 101 or 285 instances remain), while the last three use all 1,086 of them. Since classes are imbalanced, particularly in the last three cases, we used cost-sensitive classification, with the costs of misclassifications compensating for the imbalances.

3 MINING THE DATA

Since the number of combinations of data-mining algorithms, features and classes is huge, we designed a three-step data-mining procedure (described in detail in Sections 3.1–3.3):

1. Selection of algorithms that classify the data with a high accuracy and yield understandable models
2. Using the selected algorithms, selection of features that classify the data with a high accuracy and are understandable
3. Using the selected algorithms and features, selection of classes that result in accurate models

At the end of these three steps, we ended up with a number of interesting models, some of which are presented in Section 3.4.

3.1 Selection of algorithms

In the first step we used MW3-MB3 classes and the avg subset of dynamic all_act features. We compared several algorithms from the Weka suite [8] shown in Table 1. We selected the underlined algorithms for the experiments in Sections 3.2 and 3.3 due to their accuracy and in the case of JRip to have another understandable algorithm.

Table 1: Comparison of data-mining algorithms

Algorithm	Accuracy
<u>Random Forest</u>	79.3 %
<u>Naive Bayes</u>	77.4 %
<u>J48</u>	76.3 %
<u>SVM, Puk kernel</u>	74.5 %
SVM, linear kernel	74.2 %
SGD	73.8 %
Multilayer Perceptron	73.2 %
<u>JRip</u>	71.9 %
kNN, k = 1	60.9 %
kNN, k = 2	56.2 %
kNN, k = 3	47.8 %
SVM, RBF kernel	40.1 %

3.2 Selection of features

We first compared predefined features sets described in Section 2. Since the number of combinations is large, we proceeded in several sub-steps. First, we compared subsets of dynamic all_act features, finding that only avg and avg + chg subsets performed better than the rest. The results are shown in the first segment of Table 2 with the highest accuracy for each algorithm in bold. Second, we added per_act features to these two subsets of features, finding the extended features worse than all_act features alone (second segment of Table 2). And third, we combined these two subsets of features with static features, finding them best of all (third segment of Table 2). However, given the small number of patients, it is likely that the static features identified individual patients instead of taking into account their general characteristics. Because of that we retained all the underlined features for experiments in Section 3.3.

Table 2: Comparison of predefined feature sets

Algorithm Features	Naive Bayes	SVM, Puk	JRip	J48	Random Forest
Dynamic, all_act, avg + chg	75.5	**80.0**	70.6	**76.9**	**80.3**
Dynamic, all_act, avg	**77.4**	74.5	**71.9**	76.3	79.3
Dynamic, all_act, avg + sd	75.3	73.1	70.9	73.3	77.7
Dynamic, all_act, avg + chg + sd	74.0	78.7	70.3	75.2	78.3
Dynamic, all_act, chg + sd	67.1	78.6	64.6	64.9	71.9
Dynamic, all_act, chg	62.1	71.2	55.5	64.8	64.4
Dynamic, all_act, sd	58.2	65.4	63.0	64.6	66.9
Dynamic, all_act + per_act, avg	**77.0**	**72.5**	**71.6**	75.7	78.4
Dynamic, all_act + per_act, avg + chg	73.4	71.8	71.0	**76.7**	**79.1**
Dynamic + static, all_act, avg	77.5	79.2	75.5	76.4	79.3
Dynamic + static, all_act, avg + chg	**77.8**	**80.4**	**77.0**	**79.6**	**80.5**

We also tested automatic feature selection methods from the Weka suite. None of the methods performed well on its own, so we used the features selected by at least two methods out of the following: Correlation-based Feature Subset, Gain Ratio, ReliefF, Symmetrical Uncertainty and Wrapper (the end result of the Wrapper approach was the union of features selected when each of the five algorithms selected in Section 3.1 were used). As the starting point, we used all features, all dynamic features, and avg + chg subset of all_act dynamic features. The results in Table 3 show that the first and third of these starting points resulted in the best models obtained so far, although we retained all the underlined features for the experiments in Section 3.3.

Table 3: Comparison of automatic feature selection

Algorithm Features	Naive Bayes	SVM, Puk	JRip	J48	Random Forest
All features, FS	75.5	**80.0**	70.6	**76.9**	**80.3**
Dynamic, all_act, avg + chg, FS	**77.4**	74.5	**71.9**	76.3	79.3
Dynamic + static, all_act, avg + chg	75.3	73.1	70.9	73.3	77.7
Dynamic, all_act, avg + chg	74.0	78.7	70.3	75.2	78.3
Dynamic, all_act, avg	67.1	78.6	64.6	64.9	71.9
Dynamic, FS	62.1	71.2	55.5	64.8	64.4

3.3 Selection of classes

We compared the accuracies of different classes on all the algorithms selected in Section 3.1 and all the features selected in Section 3.2. In Table 4 we report the F-measure for the Random Forest algorithm (most accurate overall), averaged over all the features. The F-measure was chosen because of the class imbalance, particularly for the three 'vs. everything else' pairs of classes. One can see that MW3-MB3 performed best, probably because it strikes the best balance between the difference between the two classes in the pair, and the number of instances in the dataset. MW-MB may have too few features, while in the other cases the difference between the two classes is too small.

Table 4: Comparison of classes

Classes	MW-MB	MW3-MB3	MWW-MBB	MW-E	MW3-E	MWW-E
F-measure	0.77	**0.79**	0.66	0.55	0.56	0.61
Instances	69	101	285	1,068	1,068	1,068

3.4 Interesting models

Classification models were built with the J48 and JRip algorithms (being the most understandable of the five selected in Section 3.1) on all the features selected in Section 3.2. Two examples are presented in Figure 1 and Figure 2. They show that a high heart rate (*HR_avg_all_activities* in the figures) and short QRS interval (*QRS_avg_all_activities*, a feature of the ECG signal) are associated with the feeling of good health, which corresponds to the existing medical knowledge. Increased weight (*DRWChg*) is associated with bad health, which makes sense, since it often signifies excess fluid retention, a common problem of CHF patients. Low humidity (*HumA*) and decrease in humidity (*HumAChg*) are associated with good health, which matches the medical opinion that CHF patients often badly tolerate humid weather, although there is little hard evidence for this. Oxygen saturation (*DRS*) below 97 % is associated with bad health in the second model, which is normal, since the saturation in healthy individuals is 96 % – 100 %. Finally, the first model associates high systolic blood pressure (*SBP*) and the second low diastolic blood pressure (*DBP*) with good health. This is expected in CHF patients, since their hearts have problems pumping out enough blood (low systolic blood pressure) as well accepting enough blood (high diastolic blood pressure).

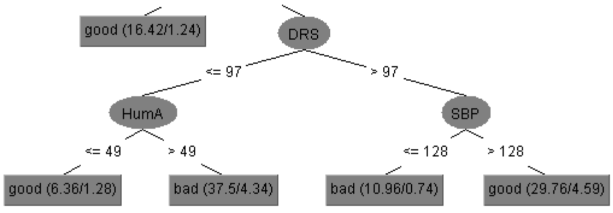


Figure 1: J48 classification tree on the avg subset of all_act dynamic features

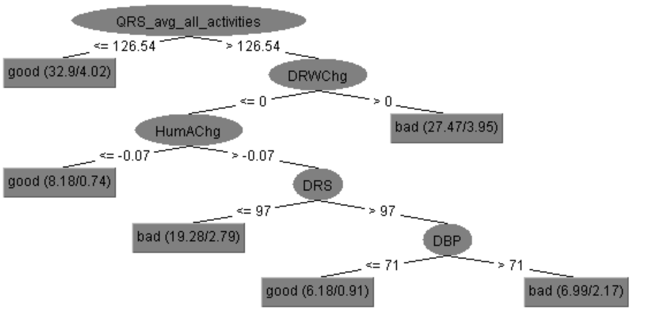


Figure 2: J48 classification tree on the avg + chg subset of all_act dynamic features

4 CONCLUSION

Telemonitoring can provide huge quantities of medically relevant data, which has the potential to revolutionize the care of patients with chronic diseases. However, before this can happen, the data must be properly interpreted, for which the current knowledge is not yet entirely adequate. This paper presents the data gathered by telemonitoring of CHF patients, and the first attempt to uncover interesting relations in the data by data mining. A systematic procedure for the selection of appropriate data-mining algorithms, features and classes was designed, whose output were a number of models associating telemonitored parameters with the feeling of good or bad health. The models correspond quite well to the current medical knowledge, which demonstrates the validity of our approach.

In the future, we need to solve the technical difficulties with extracting the ECG parameters and compute some new features that may be relevant (e.g., QT interval prolongation, a feature of the ECG signal that is known to be associated with cardiovascular problems). Furthermore, the models resulting from data mining must be carefully examined by cardiologists, both the models presented in the paper and others. Those that contain hitherto unknown relations may be even more important than those that correspond to the current medical knowledge, since the relations in them may yield new and important insights. Finally, it would be desirable to study data that contain events such as hospital admissions or even deaths, since the findings on such data would be more reliable than on data that only contains self-reported feeling of health. However, another observational study would be needed for that, which is a difficult proposition that would require substantial funding.

Acknowledgement

This research described in this paper was carried out in the Chiron project, which was co-funded by the ARTEMIS Joint Undertaking (grant agreement # 2009-1-100228) and by national authorities.

References

[1] C. Klersy, A. De Silvestri, G. Gabutti, F. Regoli, A. Auricchio. A meta-analysis of remote monitoring of heart failure patients. *Journal of the American College of Cardiology* 54, 2009, pp. 1683–1694.

[2] S. C. Inglis, R. A. Clark, F. A. McAlister, S. Stewart, J. G. Cleland. Which components of heart failure programmes are effective? A systematic review and meta-analysis of the outcomes of structured telephone support or telemonitoring as the primary component of chronic heart failure management in 8323 patients: abridged Cochrane Review. *European Journal of Heart Failure* 13, 2011, pp. 1028–1040.

[3] C. Sousa, S. Leite, R. Lagido, L. Ferreira, J. Silva-Cardoso, M. J. Maciel. Telemonitoring in heart failure: A state-of-the-art review. *Revista Portuguesa de Cardiologia* 33 (4), pp. 229–239.

[4] Chiron project. http://www.chiron-project.eu/.

[5] E. Mazomenos, J. M. Rodríguez, C. Cavero, G. Tartarisco, G. Pioggia, B. Cvetković, S. Kozina, H Gjoreski, M. Luštrek, H. Solar, D. Marinčič, J. Lampe, S. Bonfiglio, K. Maharatna. Case Studies. In *System Design for Remote Healthcare*, 2014, pp. 277–332.

[6] M. Luštrek, B. Cvetković, M. Bordone, E. Soudah, C. Cavero, J. M. Rodríguez, A. Moreno, A. Brasaola, P. E. Puddu. Supporting clinical professionals in decision-making for patients with chronic diseases. *Proc. IS 2013*, pp. 126–129.

[7] P. E. Puddu, J. M. Morgan, C. Torromeo, N. Curzen, M. Schiariti, S. Bonfiglio. A clinical observational study in the Chiron project: Rationale and expected results. In *Impact Analysis of Solutions for Chronic Disease Prevention and Management*, 2012, pp. 74–82.

[8] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten. The WEKA data mining software: An update. *SIGKDD Explorations* 11 (1), 2009, pp. 10–18.

APPROXIMATING DEX UTILITY FUNCTIONS WITH METHODS UTA AND ACUTA

Matej Mihelčić[1,3], *Marko Bohanec*[2]
[1] Ruđer Bošković Institute, Division of Electronics, Laboratory for Information Systems, Croatia
[2] Jožef Stefan Institute, Department of Knowledge Technologies, Jamova 39, Ljubljana, Slovenia
[3] Jožef Stefan International Postgraduate School, Jamova 39, Ljubljana, Slovenia
e-mail matej.mihelcic@irb.hr, marko.bohanec@ijs.si

ABSTRACT

DEX is a qualitative multi-criteria decision analysis (MCDA) method, aimed at supporting decision makers in evaluating and choosing decision alternatives. We present results of a preliminary study in which we experimentally assessed the performance of two well-known MCDA methods UTA and ACUTA to approximate qualitative DEX utility functions with piecewise-linear marginal utility functions. This is seen as a way to improve the sensitivity of qualitative models and provide a better insight in DEX utility functions. The results indicate that the approach is in principle feasible, but at this stage suffers from problems of convergence, insufficient sensitivity and inappropriate handling of symmetric functions.

1 INTRODUCTION

Multi criteria decision analysis (MCDA) [1] is an approach concerned with structuring and solving decision problems involving multiple criteria. MCDA provides a number of methods [2] to create a decision model from information provided by the decision maker. This information can be given in many ways, for instance by constructing evaluation functions directly, by providing parameters (such as criteria weights) to some predefined functions, by giving examples of decisions, or by pairwise comparison of a subset of decision alternatives. Methods also differ in the representation of this information (e.g., quantitative or qualitative) and their primary aim (choosing the best alternative, ranking several alternatives, classifying alternatives into predefined discrete classes, etc.).

Bridging the gap between different MCDA methods is sometimes highly desirable and may have a great practical value. In this work, we try to combine two MCDA methods: DEX and UTA. DEX [3] is a qualitative method; it employs discrete attributes and discrete utility functions defined in a point-by-point way (see section 2.1). This makes DEX suitable for classifying decision alternatives into discrete classes. On the other hand, UTA [9, 10] is a quantitative method that constructs numerical additive utility functions from a provided subset of alternatives (see section 2.2).

This work is motivated by the expectation that DEX's functionality would have been substantially enhanced if we were able to convert its discrete utility functions to numerical ones in some suitable way: first, the newly obtained numerical evaluations would facilitate an easy ranking and comparison of alternatives, especially those that are assigned the same class by DEX; second, the sheer form of numerical functions may tell us more about the properties of underlying DEX functions, which make them useful for verification, representation and justification of DEX models.

There have already been several attempts to approximate DEX utility functions with numeric ones for various purposes. A linear approximation method is commonly used in DEX to assess the importance (weights) of criteria [4, 5]. An early method for ranking of alternatives and improving the sensitivity of evaluation has been proposed in [6] and is now referred to as QQ [7]. Recently, extensive research has been carried out to approximate DEX functions with copulas [7, 8]. However, no known attempts have been made so far to approximate DEX functions with piecewise-linear marginal utility functions, as provided by UTA.

The aim of this study was to experimentally assess the performance of UTA and its variant, ACUTA [11], on a collection of typical DEX functions. The experiments were carried out using two software tools: DEXi [4] to develop DEX functions and Decision Deck [12] to run (AC)UTA.

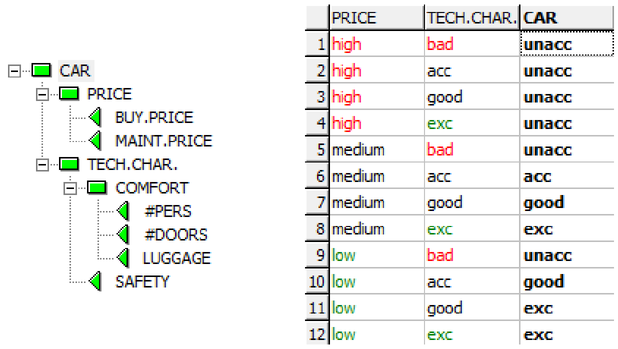
2 METHODS AND TOOLS

2.1 DEX and DEXi

DEX [3] is a qualitative MCDA method for the evaluation and analysis of decision alternatives, and is implemented in the software DEXi (http://kt.ijs.si/MarkoBohanec/dexi.html) [4]. In DEX, a decision model consists of hierarchically structured attributes: the hierarchy represents the decomposition of the decision problem into smaller sub problems, and attributes at higher levels of the hierarchy depend on those on lower levels. Figure 1 (left) shows an example of a tree of attributes for evaluating cars [4].

In the context of this paper, it is important to understand that all attributes in DEX models are qualitative and can take values represented by words; for instance, the attribute PRICE in Figure 1 can take the values *high*, *medium* and *low*. Furthermore, the aggregation of attributes at some level in the tree is defined by decision tables that consist of elementary decision rules. For example, the table in Figure 1 (right) defines the aggregation of two lower-level attributes PRICE and TECH.CHAR into the higher-level attribute

CAR: the values of CAR are specified for all combinations of values of PRICE and TECH.CHAR. Essentially, this means that utility functions in DEX are discrete and defined in a point-by-point way. This is illustrated in Figure 2, which graphically represents the same function as in Figure 1, so that each row of Figure 1 is represented by a dot in Figure 2. The connecting lines are used only for visualization and are not part of function definition.



	PRICE	TECH.CHAR.	CAR
1	high	bad	unacc
2	high	acc	unacc
3	high	good	unacc
4	high	exc	unacc
5	medium	bad	unacc
6	medium	acc	acc
7	medium	good	good
8	medium	exc	exc
9	low	bad	unacc
10	low	acc	good
11	low	good	exc
12	low	exc	exc

Figure 1: A DEX model and a utility function example [4].

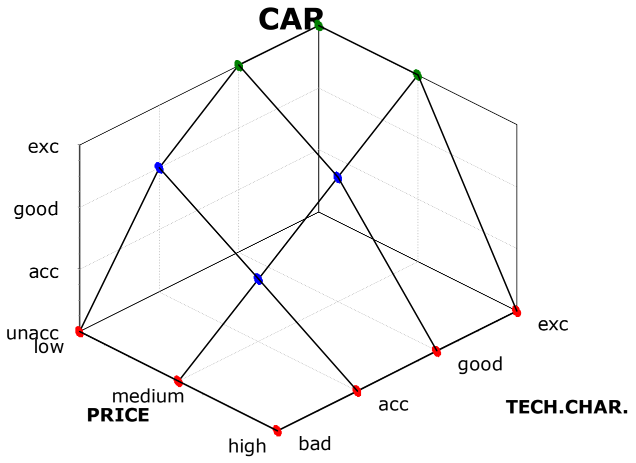


Figure 2: Graphical presentation of the CAR decision table.

Formally, a DEX utility function is defined over a set of criteria $x_1, x_2, \dots, x_n$, where all criteria are discrete and can take values from the corresponding value scales $D(x_i)$. A utility function U maps x to the higher-level attribute y:

$$U: D(x_1) \times D(x_2) \times \cdots \times D(x_n) \to D(y)$$

U is represented by a decision table that consists of elementary decision rules, where each rule defines the value of U for some combination of argument values:

$$\langle x_1, x_2, \dots, x_n \rangle \to y$$

For experiments in this study, we used a number of DEX utility functions, but in this paper we will present only three:

1. CAR function, as defined in Figures 1 and 2;
2. YM: defined over three attributes ($n = 3$), all the attributes have five values. The function is symmetric and represents a very common DEX function, which behaves as $\min(x_1, x_2, x_3)$ when any of the arguments takes the lowest possible value, and as a qualitative average of x_1, x_2 and x_3 otherwise.
3. YW: defined on the same space as YM, it represents an asymmetric DEX function defined with weights [4]; the weights assigned to the three arguments are 60%, 30% and 10%, respectively.

All these functions are defined completely for all combinations of values of their arguments.

2.2 UTA and ACUTA

The UTA method (UTilité Additive) [9,10] is used to assess utility functions which aggregate multiple criteria in a composite criterion used to rank the alternatives. Similarly as DEX, it uses a subjective ranking on a subset of the alternatives. On this basis, it creates piecewise-linear marginal utility functions.

For a set of alternatives A, $a \in A$, numerical criteria $g = (g_1, g_2, \dots, g_n)$, and the utility function $U(g) = U(g_1, g_2, \dots, g_n)$, the marginal utility functions u_i are approximated with:

$$u_i[g_i(a)] = u_i(g_i^J) + \frac{g_i(a) - g_i^J}{g_i^{J+1} - g_i^J}\left[u_i(g_i^{J+1}) - u_i(g_i^J)\right]$$

It is assumed that each attribute's values are divided to $\alpha_i - 1$ equally-sized intervals $[g_i^J, g_i^{J+1}]$.

The marginal utility functions u_i are constructed by solving the linear programming problem

$$\min \mathrm{F} = \textstyle\sum_{\mathrm{a} \in \mathrm{A}} \sigma(\mathrm{a})$$

under the constraints:

$$\sum_{i=1}^{n} u_i[g_i(a)] - u_i[g_i(b)] + \sigma(a) - \sigma(b) \geq \delta, aPb$$

$$\sum_{i=1}^{n} u_i[g_i(a)] - u_i[g_i(b)] + \sigma(a) - \sigma(b) = 0, aIb$$

$$u_i(g_i^{J+1}) - u_i(g_i^J) \geq s_i, \forall i \in \{1, \dots, n\}, J \in \{1, \dots, \alpha\}$$

$$\sum_{i=1}^{n} u_i(g_i^*) = 1$$

$$u_i(g_{i*}) = 0, u_i(g_i^J) \geq 0, \sigma(a) \geq 0,$$
$$\forall i \in \{1, \dots, n\}, J \in \{1, \dots, \alpha\}, \forall a \in A$$

Here, $\sigma(a)$ denotes potential error relative to the starting utility $U[g(a)]$. g_i^* and g_{i*} denote the high and low bounds of g_i respectively. P and I respectively denote strict preference and indifference relations.

In some cases there can be many utility functions that can represent the preferences specified. The utility functions are then assessed by means of post-optimality analysis [9].

The ACUTA method [11] offers an improvement upon UTA. It proceeds by finding an analytic center of the additive value functions that are compatible with some user assessments of preferences. In this way, ACUTA solves the model selection problem present in the UTA method when there are multiple valid solutions. Similarly as UTA, it constructs marginal utility functions by solving a constrained optimization problem, see [11] for details.

In order to approximate DEX utility functions with (AC)UTA, we mapped qualitative DEX attributes $x \in D(x)$ to equidistant numerical scales $g = [1, |D(x)|]$.

2.3 Decision Deck and Diviz

The Decision Deck (http://www.decision-deck.org/project/) is a project aimed at developing an open-source MCDA software platform [12]. Diviz is a software component developed in Decision Deck aimed at designing, executing and sharing MCDA methods, algorithms and experiments [12]. Diviz enables combining programs that implement MCDA algorithms in a modular way and connecting them in terms of workflows.

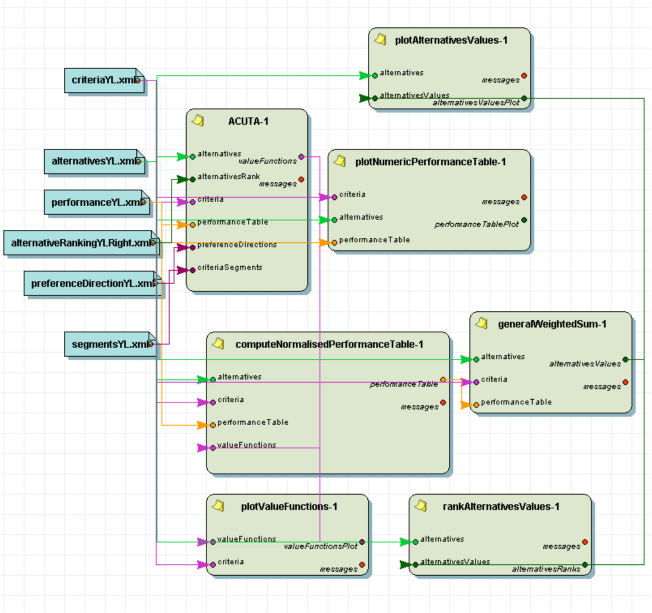


Figure 3: The ACUTA decision support workflow.

Figure 3 shows the workflow used in this study to run ACUTA. The input consists of six datasets. The *Criteria* file contains names and ID's of the decision criteria, the *Alternatives* file contains names and ID's of the alternatives, the *PerformanceTable* file contains the attribute values for each alternative, the *AlternativeValues* file contains a ranking of a small sample of the alternatives determined by the decision maker (usually called a priori ranking), the *PreferenceDirection* file indicates preferred optimization direction, and the *NumberofSegments* file defines the number of segments to which the attribute values are split. The output of the workflow is a rank of alternatives given their attribute values and the a priori ranking.

3 RESULTS

Several problems were detected when we attempted to approximate DEXi utility functions with (AC)UTA in Diviz. First, the standard UTA method could not handle DEX utility functions and returned an error message:

```
Execution terminated, but no result were produced:
you probably hit a bug in the service. […]
```

In order to get any results, we had to take only a subset of the rules, that is, remove a subset of entries from the UTA performance table.

The second problem with UTA was setting the a priori alternative ranking (i.e., the target attribute) in a way that would allow the method to converge. We used the inverse DEX attribute label score as a priori rank for the UTA method. As a result, all the alternatives with the same DEXi label score were indifferent for UTA. This required us to take only a small, targeted subset of available a priori ranks.

Overall, the results produced by UTA were poor and did not accurately approximate input functions.

The ACUTA method performed much better and the models were built on the whole domain of the DEX utility functions. However, we did experience convergence issues when using inverse DEX label attribute score as a priori rank for all the alternatives, so we had to take a subset that allowed the method to converge. The convergence error message reported by ACUTA was as follows:

```
Error  –  failed  to  converge,  due  to  bad
information. Please check your data, rescale the
problem, or try with less constraints.
```

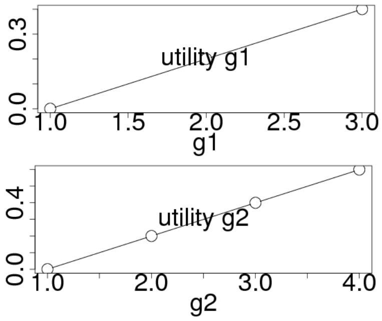


Figure 4: ACUTA results for DEX function Car.

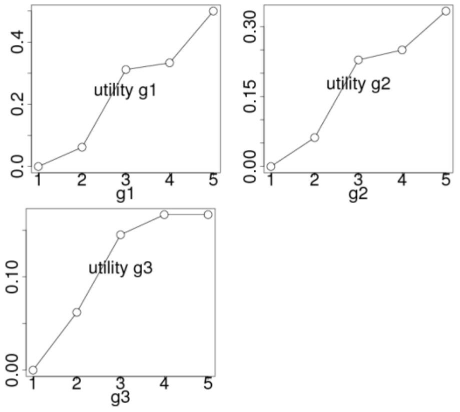


Figure 5: ACUTA results for DEX function YM.

The results for the Car utility function are shown in Figure 4, where *g1* and *g2* indicate DEX attributes PRICE and TECH.CHAR. Both marginal utility functions properly increase, and in *g1* the relations between utility values in points 1, and 2 appear right, however utility value in point 3 is too high. We noticed similar behavior in function g2.

Figure 5 shows results for the DEX function YM. In our opinion, marginal utility functions approximate YM quite well, however they indicate a common problem encountered in the experiments: YM is symmetric, therefore ACUTA's marginal functions should be equal to each other, but they are not. In this way, the resulting representation does not properly capture the symmetricity of the original function.

Marginal utility functions in Figure 6 correctly indicate that YW is asymmetric and, observing function's maximum values, that the attributes *g1*, *g2*, and *g3* are less and less important. However, some sections of these functions are almost constant, which does not hold in the original function.

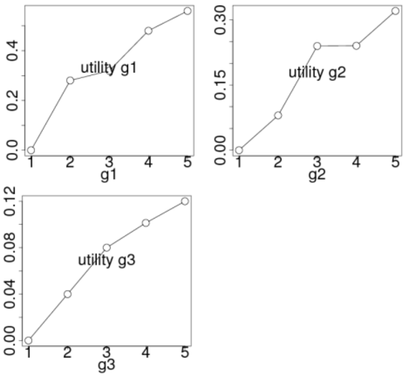


Figure 6: ACUTA results for DEX function YW.

4 CONCLUSION

In this preliminary study we tried to approximate several DEX utility functions by using the basic UTA method and its derivative, ACUTA. In general, the approach turned out to be feasible, producing marginal utility functions from DEX utility functions, which are defined by points in a discrete multidimensional space. The obtained functions are easy to interpret and do provide useful information about DEX attributes and scales (e.g., numeric utility value for each discrete attribute value), and the underlying DEX utility functions (e.g., about relative importance of attributes). Therefore, the approach is useful for representing and understanding DEX utility functions: the representation consists of a set of additive utility functions that represent attribute trends and importance's that cannot be easily observed by examining DEX utility functions themselves.

On the other hand, we encountered several problems with the methods and their implementation. UTA rarely gives any results on the original DEX functions, and even after tweaking the inputs the results were unsatisfactory. ACUTA performs much better, it can work on the whole domain of the DEX function, but the a priori rank subset needs to be carefully chosen in order to avoid convergence problems. The theoretical reasons for convergence problems of these methods are still to be determined.

Marginal utility functions, generated by ACUTA, in principle appropriately represent the marginal behavior of DEX attributes, but they exhibit two common problems:

- insufficient sensitivity to changes of attribute values (some sections of ACUTA functions are (almost) constant even though the underlying DEX function is not);
- inappropriately representing symmetric DEX functions with mutually different marginal utility functions.

In future work, we wish to theoretically and empirically address these issues and alleviate these problems, either by adopting some other method from the rich set of UTA-related methods [10], by adapting (AC)UTA to specific properties of DEX functions, or by developing entirely new methods. Eventually, the method should be able to deal with all type of DEX functions, including large ones, incompletely defined ones and those defined with distributions of classes.

References

[1] Ehrgott M., Figueira J.R., Greco S.: *Trends in Multiple Criteria Decision Analysis,* International Series in Operations Research & Management Science, Vol. 142, New York: Springer, 2010.

[2] Figueira J.R., Greco S., Ehrgott M.: *Multiple Criteria Decision Analysis: State of the Art Surveys,* Boston: Springer, 2005.

[3] Bohanec M., Rajkovič V., Bratko I., Zupan B., Žnidaršič M.: DEX methodology: Three decades of qualitative multi-attribute modelling. *Informatica* 37, 49–54, 2013.

[4] Bohanec M.: *DEXi: Program for Multi-Attribute Decision Making, User's Manual, Version 4.00.* IJS Report DP-11340, Ljubljana: Jožef Stefan Institute, 2013.

[5] Bohanec M., Zupan B.: A function-decomposition method for development of hierarchical multi-attribute decision models. *Decision Support Systems* 36, 215–233, 2004.

[6] Bohanec M., Urh B., Rajkovič V.: Evaluating options by combined qualitative and quantitative methods. *Acta Psychologica* 80, 67–89, 1992.

[7] Mileva-Boshkoska B., Bohanec M.: A method for ranking non-linear qualitative decision preferences using copulas. *International Journal of Decision Support System Technology* 4(2), 42–58, 2012.

[8] Mileva-Boshkoska B., Bohanec M., Boškoski P., Juričić Đ.: Copula-based decision support system for quality ranking in the manufacturing of electronically commutated motors. *Journal of Intelligent Manufacturing,* doi: 10.1007/s10845-013-0781-7, 2013.

[9] Jacquet-Lagreze E., Siskos J.: Assessing a set of additive utility functions for multicriteria decision-making, the UTA method, *European Journal of Operational Research* 10(2), 151–164, 1982.

[10] Siskos Y., Grigoroudis E., Matsatsinis N.F.: UTA methods. In: *Multiple Criteria Decision Analysis: State of the Art Surveys*, 297–343, Boston: Springer, 2005.

[11] Bous G., Fortemps P., Glineur F., Pirlot M.: ACUTA: A novel method for eliciting additive value functions on the basis of holistic preference statements, *European Journal of Operational Research* 206(2), 435–444, 2010.

[12] Ros J.C.: *Introduction to Decision Deck–Diviz: Examples and User Guide*, Technical report DEIM-RT-11-001, Tarragona: Universitat Rovira i Virgili, 2011.

COMPARING RANDOM FOREST AND GAUSSIAN PROCESS MODELING IN THE GP-DEMO ALGORITHM

Miha Mlakar, Tea Tušar, Bogdan Filipič

Department of Intelligent Systems, Jožef Stefan Institute and
Jožef Stefan International Postgraduate School, Jamova cesta 39, SI-1000 Ljubljana, Slovenia
e-mail: {miha.mlakar, tea.tusar, bogdan.filipic}@ijs.si

ABSTRACT

In surrogate-model-based optimization, the selection of an appropriate surrogate model is very important. If solution approximations returned by a surrogate model are accurate and with narrow confidence intervals, an algorithm using this surrogate model needs less exact solution evaluations to obtain results comparable to an algorithm without surrogate models. In this paper we compare two well known modeling techniques, random forest (RF) and Gaussian process (GP) modeling. The comparison includes the approximation accuracy and confidence in the approximations (expressed as the confidence interval width). The results show that GP outperforms RF and that it is more suitable for use in a surrogate-model-based multiobjective evolutionary algorithm.

1 INTRODUCTION

One of the most effective ways to solve problems with multiple objectives is to use multiobjective evolutionary algorithms (MOEAs). The MOEAs draw inspiration from optimization processes occuring in nature and perform many solution evaluations to find high-quality solutions. Due to the high number of solution evaluations the MOEAs are not very suitable for computationally expensive optimization problems where exact solution evaluation takes a lot of time. In order to obtain the results of such problem more quickly, we usually use surrogate models to approximate the objective functions of the problem.

But due to inaccurate approximations, the solution comparisons can be incorrect, which can result in very good solutions being discarded. In order to minimize the impact of incorrect comparisons, we defined the *relations under uncertainty* ([5]) for comparing approximated solutions presented with an approximated value and a confidence interval. By including the confidence interval in the comparison we were able to consider this additional information and minimize the number of incorrect comparisons.

We used these relations under uncertainty in the algorithm called Differential Evolution for Multiobjective Optimization based on Gaussian Process modeling (GP-DEMO) [4]. We discovered that the quality of the gained result depends greatly on the surrogate model. If the surrogate model is accurate, GP-DEMO finds high-quality results with a low number of exact solution evaluations, while if it is not, GP-DEMO needs more exact solution evaluations to achieve similar results.

Since the accuracy of the surrogate model in surrogate-model-based optimization is crucial, we decided to apply two different modeling techniques and compare their approximations to determine which one is more suitable for use in a surrogate-model-based algorithm. In addition to Gaussian process (GP) modeling that is used in GP-DEMO, we used random forest (RF) for comparison. The reason for choosing RF was the fact that the methodology is well-known and that the solutions approximated with this method in addition to approximated values return also confidence intervals.

The structure of this paper is as follows. In Section 2, we present how the comparison of RF and GP modeling techniques was carried out. In Section 3, we discus the results gained with both techniques, compare them and determine which technique performs better. Section 4 concludes the paper with an overview of the work done.

2 COMPARISON OF RF AND GP SURROGATE MODELS

In this section we compare random forest and Gaussian process modeling techniques used for solution approximations. The aim of the comparison is to determine which of the two techniques is more suitable for use in surrogate-model-based optimization.

To test the two techniques, we used relations under uncertainty to compare their approximated solutions. If two solution approximations had overlapping confidence intervals, we, in order to determine their relation, exactly evaluated one solution and compared the solutions again. Together with the number of these additional exact evaluations, we measured also the number of incorrect solution comparisons and the width of the confidence intervals.

In addition to using relations under uncertainty, we also compared the approximated solutions with Pareto dominance relations and measured the number of incorrect comparisons. With Pareto dominance relations the confidence intervals are not included in the comparisons, so in general, the number of incorrect comparisons hints at the accuracy of the approximations.

Table 1: Comparison of the relations under uncertainty and Pareto dominance relations for GP modeling on the Poloni problem

Relation type	Solutions used for surrogate model	Number of comparisons	Incorrect comparisons	Number of comparisons with confidence interval reductions	Proportion of confidence interval reductions [%]	Confidence interval width
Relations under uncertainty	20	3,940,200	1,515	3,635,805	92	26.25
	30		682	3,152,124	80	15.41
	50		138	1,218,337	31	1.29
	100		65	672,384	17	0.012
	200		13	549,380	14	0.002
Pareto dominance relations	20	3,940,200	367,684	/	/	26.25
	30		159,945	/	/	15.41
	50		22,032	/	/	1.29
	100		2,309	/	/	0.012
	200		1,219	/	/	0.002

Table 2: Comparison of the relations under uncertainty and Pareto dominance relations for GP modeling on the OSY problem

Relation type	Solutions used for surrogate model	Number of comparisons	Incorrect comparisons	Number of comparisons with confidence interval reductions	Proportion of confidence interval reductions [%]	Confidence interval width
Relations under uncertainty	20	3,940,200	74,181	2,289,682	58	42.81
	30		21,861	1,934,212	49	25.98
	50		19,342	1,426,775	36	25.05
	100		144	712,298	18	0.07
	200		152	271,821	7	0.03
Pareto dominance relations	20	3,940,200	336,049	/	/	42.81
	30		136,357	/	/	25.98
	50		49,790	/	/	25.05
	100		1,736	/	/	0.07
	200		1,453	/	/	0.03

Table 3: Comparison of the relations under uncertainty and Pareto dominance relations for the GP modeling on the SRN problem

Relation type	Solutions used for surrogate model	Number of comparisons	Incorrect comparisons	Number of comparisons with confidence interval reductions	Proportion of confidence interval reductions [%]	Confidence interval width
Relations under uncertainty	20	3,940,200	7,407	2,703,783	69	50.03
	30		16	2,338,535	59	0.074
	50		2	749,258	19	0.099
	100		3	359,952	9	0.022
	200		11	183,625	5	0.009
Pareto dominance relations	20	3,940,200	188,401	/	/	50.03
	30		161	/	/	0.074
	50		543	/	/	0.099
	100		645	/	/	0.022
	200		648	/	/	0.009

Table 4: Comparison of the relations under uncertainty and Pareto dominance relations for RF modeling on the Poloni problem

Relation type	Solutions used for surrogate model	Number of comparisons	Incorrect comparisons	Number of comparisons with confidence interval reductions	Proportion of confidence interval reductions [%]	Confidence interval width
Relations under uncertainty	20	3,940,200	22,497	3,906,474	99	32.89
	30		5,206	3,937,230	99	31.83
	50		2,180	3,935,723	99	28.42
	100		125	3,930,277	99	23.97
	200		4	3,909,386	99	19.76
	1,000		2	3,619,402	92	12.11
Pareto dominance relations	20	3,940,200	1,021,750	/	/	32.89
	30		965,491	/	/	31.83
	50		1,043,216	/	/	28.42
	100		894,889	/	/	23.97
	200		733,044	/	/	19.76
	1,000		379,928	/	/	12.11

Table 5: Comparison of the relations under uncertainty and Pareto dominance relations for RF modeling on the OSY problem

Relation type	Solutions used for surrogate model	Number of comparisons	Incorrect comparisons	Number of comparisons with confidence interval reductions	Proportion of confidence interval reductions [%]	Confidence interval width
Relations under uncertainty	20	3,940,200	1	2,663,597	68	842.41
	30		0	2,663,597	68	789.04
	50		0	2,663,597	68	767.92
	100		0	2,663,597	68	720.44
	200		0	2,663,597	68	677.79
	1,000		0	2,663,597	68	548.19
Pareto dominance relations	20	3,940,200	885,416	/	/	842.41
	30		770,439	/	/	789.04
	50		810,251	/	/	767.92
	100		683,578	/	/	720.44
	200		661,919	/	/	677.79
	1,000		555,983	/	/	548.19

Table 6: Comparison of the relations under uncertainty and Pareto dominance relations for the RF modeling on the SRN problem

Relation type	Solutions used for surrogate model	Number of comparisons	Incorrect comparisons	Number of comparisons with confidence interval reductions	Proportion of confidence interval reductions [%]	Confidence interval width
Relations under uncertainty	20	3,940,200	18	3,384,351	86	359.51
	30		0	3,385,285	86	350.55
	50		0	3,385,242	86	308.94
	100		0	3,384,910	86	266.55
	200		0	3,378,456	86	224.77
	1,000		0	3,133,626	79	139.89
Pareto dominance relations	20	3,940,200	387,854	/	/	359.51
	30		425,691	/	/	350.55
	50		365,606	/	/	308.94
	100		288,611	/	/	266.55
	200		216,634	/	/	224.77
	1,000		136,656	/	/	139.89

The solutions selected for testing were not generated randomly, but rather produced by the well-known NSGA-II algorithm [3]. This ensured that the solution comparisons were similar to the comparisons performed in evolutionary multiobjective algorithms and thus provided relevant results.

In every generation NSGA-II creates a new set of solutions, adds them to the current ones and then performs selection on the union to identify the most promising ones. The selection procedure includes comparing every solution with all other solutions to determine its dominance status. These were the comparisons used in our study.

The experiments were performed on three benchmark multiobjective optimization problems. One is the Poloni optimization problem [6] and two are from [2], called OSY and SRN. All of them are two-objective problems.

For testing purposes we used GP modeling as proposed by [7] and RF modeling as proposed in [1]. For the confidence interval width of the approximation we used two standard deviations (2σ), which corresponds to about 95% of the normal distribution of the approximations. The number of trees used for building RF was 10,000 and the minimum number of elements in the leaves was set to 1.

To test the correlation between the surrogate model accuracy and the incorrect comparisons, different models of increasing accuracy were built—each on larger number of solutions. Since building an RF surrogate model is faster than building a GP surrogate model, we, in addition to building surrogate models from 20, 30, 50, 100 and 200 exactly evaluated solutions, also built an RF surrogate model from 1000 exactly evaluated solutions. We tested how much the larger RF surrogate model built from 1000 exactly evaluated solutions increases the accuracy of the approximations.

The NSGA-II parameter values used in the experiments were the same for both modeling techniques and for all three problems. They were set as follows:

- population size: 100,
- number of generations: 100,
- number of runs: 30.

The results averaged over 30 runs are presented in Tables 1–3 (for GP modeling) and in Tables 4–6 (for RF modeling).

3 DISCUSSION

The results gained with both modeling techniques show that, irrespectively of the accuracy of a surrogate model, using relations under uncertainty reduces the number of incorrect comparisons.

The comparison of the results gained with RF and GP reveals certain differences between the techniques. The main difference is in the width of the confidence intervals. RF surrogate models produce wider confidence intervals. Consequently, the number of comparisons with confidence interval reductions for RF is much higher than for GP.

In addition to yielding wider confidence intervals, the RF surrogate models are also less accurate. Comparing the number of incorrect comparisons performed with Pareto dominance relations where the confidence intervals are not considered, we can see that the number of incorrect comparisons is higher with the RF surrogate models.

Another difference is in the correlation between the number of solutions used for building the surrogate model and the accuracy of the surrogate model. By increasing the number of solutions used, the RF surrogate models do not improve as quickly as the GP models. Even in the cases where 1000 exactly evaluated solutions were used for building the RF surrogate models the confidence interval widths were not greatly reduced and the intervals were still much wider than the confidence intervals gained with GP models built from 200 solutions.

Looking at the number of incorrect comparisons, we can see that by using relations under uncertainty with RF the results are slightly better than with GP. The reason for that is in the fact, that the approximations with RF have relatively wide confidence intervals which results in high number of confidence interval reductions. Therefore, most solutions have to be exactly evaluated in order to perform the comparisons. So the reason for a lower number of incorrect comparisons is not the higher quality of the surrogate models, but in the fact that more solutions are exactly evaluated and are therefore without uncertainty. Since in surrogate-model-based optimization exactly evaluated solutions are typically computationally expensive, a modeling technique that exactly evaluates most of the solutions is not very useful.

4 CONCLUSION

In this paper we compared random forest and Gaussian process modeling techniques in the context of surrogate-model-based multiobjective optimization. We compared their approximation accuracy and width of the confidence intervals.

The results show that surrogate models built with GP modeling produce more accurate approximations with narrower confidence intervals. Due to narrower confidence intervals the comparisons of solutions approximated with GP modeling require less additional exact solution evaluations. As a result, we can conclude that GP modeling is more appropriate for use in a surrogate-model-based algorithm than RF.

References

[1] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[2] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, New York, 2001.

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[4] M. Mlakar, D. Petelin, T. Tušar, and B. Filipič. GP-DEMO: Differential evolution for multiobjective optimization based on Gaussian process models. *European Journal of Operational Research*, 2014, doi: 10.1016/j.ejor.2014.04.011.

[5] M. Mlakar, T. Tušar, and B. Filipič. Comparing solutions under uncertainty in multiobjective optimization. *Mathematical Problems in Engineering*, 2014, doi: 10.1155/2014/817964.

[6] C. Poloni, A. Giurgevich, L. Onesti, and V. Pediroda. Hybridization of a multi-objective genetic algorithm, a neural network and a classical optimizer for a complex design problem in fluid dynamics. *Computer Methods in Applied Mechanics and Engineering*, 186(2):403–420, 2000.

[7] C. E. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.

COMPREHENSIBILITY OF CLASSIFICATION TREES – SURVEY DESIGN

Rok Piltaver[1,2], *Mitja Luštrek*[2], *Matjaž Gams*[1,2], *Sanda Martinčić – Ipšić*[3]
Jožef Stefan Institute - Department of Intelligent Systems, Ljubljana, Slovenia [1]
Jožef Stefan International Postgraduate School, Ljubljana, Slovenia [2]
University of Rijeka - Department of Informatics, Rijeka, Croatia [3]
rok.piltaver@ijs.si, mitja.lustrek@ijs.si, matjaz.gams@ijs.si, smarti@inf.uniri.hr

ABSTRACT

Comprehensibility is the decisive factor for application of classifiers in practice. However, most algorithms that learn comprehensible classifiers use classification model size as a metric that guides the search in the space of all possible classifiers instead of comprehensibility - which is ill-defined. Several surveys have shown that such simple complexity metrics do not correspond well to the comprehensibility of classification trees. This paper therefore suggests a classification tree comprehensibility survey in order to derive an exhaustive comprehensibility metrics better reflecting the human sense of classifier comprehensibility and obtain new insights about comprehensibility of classification trees.

1 INTORDUCTION

Comprehensibility of data mining models, also termed interpretability [15] or understandability [1], is the ability to understand the output of induction algorithm [14]. Its importance has been stressed since the early days of machine learning research [17, 19]. Kodratoff even reports that it is the decisive factor when machine learning approaches are applied in industry [13]. Application domains in which comprehensibility is emphasized are for instance medicine, credit scoring, churn prediction, bioinformatics, and others [8].

A metric of comprehensibility is therefore needed in order to compare learning systems performance and as a (part of) heuristic function used by a learning algorithm [9, 21]. Majority of algorithms for learning comprehensible models use simple measures based on model size which may oversimplify the learned models. Humans by nature are mentally opposed to too simplistic representations of complex relations [7], therefore it is no surprise that empirical studies have shown comprehensibility to be negatively correlated with the complexity (size) of a classifier in at least some cases [1]. Such simple measures based on model complexity are therefore regarded as an over-simplistic notion of comprehensibility [8].

Those facts motivated us to propose a survey design, with the goal to derive an exhaustive comprehensibility metrics better reflecting the human sense of classifier comprehensibility. Obtained insights into evaluator's judgments about classifier comprehensibility will provide means for inducing definition of comprehensibility metrics that capture fine-grained differences in classifier comprehensibility and for evaluating the induced metrics. User survey based approach, which follows the observation that comprehensibility is in the eye of the beholder [16], is advocated; defining comprehensibility metric directly is not possible because it is comprehensibility is ill-defined [13].

2 REVIEW OF RELATED WORK

According to [16] comprehensibility measures the "mental fit" [15] of the classification model, which has two main drivers: the type of classification model and its size or complexity. It is generally accepted that tree and rule based models are the most comprehensible while SVM, ANN and ensembles are in general black box models that can be hardly interpreted by users [8, 16, 20]; however there are domain and user specific exceptions from this rule of thumb. For a given classification model, the comprehensibility generally decreases with the size [2]. This principle is motivated by Occam's razor, which prefers simpler models over more complex ones [6]. Furthermore, a rule based model with few long clauses is harder to understand than one with shorter clauses, even if the models are of the same absolute size [20]. Comprehensibility also decreases with increasing number of variables and constants in a rule [20] and amount of inconsistency with existing domain knowledge [1, 18].

User-oriented assessment of classifier comprehensibility [1] compared outputs of several tree and rule learning algorithms and concluded that trees are more comprehensible then rules, and that in some cases tree size is negatively correlated with comprehensibility. Note that the trees included in the study were simple and were probably perceived as less comprehensible because they did not agree with the users' knowledge. Another study [12] (based on inexperienced users) compared comprehensibility of decision tables, trees and rules. The results showed that the respondents were able to answer the questions faster, more accurately and more confidently using decision tables than using rules or trees and were clearly able to assess the difficulty of the questions. Larger classifiers resulted in a decrease in answer accuracy, an increase in answer time, and a decrease in confidence in answers. Evidence that answering logical questions (e.g. validate a classifier) is considerably more difficult than classifying a new instance was found. However, proposition that cognitive fit of classifier with the given task type influences users' performance received limited support. A paper on comprehensibility of classification trees, rules, and tables, nearest neighbor and Bayesian network classifiers [8] stressed that graphical representation, hierarchical structure, including only subset of attributes in a tree, and independence of tree branches are advantages of classification trees. On the other hand, possible irrelevant attributes and replicated subtrees enforced by the tree structure decrease comprehensibility and may lead to overfitting. This can be mitigated by converting a tree into a rule set, which enables more flexible pruning resulting in a more comprehensible representation. Another recognized downside of classification trees is their Boolean logic-based nature as opposed to the probabilistic interpretation of naïve Bayes, which might be preferred in some applications [8].

This paper focuses on the comprehensibility of classification trees; however most of the suggested ideas could be analogically implemented on classification rules and tables as well. The survey design enables analysis of the influence of tree complexity and visualization on its comprehensibility. The complexity of classification tree is usually measured with the number of leaves or nodes in a tree or the number of nodes per branch [16, 20] while the suggested survey considers some additional complexity measures as well. The influence of visualization on comprehensibility has been stressed [16] but empirical studies are missing, therefore the suggested survey also considers visualization factors. The past empirical studies of classifier comprehensibility [1, 12] were performed only on homogenous groups of students, therefore we suggest adding data mining experts with different cultural background to the group of participants in future studies.

3 SURVEY DESIGN

One possible way to estimate comprehensibility of a classifier is to present it to a survey respondent, who will analyze it, and then conduct an interview about comprehensibility. This approach is very time consuming and may be unintentionally biased by both involved persons, e.g. asking a question about comprehensibility of a model may help the respondent in comprehending the classifier. Therefore the indirect and more objective approach that was also used in previous studies [1, 12] is preferred. It measures the performance of respondents asked to solve tasks that involve interpretation and understanding of classifiers. The following subsections of the paper define the selected survey tasks, performance metrics, observed properties of classifiers, and strategies that prevent bias.

3.1 Survey tasks (question types)

The comprehensibility survey consists of six tasks. The first task - **classify** asks respondent to classify an instance according to a given classifier (same as in [1, 12]). Tasks 2-4 are based on [4], which reports that comprehensibility is required to explain individual instance classifications, validate the classifier, and discover new knowledge. Thus the second task - **explain** ask the respondent to answer which attributes values must be changed or retained in order to classify a given instance into another class. For example, which habits (values of attributes) would a patient with high probability of getting cancer (class) have to change in order to stay healthy? The third task - **validate** requires the respondent to check whether a statement about the domain is confirmed or rejected according to the presented classifier. For example: does the tree say that persons smoking more than 15 cigarettes per day are likely to get cancer. Similar questions were also asked in [12]. The fourth task - **discover** asks the respondent to find a property (attribute-value pair) that is unusual for instances from one class; this corresponds to finding a property of outliers. For example, people that lead healthy life are not likely to get cancer, except if they have already suffered from it in the past.

The fifth task - **rate** requests the user to give the subjective opinion about the classification trees on a scale with five levels: very easy to comprehend, easy to comprehend, comprehensible, difficult to comprehend, and very difficult to comprehend. Each label of the scale is accompanied with an explanation that relates to the time needed to comprehend the tree and difficulty of remembering it and explaining it to another person. The purpose of explanations is to prevent variation in subjective interpretations of the scale. The task intentionally follows the first four tasks in which the respondents use the classifiers and obtain hands on experience, which enables them to rank the comprehensibility. The classifiers are learned on a single dataset and visualized using Orange tool [5] in order to be consistent across all the tasks and enable reliable and prompt responding. For the same reason meaningful attribute and class names are used. The first five tasks measure the influence of classifier complexity (i.e. the number of leaves, depth, branching) while the final task measures the influence of different representations of the same tree on the comprehensibility.

Task six - **compare** asks the respondents to rate which of the two classification trees shown side by side is more comprehensible on the scale with three levels: the tree is much more comprehensible, the tree is more comprehensible, and the trees are equally comprehensible. One of the trees in this task is already used in the previous five tasks - serving as a known frame of reference - while the other one is a previously unseen tree with the same content but represented in different style. The position of a tree (left or right) is randomized in order to prevent bias, e.g. assuming that the left tree is always more comprehensible.

3.2 Performance metrics

The tasks rate and compare are directed toward obtaining subjective opinions rated on the given scales. The tasks classify, explain, validate, and discover are directed toward objectively quantifying respondents' performance (e.g. time and correctness of answers). Corresponding performance metrics are derived from the six metrics proposed in the experiments on conceptual model understandability [11]. The first three are explicitly measured by the survey: the time needed to understand a model translates to time to answer a question (longer time - less comprehensible classifier); correctly answering questions about the content translates to the probability of correct answer (higher probability - more comprehensible classifier); the perceived ease of understanding is expressed with subjective judgment of a questions difficulty (rated on scale very easy, easy, medium, difficult and very difficult). The other measures are implicitly embedded in the survey design: difficulty of recalling a model is captured through descriptions of the five levels of comprehensibility scale in the rate task; problem-solving based on the model content is embedded in tasks 1-4; and verification of model content is in the validate task.

3.3 Observed classifier properties

Motivated by the related work [1, 8, 12, 20] and authors' experience the following **tree complexity properties** are proposed: number of leaves or nodes, branching factor, number of nodes in a branch, and number of instances belonging to a leaf. Proposed tree complexity properties are systematically varied in the first five tasks of the survey. Also, the proposed **tree visualization properties** are varied in the compare task: using color to enhance readability (e.g. pie-charts corresponding to class distributions in nodes), layout of the tree based on the depth of subtrees, and general layout and readability of the visualized tree (e.g. plain text output vs. default Weka [10] and Orange [5] visualization). Additionally, the survey enables **contrasting**: meaningful names of attributes, attribute values and classes to meaningless ones; attributes with high information gain to the ones with low gain; and meaningful aggregated attributes contrasted to conjunctions of isolated attribute-value pairs (i.e. deep structure [20]). Finally, the survey design also enables various statistical analysis for the each single leaf (branch of the tree) or for the entire tree.

3.4 Avoiding implicit survey bias

In order to prevent bias the following issues must be considered: choice of the classification domain, classifiers, and respondents group, and the ordering of questions. The **classification domain** has to be familiar to respondents - all of them are aware of relations among attribute values and classes and none of them have significant advantage of more in-depth knowledge about the domain. At the same time, the domain must be broad and rich enough to enable learning a range of classifiers with various properties listed in 3.3. Furthermore, choosing an interesting domain motivates the respondents to participate in the survey. The Zoo domain from the UCI Machine Learning Repository [3] meets all the requirements and is highly appropriate for general and heterogeneous population. It requires only elementary knowledge about animals expressed with 17 (mostly binary) attributes: are they aquatic or airborne, do they breathe, how many legs they have, do they have teeth, fins or feathers, etc. The Zoo domain induces 7 classes: mammals, fish, birds, amphibian, reptile, mollusk, and insect.

The selected **classifiers** must vary in complexity but not in other parameters that may influence comprehensibility and hence bias the results. In addition, classifiers are learned using well-known machine learning algorithm rather than manually constructed. Using different pruning parameters produces trees with different sizes. Higher branching factor can be achieved by replacing original binary attributes with constructed attributes, which can be interpreted as building deep models [20]. If possible, order of the leaves or at least their grouping in subtrees should remain the same as in the binary tree. Choosing a question for a given tree determines the number of nodes in a branch that the user will have to analyze in order to answer. In each group of questions a single parameter changes while the others remain constant. Finally, a well-known and comprehensible classifier visualization style must be used, e.g. Orange [5].

Order of the question may also induce bias. For example, the learning effect can occur: the respondents need more time to answer the first few questions, after that they answer quicker. Next, the performance of respondents drops if they get tired or loose motivation, therefore the number of questions must be limited. To prevent those effects, Latin square ordering is used, where each question occurs exactly once at each place in the ordering and subsequently each respondent gets a different ordering of the questions. Finally, starting each task with a test question (from the different domain) reduces the learning effect as well.

The survey design **assumes the following order of tasks**: starts with the simpler and progresses to more difficult ones. The compare and rate tasks, related to subjective opinions, are placed toward the end – after the respondents acquire experience with the classifiers.

Demographic data (DM knowledge, age, sex, language) reflects the heterogeneity of the respondents group and enables detailed analysis of classifier comprehensibility per different subgroups like students or experts. Hence, the **test group** consists of data mining experts on one hand and non-experts with basic knowledge about classification on the other. Comparing the results of the two groups as well as considering the cultural background (e.g. different mother tongues), can provide new insights into classifier comprehensibility. Finally, obtaining statistically significant results requires high enough number of respondents.

4 SURVEY IMPLEMENTATION

This work proposes **online survey** in order to facilitate accurate measurements of time, automatic checking the correctness of answers, saving the answers in a database and allowing remote participation. Several tools for designing and performing online surveys exist but do not meet all of the design requirements (see section 3): Latin square design, measuring the time of answering each question, automatic translation to several languages, using templates to quickly define questions for a given task and automatically checking the correctness of answers. Therefore, custom online survey is implemented using MySQL database, PHP and JavaScript programming languages, and CSS for webpage formatting.

The **database** includes one table for demographic data with auto-increment user id as the primary key and one table per task with user id and question id as the primary key. Each task table includes a field representing question order number, a date-time field, and field(s) representing the respondents' answer. Tables for tasks 1-4 additionally include fields with the measured answering time, list of all respondent clicks and associated times, and the indicator of correct answer. PHP is used to dynamically **generate survey webpages** with correct ordering of questions for each respondent and storing the answers into the database. Question webpages are generated by a separate PHP script for each task based on a template and a simple data structure defining the questions. An additional PHP script is used as a library of shared functions and data structures: one represents instances used in the survey and the other terms (instructions, attribute names and value, classes, etc.) translated into English, Slovenian and Croatian languages. Additionally, PHP scripts are used for backing-up and checking correctness of answers, login and help pages, and a respondent home-page providing feedback on personal progress and performance compared to the group. **SVG images** representing the classification trees exported from Orange [5] were automatically translated into the three languages using a Java program – the translation table is the same as in the PHP library script.

JavaScript is used to **measure the time of answering** each question. When a webpage is opened, only the instructions and footer of the page are visible. Clicking on the button "*Start solving"* calls a JavaScript function that displays the question (e.g. table with attribute-value pairs and image of a tree) and the answer form (drop-down lists, radio buttons) and starts the timer. Changing a value of the answer form field records the relative time and action type. When the respondent clicks the "*Finish button"*, the answer fields are disabled, time is calculated, and question difficulty rating options are displayed. When the "*Next button"* is clicked, the collected values are assigned to hidden form fields in order to pass them to the PHP script that stores the data in the database and displays the next question.

A psychologist and two DM experts analyzed the initial survey and improved version was implemented based on their comments. It passed a **validation** test with 15 students answering the first task at the same time. Preliminary analysis of the results for 10 respondents is in line with the expectations, thus the survey is ready to be used in order to collect data about tree comprehensibility.

References:

[1] H. Allahyari, N. Lavesson, User-oriented Assessment of Classification Model Understandability, *11th Scandinavian Conf. on AI*, pp. 11-19, 2011.

[2] I. Askira-Gelman, Knowledge discovery: comprehensibility of the results. *Proceedings of the 31st Annual Hawaii Int. Conf. on System Sciences, 5*, pp. 247, 1998.

[3] K. Bache, M. Lichman. UCI Machine Learning Repository, http://archive.ics.uci.edu/ml. University of California, School of Inf. and Comp. Science, 2014.

[4] M. W. Craven, J. W. Shavlik. Extracting Comprehensible Concept Representations from Trained Neural Networks. *Working Notes on the IJCAI'95 WS on Comprehensibility in ML*, pp. 61-75, 1995.

[5] J. Demšar, T. Curk, A. Erjavec. Orange: Data Mining Toolbox in Python. *Journal of Machine Learning Research, 14 (Aug)*, pp. 2349−2353, 2013.

[6] P. Domingos. The role of occam's razor in knowledge discovery, *Data Mining and Knowledge Discovery, 3*, pp. 409–425, 1999.

[7] T. Elomaa. In Defense of C4.5: Notes on learning one-level decision trees. *Proceedings of 11th Int. Conf. on ML*, pp. 62-69, 1994.

[8] A. A. Freitas. Comprehensible classification models - a position paper. *ACM SIGKDD Explorations, 15 (1)*, pp. 1-10, 2013.

[9] C. Giraud-Carrier. Beyond predictive accuracy: what? *Proceedings of the ECML-98 Workshop on Upgrading Learning to Meta-Level: Model Selection and Data Transformation*, pp. 78-85, 1998.

[10] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten. The WEKA Data Mining Software: An Update. *SIGKDD Explorations, 11 (1),* 2009.

[11] C. Houy, P. Fettke, P. Loos. Understanding Understandability of Conceptual Models – What Are We Actually Talking about? *Conceptual Modeling - Lecture Notes in Comp. Sc. vol. 7532*, pp. 64-77, 2012.

[12] J. Huysmans, K. Dejaeger, C. Mues, J. Vanthienen, B. Baesens. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems, 51 (1)*, pp. 141-154, 2011.

[13] Y. Kodratoff, The comprehensibility manifesto, KDD Nuggets (94:9), 1994.

[14] R. Kohavi. Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid. *Proceedings of the 2nd Int. Conf. on KD and DM*, pp. 202-207, 1996.

[15] O. O. Maimon, L. Rokach, Decomposition Methodology for Knowledge Discovery and Data Mining: Theory and Applications, World Scientific Publishing Company, 2005.

[16] D. Martens, J. Vanthienen, W. Verbeke, B. Baesens. Performance of classification models from a user perspective. *Decision Support Systems, 51 (4)*, pp. 782-793, 2011.

[17] R. Michalski, A theory and methodology of inductive learning, *Artificial Intelligence 20*, pp. 111–161, 1983.

[18] M. Pazzani. Influence of prior knowledge on concept acquisition: experimental and computational results. *Journal of Experimental Psychology. Learning, Memory, and Cognition 17*, pp. 416–432, 1991.

[19] Quinlan, J.R. Some elements of machine learning. *Proc. 16th Int. Conf. on Machine Learning (ICML-99)*, pp. 523-525, 1999.

[20] E. Sommer. An approach to quantifying the quality of induced theories. *Proceedings of the IJCAI Workshop on Machine Learning and Comprehensibility*, 1995.

[21] Z.-H. Zhou. Comprehensibility of data mining algorithms. *Encyclopedia of Data Warehousing and Mining*, pp. 190-195, Hershey, 2005.

PAMETNO VODENJE SISTEMOV V STAVBAH S STROJNIM UČENJEM IN VEČKRITERIJSKO OPTIMIZACIJO

Rok Piltaver, Tea Tušar, Aleš Tavčar, Nejc Ambrožič, Tomaž Šef, Matjaž Gams, Bogdan Filipič
Institut "Jožef Stefan", Odsek za inteligentne sisteme
Jamova cesta 39, 1000 Ljubljana, Slovenija
e-mail: {rok.piltaver, tea.tusar, ales.tavcar, tomaz.sef, matjaz.gams, bogdan.filipic}@ijs.si

ABSTRACT

Prispevek opisuje programsko opremo za pametno in celovito vodenje sistemov v stavbi, kot so ogrevanje, prezračevanje, senčenje, razsvetljava in upravljanje z viri energije. Cilj je zagotoviti čim nižje stroške in hkrati čim višje udobje za stanovalce. Sistem pametne stavbe pridobi podatke s senzorjev, nameščenih v stavbi, in se iz njih nauči navad in akcij uporabnikov v preteklem obdobju. V drugem koraku uporabi večkriterijsko optimizacijo, ki na podlagi simulacij išče najboljše nastavitve parametrov za vodenje sistemov v stavbi. Uporabniku se najboljše nastavitve parametrov prikažejo v oblik urnikov. Za vsak urnik sta dana dva podatka, udobje in cena, na podlagi katerih uporabnik izbere najprimernejši urnik in s tem na preprost način nastavi parametre za avtomatizacijo sistemov v stavbi, ki zagotovijo želeni kompromis med udobjem in stroški.

1 UVOD

Bivalni objekti v Evropi so leta 2004 porabili 37% vse porabljene energije [3], v Združenih državah Amerike pa je bil v letu 2010 ta delež kar 41% [2]. Iskanje strategij za zmanjšanje porabe energije je torej ena izmed ključnih nalog sodobne družbe in tema številnih raziskav, ki se ukvarjajo z razvojem učinkovitih metod vodenja naprav, ki porabijo veliko energije. Sistemi za ogrevanje, hlajenje in prezračevanje prostorov npr. porabijo 50% vse energije, ki jo stanovanjske hiše potrebujejo za obratovanje [3]. Dobre strategije morajo ustrezno obravnavati nasprotujoče si zahteve uporabnikov, kot sta npr. sočasno doseganje energetske varčnosti in visoke stopnje ugodja.

Zmanjševanje stroškov shranjevanja in obdelave podatkov, dostopnost senzorjev in aktuatorjev ter enostavno povezovanje različnih naprav v skupen sistem omogočajo uporabo kompleksnih metod vodenja tudi v manjših bivalnih enotah. Obstoječi sistemi pametnih hiš sicer omogočajo avtomatizacijo delovanja sistemov v stavbah po vnaprej nastavljenih urnikih, preklapljanje med načini delovanja glede na zaznano prisotnost uporabnikov ali na zahtevo uporabnika preko spletnega vmesnika. Vendar večini uporabnikov ne uspe nastaviti primernega urnika za avtomatizacijo, saj morajo pri tem nastaviti veliko pogosto nerazumljivih parametrov in upoštevati nenehne spremembe svojih potreb in zunanjih vplivov, kot so vreme in cene energentov. Poleg tega take rešitve ne izrabijo celotnega potenciala, ki jih sistemi hišne avtomatizacije omogočajo. Zato v [4] predlagajo uporabo tehnik strojnega učenja za prepoznavanje navad uporabnikov in gradnjo napovednih modelov njihovega obnašanja ter uporabo večkriterijske optimizacije za zagotavljanje ustreznega upravljanju inteligentnega doma, ki zadovoljuje nasprotujoče si kriterije.

Pričujoči prispevek v 2. razdelku opisuje delovanje sistema OpUS, ki implementira predlagane rešitve za pametno vodenje sistemov v stavbah na podlagi učenja in večkriterijkse optimizacije. Rezultati delovanja sistema OpUS so predstavljeni na primeru uporabe v 3. razdelku Prispevek se zaključi z razpravo v 4. razdelku.

2 SISTEM OPUS

Programska oprema sistema OpUS, prikazana na sliki 1, je razdeljena v štiri sklope: beli kvadrati predstavljajo vhodno/izhodne module, modra kvadrata ustrezata moduloma za učenje, zelena modulu za optimizacijo in oranžna moduloma za simulacijo. Številke predstavljajo zaporedje toka podatkov skozi sistem od vhodnih senzorskih podatkov (1) do parametrov za avtomatizacijo sistemov v stavbi (10). Vsebina podatkovnih tokov in delovanje posameznih modulov sta opisana v nadaljevanju.

2.1 Pridobivanje senzorskih podatkov

Obstoječi sistemi za hišno avtomatizacijo ponujajo široko paleto senzorjev: od senzorjev gibanja, temperature, vlažnosti in kakovosti zraka, osvetljenosti, pretoka vode in porabe električne energije do pametnih stikal in podatkov o delovanju posameznih naprav. Poleg tega omogočajo tudi zbiranje, shranjevanje in posredovanje senzorskih podatkov zunanjim sistemom (slika 1, točka 1). Sistem OpUS uporablja modul za pridobivanje senzorskih podatkov, ki mora biti prilagojen protokolu komunikacij in formatu podatkov, ki ga podpira sistem hišne avtomatizacije – to omogoča prilagoditev sistema OpUS različnim sistemom hišne avtomatizacije. Pridobljeni senzorski podatki se pretvorijo v poenoteno obliko, ki ob vsaki spremembi shrani čas, tip senzorja (določa mersko enoto, natančnost in frekvenco meritev ipd.) in identifikacijo senzorja (določa lokacijo senzorja in povezavo z zabeleženimi preteklimi vrednostmi) ter novo vrednost. Poenoteni podatki se shranijo v podatkovno bazo za kasnejše analize in prikaz uporabniku ter se na zahtevo posredujejo moduloma za učenje (slika 1, točka 2).

2.2 Modula za učenje

Pretekle raziskave so pokazale, da lahko z uporabo podatkov o prisotnosti uporabnikov in njihovih akcijah napovemo prisotnost ali odsotnost uporabnikov ter njihove navade z relativno visoko točnostjo [1]. Na tej osnovi sta bila razvita modula za učenje navad in akcij, opisana v nadaljevanju.

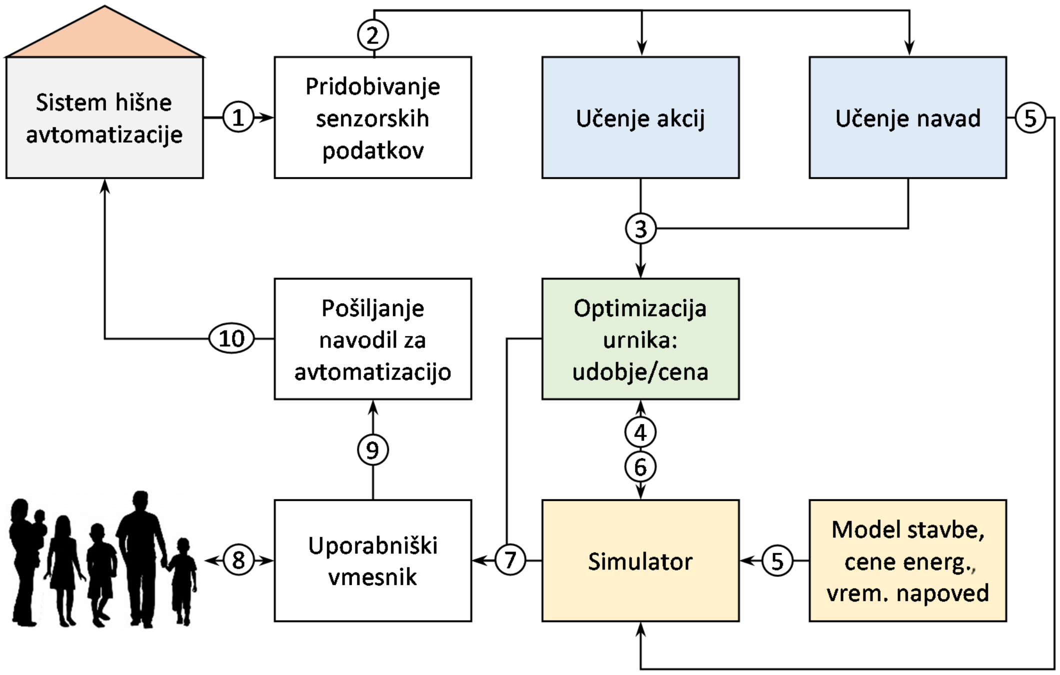
Modul za učenje navad periodično zahteva časovno okno podatkov, iz katerih prepozna prisotnost in odsotnost uporabnikov ter njihove aktivnosti: spanje, pripravo obroka in prehranjevanje, uporabo kopalnice ipd. Prisotnost uporabnika v določenem prostoru prepozna neposredno iz podatkov o uporabi stikal v prostoru in zaznavah senzorjev gibanja, ostale aktivnosti pa s pomočjo zlivanja senzorskih podatkov in uporabo konteksta: čas, prostor in predhodne aktivnosti. Npr. prižgana luč v kopalnici in 7 minut pretoka tople vode sovpadata z aktivnostjo uporabe kopalnice; ugasnjene ali zatemnjene luči, odsotnost gibanja ter drugih akcij uporabnikov ob podatku, da oseba ni zapustila stavbe ter da je ura 4 zjutraj, sovpadajo z aktivnostjo spanje. Prepoznavanje aktivnosti je pomembno, ker določa okoljske parametre, ki so ob določeni aktivnosti za uporabnika udobni: v času aktivne prisotnosti mora biti temperatura v stavbi primerna, zrak svež in ne presuh ali preveč vlažen; v času spanja so lahko temperatura, osvetljenost in zaloga tople vode nižje; v času odsotnosti temperatura in osvetljenost nista pomembni, okna pa morajo biti zaprta. Samodejno učenje spreminjajočih se uporabnikovih navad odpravi potrebo po ročnem (po)nastavljanju urnikov za avtomatizacijo sistemov v stavbi ter hkrati omogoči boljše nastavitve, ki temeljijo na natančnih statističnih podatkih o pretekli uporabi.

Modul za učenje akcij periodično zahteva podatke o akcijah uporabnika, ki jih le-ta izvede, kadar okolje v stavbi zanj ni dovolj udobno: npr. previsoka temperatura, slaba osvetljenost ali kakovost zraka. Zbrane podatke modul analizira v kontekstu časa, prostora in prepoznane aktivnosti. Če zazna, da uporabnik pri določeni aktivnosti v določenem prostoru večkrat izvede enako akcijo, iz tega sklepa, da je nastavitev scene (opisana v [5]) za to aktivnost in prostor neprimerna ter predlaga njeno spremembo.

Oba modula poleg specifičnih metod za prepoznavanje akcij in aktivnosti uporabljata standardne algoritme strojnega učenja, da zgradita model, ki uporabniku zagotavljajo udobje. Model vsebuje podatke o tem, kakšna je verjetnost, da uporabnik na določen dan v tednu ob določenem času potrebuje neko sceno (povezano z aktivnostjo uporabnika), in kake vrednosti parametrov naj bodo nastavljene za posamezno sceno (temperatura zraka, osvetljenost, zaprta okna idr.) [5]. Model se posreduje modulu za optimizacijo, kot prikazuje slika 1, korak 3.

2.3 Modul za optimizacijo

Cilj optimizacije je poiskati nedominirane (t.j. najboljše) urnike po kriterijih udobja in cene – namesto te se lahko uporablja tudi količina porabljenih energentov ali količina posledično izpuščenega CO_2. Ker sta si kriterija udobje in cena nasprotujoča, je izhod postopka optimizacije množica urnikov, ki so med sabo neprimerljivi (boljši v enem kriteriju in slabši v drugem) in boljši od vseh ostalih urnikov. Urnik je predstavljen kot zaporedje 15-minutnih časovnih intervalov za katere je treba določiti parametre vodenja posameznih sistemov v stavbi. Za iskanje nedominiranih urnikov se uporablja algoritem večkriterijske optimizacije, ki podatka o ceni in udobju urnika pridobi od modula za simulacijo (slika 1, korak 6).


Sistem hišne avtomatizacije
1
Pridobivanje senzorskih podatkov
2
Učenje akcij
Učenje navad
5
3
Optimizacija urnika: udobje/cena
10
Pošiljanje navodil za avtomatizacijo
4
6
9
8
Uporabniški vmesnik
7
Simulator
5
Model stavbe, cene energ., vrem. napoved

Slika 1: Programski moduli sistema OpUS in podatkovni tokovi med njimi

2.4 Simulacija

Modul za simulacijo na vhodu sprejme (slika 1, korak 5) model stavbe, ki npr. določa toplotne izgube, porabo energije posameznih sistemov ipd.; cene energentov, ki omogočijo izračun stroškov določenega urnika; vremensko napoved, ki določa pričakovane zunanje vplive na pogoje v stavbi; in model uporabnikovih navad, ki je osnova za izračun udobja danega urnika. Simulator je osnovan na obstoječih splošnih simulatorjih delovanja sistemov v stavbah, lastnostih stavbe in konkretnih sistemov, prisotnih v stavbi ter metodi za izračun neudobja uporabnika glede na okoljske pogoje v stavbi in želene pogoje. Rezultati simulacije, ki so odvisni od točnosti simulatorja in vhodnih podatkov, se vrnejo modulu za optimizacijo (slika 1, korak 6), ki na njihovi podlagi predlaga nove urnike (slika 1, korak 4). Po končani optimizaciji se izbrani urnik in pripadajoče udobje ter cena posredujejo uporabniškem vmesniku (slika 1, korak 7).

2.5 Uporabniški vmesnik

Uporabniški vmesnik ponuja vizualizacijo, iz katere sta razvidna cena in udobje najboljših urnikov. Uporabniku se ob izbiri posameznega ponujenega urnika prikažejo razlike v trajanju in nastavitvah scen ter udobju in ceni med izbranim in trenutno nastavljenim urnikom. Množica najboljših (nedominiranih) rešitev omogoča, da uporabnik dobi vso informacijo o delovanju sistema in se na podlagi te informacije odloči, kateri kriterijje zanj pomembnejši ter kakšen kompromis med kriterijema mu bolj ustreza. Uporabnik lahko v koraku 8 (slika 1) izbere enega od predlaganih urnikov ter ga po potrebi prilagodi svojim željam – v tem primeru se ponovno izvede optimizacija izvajanja urnika in simulacija za oceno cene in udobja predlaganih sprememb urnika.

Sistem OpUS začne delovati z vnaprej nastavljenim urnikom, ki je dober približek splošno uporabnega urnika. Skozi čas se sistem nauči navad in potreb uporabnika ter predlaga boljše urnike. Izboljšan urnik je primeren za uporabo, dokler ne pride do sprememb navad uporabnikov ali do spremembe zunanjih vplivov: vremena kot posledice letnih časov ali bistvene spremembe cen energentov na trgu. Poleg izbire in primerjave urnikov uporabniški vmesnik ponuja tudi pregled nad preteklo porabo in skladnostjo izbranega urnika s prepoznanimi potrebami uporabnika ter ročno upravljanje s sistemom za hišno avtomatizacijo.

2.6 Vodenje sistemov v stavbi

Izbrani urnik in pripadajoči parametri za vodenje sistemov v stavbi se iz uporabniškega vmesnika pošljejo modulu za pošiljanje navodil za avtomatizacijo (slika 1, korak 9). Le-ta je izhodni modul, ki mora biti prilagojen konkretnemu sistemu hišne avtomatizacije podobno kot modul za pridobivanje senzorskih podatkov. Poleg določenega urnika modul sprejme tudi podatke o zaznanih aktivnostih uporabnikov, ki jih prepozna modul za učenje navad, na podlagi katerih preklaplja med scenami, kadar se pričakovana aktivnost na urniku ne ujema z zaznano aktivnostjo (slika 1, korak 10).

3 PRIMER UPORABE

V tem razdelku predstavljamo rezultate pametnega vodenja na primeru stavbe s fotovoltaičnimi paneli, kjer lahko določamo polnjenje in praznjenje baterije ter delovanje nekaterih porabnikov, medtem ko so scene omejene na nastavljanje želene temperature v stavbi. Stavbo modeliramo s simulatorjem, ki za dani urnik vrača njegove stroške in udobje. V stroških upoštevamo tudi neporabljeno energijo v bateriji, ki predstavlja prihodnji dobiček. Optimizacijo izvajamo z evolucijskim večkriterijskim optimizacijskim algoritmom, ki poišče kompromisne urnike glede na obravnavana nasprotujoča si kriterija.

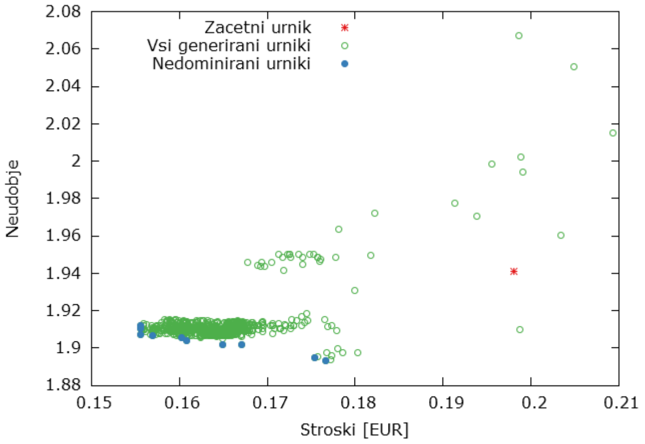
Optimizacijski algoritem kot vhodne podatke uporabi podani začetni (neoptimirani) urnik stavbe in podatke o ceni energije, napovedi sončne energije, porabnikih in navadah uporabnikov. Dokler ni izpolnjen ustavitveni pogoj (čas, ki je na voljo za optimizacijo), poteka preiskovanje prostora urnikov in njihovo vrednotenje preko omenjene simulacije.

Vsak urnik opisuje delovanje stavbe med podanima začetnim in končnim časom, pri čemer je vmesno obdobje razdeljeno na 15-minutne intervale. Urnik je sestavljen iz naslednjih štirih komponent:

- *Temperatura:* Za vsak interval določimo želeno temperaturo v stopinjah Celzija, ki mora zadoščati omejitvam (biti mora vsebovana v [T_{min}, T_{max}], kjer sta temperaturi T_{min} in T_{max} lahko podani za vsak interval posebej).
- *Energija+:* V primeru, da imamo presežek energije (fotovoltaični paneli proizvedejo več energije, kot je hiša porabi), za vsak interval določimo delovanje baterije. Možni sta le dve vrednosti, in sicer 1 (baterija naj se polni) in 0 (baterija naj se ne polni). Če se baterija ne polni, presežek energije prodajamo.
- *Energija–:* V primeru, da imamo primanjkljaj energije (fotovoltaični paneli proizvedejo manj energije, kot je hiša porabi), za vsak interval določimo delovanje baterije. Možni sta le dve vrednosti, in sicer 1 (baterija naj se prazni) in 0 (baterija naj se ne prazni). Če se baterija ne prazni, potrebno energijo črpamo iz omrežja.
- *Porabniki:* Za vsakega porabnika določimo čas, ko naj začne delovati, t.j. porabljati energijo. Končni čas in količina porabljene energije se izračunata iz lastnosti porabnika.

Optimizacijo smo preizkusili na naslednjem konkretnem primeru. Želimo optimirati vodenje stavbe s fotovoltaičnimi paneli, eno baterijo in enim porabnikom, ki mora delovati enkrat dnevno. Zanima nas vodenje stavbe za dva naslednja dneva: v prvem je napovedano jasno (sončno) vreme, v drugem pa pretežno oblačno vreme. Začetni urnik je določen na podlagi vremenske napovedi in uporabnikovih navad ter želenih temperatur. Optimizacijo izvajamo, dokler ne pregledamo 1000 urnikov.

Slika 2 predstavlja rezultate tega poskusa. Zeleni krožci prikazujejo vse generirane urnike. Začetni urnik je obarvan rdeče, nedominirani urniki (vseh je deset) pa so predstavljeni



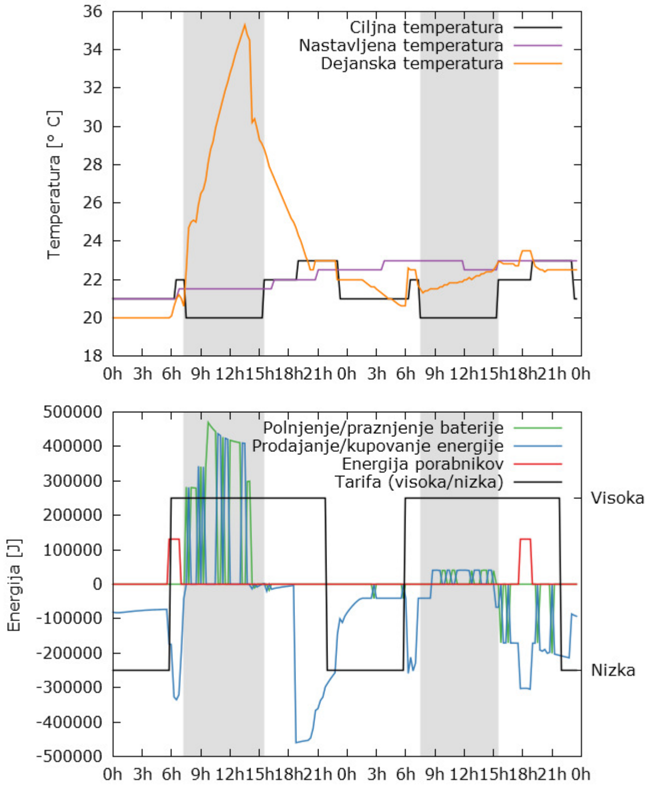
Slika 2: Vsi urniki dobljeni po postopku večkriterijske optimizacije

z modrimi pikami. Kot lahko vidimo, optimizacijski algoritem najde različne kompromise med stroški in udobjem, ki so po obeh kriterijih boljši od začetnega urnika. Najugodnejši dobljeni urnik je podrobneje obrazložen v nadaljevanju.

Slika 3 prikazuje dva grafa s podrobnejšo informacijo o najugodnejšem dobljenem urniku. Siva območja označujejo obdobja, ko uporabnik ni prisoten v stavbi. Takrat se meri samo poraba energije, ne pa tudi udobje. Zgornji graf na sliki 3 prikazuje ciljno temperaturo (tisto, ki ustreza največjemu udobju), nastavljeno temperaturo ter dejansko temperaturo, ki jo izmeri simulator, ko stavba poskuša voditi sisteme gretja in ohlajanja tako, da se čim bolj približa nastavljeni temperaturi. Vidimo, da je razkorak med nastavljeno in dejansko temperaturo precejšen predvsem v obdobjih neprisotnosti, ko se stavba ne hladi oz. ogreva. Spodnji graf kaže, kaj se v določenem intervalu dogaja z energijo. Baterija se včasih polni (pozitivna energija) in včasih prazni (negativna energija), energijo prodajamo (pozitivna energija) in kupujemo (negativna energija), vidimo tudi, v katerih intervalih obratuje porabnik. Na grafu je označena tudi tarifa kupovanja energije, ki je lahko bodisi visoka bodisi nizka. V prvem, sončnem dnevu, fotovoltaični paneli proizvedejo veliko energije, ki se deloma shrani v baterijo, deloma pa proda. V drugem dnevu je takšne energije zelo malo. Večina energije, ki se je shranila v baterijo, ostaja v bateriji tudi po koncu urnika (za porabo v prihodnjem obdobju).

4 RAZPRAVA

Prispevek opisuje arhitekturo programske opreme, ki s pametnim vodenje sistemov v stavbi rešuje pereč problem zagotavljanja visoke stopnje udobja in hkrati nizkih stroškov. Arhitektura temelji na ideji uporabe strojnega učenja uporabnikovih navad in potreb ter večkriterijske optimizacije parametrov vodenja na podlagi simulacije. Poleg arhitekture, ki omogoča vključitev v obstoječe sisteme pametnih stavb, so predlagane tudi formalne predstavitev problemov učenja, optimizacije in simulacije ter algoritmi, ki so primerni za reševanje teh problemov. Primer uporabe predlaganih rešitev kaže, da je tak sistem sposoben predlagati urnike, ki so udobnejši in cenejši od ročno nastavljenih urnikov, saj se lahko sproti prilagajajo zunanjim vplivom in potrebam uporabnikov. Uporaba takega sistema odpravi tudi potrebo po ročnem nastavljanju urnikov in hkrati spodbuja energetsko varčnost uporabnikov.



Slika 3: Podrobnosti najugodnejšega urnika. Siva območja označujejo obdobja, ko uporabnik ni prisoten v stavbi.

LITERATURA

[1] Gjoreski M., Gjoreski H., Piltaver R., Gams M. "Predicting the arrival and the departure time of an employee." *Zbornik 16. mednarodne multikonference Informacijska družba* – IS 2013, str. 43–46.

[2] *Annual Energy Outlook 2012*. U.S. Energy Information Administration (EIA), 2012.

[3] Perez-Lombard L., Ortiz J. in Pout C. "A review on buildings energy consumption information." Energy and Buildings, 40 (3): 394–398, 2008.

[4] Šef T., Piltaver R., Tušar T. "Projekt OpUS: optimizacija upravljanja energetsko učinkovitih pametnih stavb." *Zbornik 16. mednarodne multikonference Informacijska družba* – IS 2013, str. 110–113.

[5] Tavčar A., Piltaver R., Zupančič D., Šef T., Gams M. "Modeliranje navad uporabnikov pri vodenju pametnih hiš." *Zbornik 16. mednarodne multikonference Informacijska družba* – IS 2013, str. 114–117.

DETERMINATION OF CLASSIFICATION PARAMETERS OF BARLEY SEEDS MIXED WITH WHEAT SEEDS BY USING ANN

Kadir Sabancı[1], *Cevat Aydın*[2]
[1] Department of Electrical and Electronics Engineering, Batman University, Batman, Turkey
[2] Department of Agricultural Machinery, Selçuk University, Konya, Turkey
Tel: +904882173500; fax: +904882173601
e-mail: kadir.sabanci@batman.edu.tr

ABSTRACT

One of the basic problems that cause loss of yield in wheat is weed seeds that mixed with wheat seeds. In this study, discrimination of barley seed which mixed with wheat seeds has been realized. Classification of wheat and barley seeds has been achieved by using artificial neural network and image processing techniques. In the study, image processing techniques and the use of artificial neural network have been made possible with Matlab software. By using Otsu method, histogram data of seed images that were taken from web camera was obtained. By using histogram data, with multi-layered artificial neural network model, the system was educated and classification was made. Besides, wheat and barley seeds in the picture info where mixed seeds taken from the web camera exist were counted.

1 INTRODUCTION

Quality is one of the important factors in agricultural products marketing. Grading machines have great role in quality control systems. The most efficient method used in grading machines today is image processing.

Digitisation of the image is the process in which the image in the camera is converted to electical signals with optical – electrical mechanism [1].

Image processing, as a general term, is manipulation and analysis of the pictorial information [2].

Image processing techniques are used in different areas such as industry, security, geology, medicine, agriculture. Image processing and artificial neural networks are used in agriculture in fruit color analysis and classification, root growth monitoring, measurement of leaf area, determination of weeds [3,4,5,6,7].

Artificial neural networks is an information processing system which have been exposed with inspiration of biological neural networks and includes some similar performance characteristics to biological neural networks [8]. Simply, ANN that imitates the function of the human brain has several important features such as learning from data, generalizing, working with an unlimited number of variables etc.

It is seen that ANN is used in crop production, which constitutes an important field of agriculture engineering, in identification and classification stages of a wide range of agricultural products such as grape, wheat, peppers and olives [9].

In this study, a software has been developed for distinguishing the wheat and barley seeds which has been mixed during harvesting. Wheat and barley seeds which has been mixed, have been attempted to distinguish by using image processing techniques and artificial neural networks. Multilayer artificial neural networks has been performed for the process to be more precise and faster. System has been trained by using barley and wheat seeds pictures. Wheat and barley seeds have been classified successfully by using improved system. This study exemplifies image processing and artificial neural networks in agriculture.

2 MATERIAL METHOD

In this study, image of wheat and barley seeds photos have been taken by using a webcam with 1.3 MP (Mega Pixels) and having CCD sensor. Usage of image processing and artificial neural networks are provided by Matlab. In this study, 50 wheat seeds, 50 barley seeds were used. Black background is used at the stage of image processing for faster and correct results.

Firstly, wheat and barley seeds image information was received to obtain image informations that was to enter to Artificial neural networks. Picture information of wheat and barley seeds are shown in Figure 1.

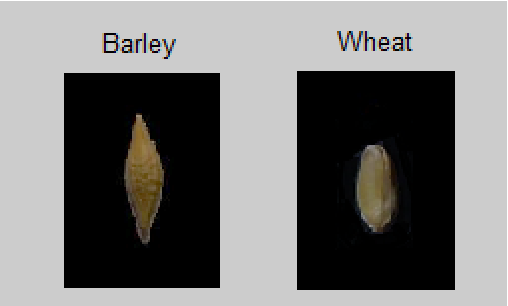


Figure 1: *Wheat and barley seeds*

Wheat and barley seeds image information was converted to gray level images. Filtration was performed to pictures tor reduce noise and interference. Wheat and barley seeds pictures which were converted into gray levels are shown in Figure 2.

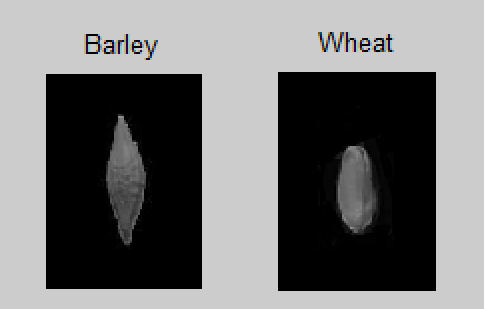


Figure 2: *Gray level images belong to wheat and barley seeds*

Image information which is at gray level were converted to black and white picture by using Otsu method. Otsu algorithm provides the clustering of these pixels according to the distribution of pixel values in the image. Thresholding process is one of the important processes in image processing. Especially, this method is used for highlighting closed and discrete areas of the object in the image. It includes the arrangement of image which was divided into pixels until to the image in dual structure. Simply, thresholding process is a process of discarding pixel values on the image according to specific values, and replacing other value / values. Thus determination of object lines and backgrounds of the object on the image were provided [10]. Threshold value is determined by using Otsu method. if it is under this value, pixels are converted to 0 value; if it is over this value pixels are converted to 1 value. Wheat and barley seeds pictures in black and white pictures are shown in Figure 3.

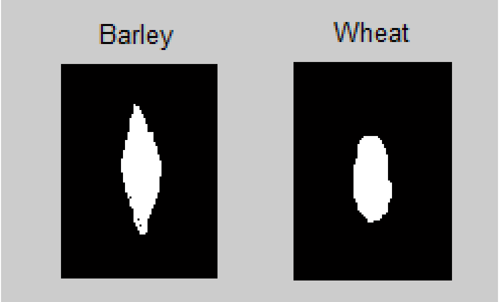


Figure 3: *Binary image information belong to wheat and barley seeds*

In this study, Matlab Software's Artificial Neural Network toolbox were used to distinguish wheat and barley seeds. ANN 's main tasks are to learn structure in the model data set, to make generalizations in order to fulfill to required task. To make this, the network is trained with the samples of related event to make generalization. Multi-layered artificial neural networks are the most commonly used in ANN models.

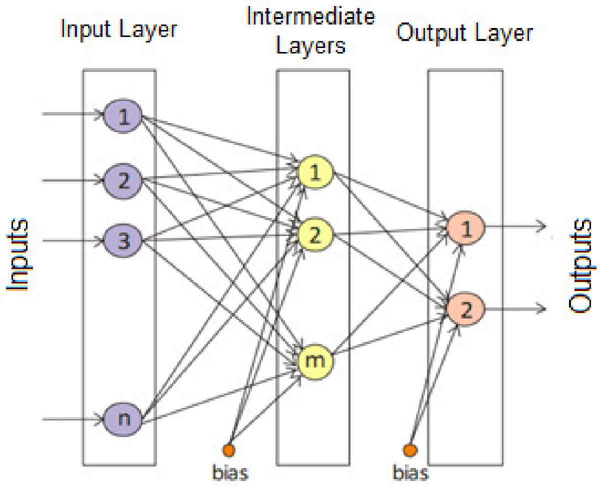


Figure 4: *Multi-layered artificial neural network*

In the study, neural network model with multilayer, feedforward, back propagation was used. Multilayer Perceptron (MLP) networks are a feedforward neural network model which has different number of neurons in the input layer, an intermediate layer consisting of one or more layers(s) and consisting of output layer. The structure of MLP neural network is shown in Figure 4. MLP neural network outputs of the neurons in a layer are connected to all input of the neurons with weights. The number of neurons in the input and output layer is determined according to the implementation problems. The number of intermediate layers, number of neurons in the intermediate layer and activation function are determined by the designer by trial and error method [11].

Segmentation process was performed by using digital image processing techniques on images belonging to mixed wheat and barley seeds and by determining the place of each seeds on the picture. Each pistachios was cropped in 100x100 pixels size. First of all, digital images of each seeds were converted to gray level images.

Picture was filtered in order to remove noise and very small objects (dust, etc..). Noise removed gray level pictures was converted to black and white picture by using Otsu method. data sets, which will enter to ANN, will be created by converting black and white picture informations in 100x100 size of each seeds to column matrix.

3 CONCLUSION

Classification process with MLP model average success of the test was determined %100 in the structure, where 100 neurons are used, in the hidden layer. When creating the MLP structure, neurons in the hidden layer and output layer activation function was used as logarithmic sigmoid. The error back-propagation was used in training of the ANN model algorithm and network was trained 250 steps. The results which was obtained in classification byusing MCA process are presented in Table I.

Table 1: *Classification results with using MCA process*

Number Of Neurons in Hidden Layer	Classification success (%)		
	Wheat seeds	Barley seeds	Average success
25	44	48	92
50	45	49	94
75	47	49	96
100	50	50	100

In this study, gray level images information of wheat and barley seeds by using image processing techniques. Afterwards, the system was trained by using Otsu Method, by converting binary picture information, by using multilayer neural network model. Then, in the realized system, the distinguishing of mixed wheat and barley seeds was performed.

System can be developed by using moving band and camera system and distinguishing of wheat and barley seeds can be carried out in real-time. Also, packaging process of seeds in a certain number can be performed. This study is an example of using image processing and neural network in agricultural field.

References

[1] Yaman, K., 2000. Görüntü işleme yönteminin Ankara hızlı raylı ulaşım sistemi güzergahında sefer aralıklarının optimizasyonuna yönelik olarak incelenmesi. Yayınlanmamış *Yüksek Lisans Tezi*, Gazi Üniversitesi, Fen Bilimleri Enstitüsü.

[2] Castelman, R. K., 1996. Digital image processing. Prentice hall, Englewood Cliffs, New Jersey, USA. Neuman, M. R., H. D. Sapirstein, E. Shwedyk and W. Bushuk. 1989. Wheat grain colour analysis by digital image processing. II. Wheat class discrimination. *Journal of Cereal Science* 10: 183-188.

[3] Keefe, P. D. 1992. A Dedicated wheat grain image analyzer. *Plant Varieties and Seeds* 5: 27-33.

[4] Trooien, T. P. and D. F. Heermann, 1992. Measurement and simulation of potato leaf area using image processing.Model development. *Transactions of the ASAE* 35(5):1709-1712.

[5] Pérez, A. J., F. Lopez, J. V. Benlloch and S. Christensen. 2000. Colour and shape analysis techniques for weed detection in cereal fields. *Computers and Electronics in Agriculture* 25: 197-212.

[6] Dalen, G. V. 2004. Determination of the size distribution and percentage of broken kernels of rice using flatbed scanning and image analysis. *Food Research International* 37: 51-58.

[7] Jayas, D. S. and C. Karunakaran. 2005. Machine vision system in postharvest tecnology. *Stewart Postharvest Rewiev,* 22.

[8] Fausett, L., 1994. Fundamentals of Neural Networks: Architectures, Algorithms and Applications, Prentice Hall.

[9] Kavas, G., Kavas, N., 2012. Gıdalarda yapay sinir ağları ve bulanık mantık. DÜNYA yayıncılık, *GIDA Dergisi* 2012-01: 93-96.

[10] Yaman, K., Sarucan, A., Atak, M., Aktürk, N., 2001. Dinamik çizelgeleme için görüntü işleme ve ARIMA modelleri yardımıyla veri hazırlama. *Gazi Üniv. Müh. Mim. Fak. Dergisi,* 16(1): 19-40.

[11] Öztemel E., 2003. Yapay Sinir Ağları. İstanbul: *Papatya Yayıncılık*

NOVI GOVOREC: NARAVNO ZVENEČ KORPUSNI SINTETIZATOR SLOVENSKEGA GOVORA

Tomaž Šef
Odsek za inteligentne sisteme, Institut "Jožef Stefan", Jamova cesta 39, 1000 Ljubljana
e-mail: tomaz.sef@ijs.si

POVZETEK

V članku je predstavljen prototip novega naravno zvenečega korpusnega sintetizatorja slovenskega govora. Temelji na govorni zbirki, ki jo razvijata Institut »Jožef Stefan« in podjetje Amebis. Trenutna demo verzija prototipa sintetizatorja uporablja četrtino te zbirke. Podprta sta po en moški in en ženski glas. Prototip je bil razvit v okviru projekta »Analiza in ovrednotenje naprednih tehnologij govorjenega jezika v pametnih stavbah« (Raziskovalni vavčer 2012, Amebis d.o.o., Kamnik).

1 UVOD

Za angleški jezik in druge večje jezike so različni govorno podprti sistemi že nekaj časa dosegljivi in imajo zelo širok krog uporabnikov. V zadnjem času se čedalje pogosteje uporabljajo tudi v različnih mobilnih aplikacijah, ki pa v našem domačem slovenskem jeziku žal niso dostopne oz. ne delujejo.

Najbolj naravno zveneči sintetizatorji govora temeljijo na korpusni sintezi. Metoda temelji na preiskovanju vnaprej posnete in označene govorne zbirke. Išče se zaporedja tistih posnetih glasov pri katerih se želene lastnosti čim bolj ujemajo. Kvaliteta takšnih sintetizatorjev govora je predvsem odvisna od zasnove govorne zbirke na kateri temeljijo. V splošnem velja, da je sintetizator govora kvalitetnejši, če uporabljamo za sintezo daljše osnovne segmente s čim manj spremembami prozodičnih parametrov, saj te povzročajo dodatna popačenja sintetiziranega govora [1]. Strošek razvoja korpusnih sintetizatorjev je izredno visok, zato je večinoma na razpolago le omejeno število glasov.

Komercialne raziskave s področja govornih in jezikovnih tehnologij so pri nas pogojene z majhnostjo slovenskega trga. Iz stroškovnega vidika je povsem vseeno ali razvijamo npr. sintetizator govora za jezik, ki ga govori milijarda ljudi, ali pa zgolj dva milijona. Slovenski trg je izredno majhen zato brez spodbud in subvencij s strani države razvoj tako kompleksnih tehnoloških izdelkov in storitev ni mogoč. Ob ustrezni subvenciji se za podjetje zmanjša tveganje (pre)velikih vlaganj v raziskave in razvoj, zato je podjetje pripravljeno vložiti tudi del lastnih sredstev. Potrebno je učinkovito sodelovanje z raziskovalno razvojnimi organizacijami, ker s tem podjetje pridobi dostop do najnovejšega znanja in tehnologij. Povečuje se delež vlaganj v raziskovalno razvojno dejavnost v celotnem prometu podjetja, poveča pa se tudi obseg sredstev za raziskovalno razvojno dejavnost.

Pričujoče delo je nastalo v okviru projekta »Analiza in ovrednotenje naprednih tehnologij govorjenega jezika v pametnih stavbah« (Raziskovalni vavčer 2012, Amebis d.o.o., Kamnik). *Namen projekta oz. raziskave* je pridobitev novega znanja in spretnosti za nadgradnjo obstoječih sistemov govornih in jezikovnih tehnologij z namenom uporabe v sodobnih inteligentnih vmesnikih pametnih stavb. Posebna pozornost je namenjena dinamičnemu podajanju govornih informacij. Projekt vsebuje dve aktivnosti. V okviru *prve aktivnosti* so se kritično analizirale in ovrednotile napredne tehnologije govorjenega jezika v pametnih stavbah. Sinteza govora, razpoznavanje govora, razpoznavanje govorcev ter njihovega psihofizičnega stanja s pomočjo računalniške analize govorjenega zvočnega signala, odpirajo povsem nove dimenzije razvoja inteligentnih uporabniških vmesnikov. Govorni vmesniki so nadvse primerna tudi kot pomoč invalidom (npr. slepim in slabovidnim), starejši populaciji in nekaterim drugim družbenim skupinam. *Druga aktivnost* se osredotoča predvsem na dinamično podajanje informacij ali opozoril v govorni obliki. Takšni sistemi so jezikovno odvisni, zato tujih rešitev ni mogoče kupiti oz. ustrezno prilagoditi našim potrebam. V Sloveniji se pojavlja čedalje večja potreba oz. povpraševanje po kvalitetnem, naravno zvenečem, razumljivem, čim širše sprejemljivem in splošno dostopnem govornem bralniku slovenskih besedil, zato je bila v okviru druge aktivnosti predlagana čim bolj optimalna zasnova in izvedba takšnega sistema. Rezultat te aktivnosti je tudi prototip novega Govoraca, ki bo v nadaljevanju podrobneje predstavljen.

2 GOVORNA ZBIRKA ZA KORPUSNO SINTEZO GOVORA

Najpomembnejša dejavnika pri snovanju govorne zbirke za potrebe korpusne sinteze govora sta izbira njene vsebine in označevanje posnetkov. Izbira velikosti govorne zbirke je posledica kompromisa med želenim številom variacij glasov oz. njihovim pokritjem na eni strani ter časom in stroški vezanimi na razvoj na drugi strani. Upoštevati je potrebno tudi čas za kasnejše preiskovanje govorne zbirke in potreben prostor za njeno hranjenje [2]. Kakovostna korpusna sinteza zahteva, da ima govorna zbirka pravilno označeno tako identiteto posameznih govornih segmentov kot njihov natančen položaj znotraj zbirke. Običajno avtomatskim metodam in postopkom sledi »ročno« popravljanje oznak, ki ga je ne glede na hiter razvoj tehnologije še vedno zelo veliko.

2.1 Zasnova govorne zbirke

Razvoj govorne zbirke za korpusno sintezo govora obsega naslednje korake:

- ustvari se obsežno tekstovno zbirko besedil, ki pokriva različne zvrsti (dnevni časopis, revije, leposlovje ipd.),
- iz zbirke besedil se odstrani vse oznake vezane na oblikovno podobo (glava besedila, tabele ipd.),
- okrajšave, števila ipd. se pretvori v polno besedno obliko (normalizacija besedil),
- besedila se pretvori v predvideni fonetični prepis (grafemsko-fonemska pretvorba),
- optimizira se obseg zbirke glede na vnaprej pripravljene kriterije (metoda požrešnega iskanja); doseči želimo statistično ustrezno vzorčenje izbranega področja govorjenega jezika,
- izbrane stavke se posname (ali pa se izlušči del obstoječih zvočnih zapisov),
- posneto govorno gradivo se fonetično in prozodično označi (samodejno grobo označevanje, fino ročno popravljanje).

Postopek za čim optimalnejšo izbiro povedi:

1. *Statistična obdelava besedil:*
   - Statistično obdelamo celoten besedni korpus in določimo pogostost pojavljanja posameznih glasov in glasovnih nizov v besedilu. Pri tem razlikujemo še med naglašenimi in nenaglašenimi glasovi ter glasovi, ki se pojavljajo na koncu stavka (oz. na mestih zajema zraka - ločila). Presledke na drugih mestih lahko ignoriramo oz. odstranimo.
   - Vključimo vse stavke (povedne, velelne, vprašalne itd.) in izdelamo statistiko posameznih vrst povedi oz. stavkov.
2. *Izdelava spiska glasovnih nizov z oceno zaželenosti posameznega niza:*
   - V spisek vključimo nabor vseh teoretično možnih kombinacij difonov; tudi tiste na katere pri statistični obdelavi nismo naleteli (zaradi robustnosti sintetizatorja govora).
   - V spisek vključimo vse trifone, štirifone in (po potrebi) ostale zaželene (najpogostejše) polifone, na katere smo naleteli pri statistični obdelavi besedil.
   - Utež oz. ocena zaželenosti niza je odvisna od pogostosti njegovega pojavljanja v besedilu.
3. *Postopek izbire povedi:*
   - Ocenimo doprinos glasovnih nizov za vsako poved iz tekstovnega korpusa.
   - Doprinos povedi je enak vsoti vseh ocen zaželenosti nizov (iz spiska), ki se v povedi pojavijo.
   - Doprinos posamezne povedi normiramo z dolžino povedi (št. besed v povedi ali št. fonemov v povedi).
   - Določimo takšno utež, da bodo dolžine izbranih stavkov čim bolj ustrezale statistični porazdelitvi dolžin stavkov iz korpusa.
   - Izberemo poved z najvišjim normiranim doprinosom.
   - Iz spiska odstranimo vse glasovne nize, ki jih izbrana poved vsebuje.
   - Ponovno ocenimo vsako poved in izberemo najboljšo (glede na novi spisek v katerem so izločeni tisti glasovni nizi, ki smo jih že pokrili) ter popravimo spisek.
   - Postopek ponavljamo dokler ne izberemo želenega števila povedi.
4. *Ovrednotenje rezultatov:*
   - Vsakih 1000 povedi izdelamo statistiko difonov, trifonov, štirifonov in drugih polifonov, ki jih že pokrivamo (gre za glasovne nize, ki smo jih do takrat že izločili iz zgoraj omenjenega spiska).
5. *Dodatne izboljšave algoritma:*
   - Ker mora zbirka vsebovati vse možne kombinacije difonov, algoritem popravimo tako, da difone dodatno utežimo glede na ostale polifone. Na takšen način bo algoritem na začetku dajal prednost povedim, ki bodo pokrile čim več novih difonov. Predvidoma se vsi difoni pokrijejo že po ca. 100 stavkih.
   - Pri trifonih in štirifonih upoštevamo pri robnih glasovih tudi podatek o glasovni skupini, ki ji pripadajo (npr. štirifon "krak" ne bo doprinesel prav dosti novega v našo zbirko, če ta že vsebuje štirifon "krat"; zato oceno koristnosti takega štirifona popravimo navzdol). To lahko naredimo preprosto tako, da v spisek vnesemo dodatne nize skupaj z njihovimi frekvencami pojavljanja v korpusu (primer takega štirifona: "k"+"r"+"a"+"pripornik").
   - Algoritem z različnim uteževanjem izboljšamo tako, da končni nabor vsebuje različne povedi (povedne, vprašalne, velelne, enostavne, sestavljene, naštevanje, itd.). Tako lahko isti korpus učinkovito uporabimo tudi za generiranje prozodičnih parametrov pri sintezi govora.

2.2 Snemanje govorne zbirke

Snemanje govorne zbirke je potekalo v studiu RTV Slovenija ob prisotnosti izkušenega tonskega tehnika. Med 10 profesionalnimi govorci smo izbrali najustreznejši moški in ženski glas. Med branjem besedila so govorci imeli nameščene elektrode Laryngographa, s katerimi smo spremljali nihanje glasilk za lažje kasnejše označevanje period govornega signala. Samo snemanje je zaradi obsežnosti besedila, ki ga je bilo potrebno prebrati trajalo več mesecev. Pri tem so nastavitve opreme ves čas ostale nespremenjene. Pred vsakim snemanjem je govorec poslušal svoje predhodne posnetke, s čimer se je skušalo zagotoviti čim bolj enak način govora, z enako intonacijo ipd.

2.3 Statistični podatki o govorni zbirki

V tabeli 1 so podani osnovni statistični podatki o govorni zbirki za korpusni sintetizator slovenskega govora Amebis Govorec.

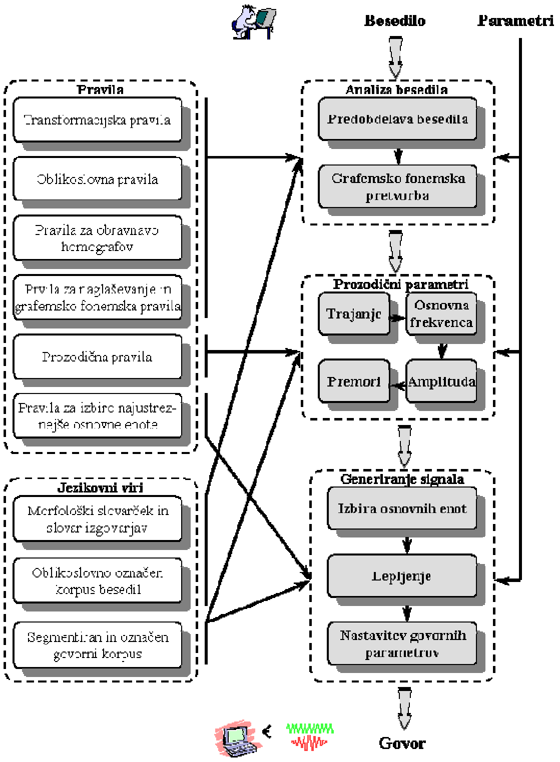
Velikost besednega korpusa	7.145.345 povedi 77 milijonov besed
Obseg govorne zbirke	**4.000 povedi** (46.785 besed)
Število različnih difonov	1.883
Število različnih trifonov (št. kombinacij v korpusu)	21.369 (24.702)

Tabela 1: *Statistični podatki o govorni zbirki Amebis Govorec*

3 NOVI GOVOREC

Novi Govorec za sintezo neomejenega slovenskega govora v osnovi ohranja nespremenjeno arhitekturo (slika 1) [3]:

- analiza besedila (predobdelava besedila, grafemsko fonemska pretvorba),
- nastavljanje prozodičnih parametrov (trajanje, osnovna frekvenca, amplituda, premori) in
- generiranje govornega signala (izbira osnovne enote, lepljenje, sprememba govornih parametrov).


Besedilo
Parametri
Pravila
Transformacijska pravila
Oblikoslovna pravila
Pravila za obravnavo homografov
Pravila za naglaševanje in grafemsko fonemska pravila
Prozodična pravila
Pravila za izbiro najustreznejše osnovne enote
Jezikovni viri
Morfološki slovarček in slovar izgovarjav
Oblikoslovno označen korpus besedil
Segmentiran in označen govorni korpus
Analiza besedila
Predobdelava besedila
Grafemsko fonemska pretvorba
Prozodični parametri
Trajanje
Osnovna frekvenca
Premori
Amplituda
Generiranje signala
Izbira osnovnih enot
Lepljenje
Nastavitev govornih parametrov
Govor

Slika 1: *Zgradba sistema Amebis Govorec za sintezo slovenskega govora*

Spremenjenih oz. na novo napisanih je le nekaj modulov:

- modul za nastavljanje prozodičnih parametrov je izpuščen; optimizacija nastavljanja teh parametrov je sestavni del algoritma za izbiro najustreznejših govornih segmentov
- možnost spreminjanja govornih parametrov je namenoma okrnjena; algoritem skrbi le še za glajenje prehodov na mestih lepljenja.

Potek korpusne sinteze [4, 5]:

- na razpolago imamo večje število primerkov posamezne enote,
- za vsak segment (difon), ki ga potrebujemo pri sintezi, v govorni bazi poiščemo takšnega, ki bo »najbolje« sintetiziral ciljni segment,
- najboljše zaporedje segmentov je tisto, ki minimizira ciljno ceno (angl. »target cost) in ceno združevanja (angl. »joint cost«) segmenta; problem je rešljiv z Viterbijevim algoritmom,

$$C(t_1^n, u_1^n) = \sum_{i=1}^{n} C^{target}(t_i, u_i) + \sum_{i=2}^{n} C^{join}(u_{i-1}, u_i)$$

  u_i predstavlja parametre i-tega segmenta, u_{i-1} parametre njemu predhodnega segmenta, t_i pa ciljne parametre i-tega segmenta,
- prva vsota ponazarja ceno zaradi razlike med ciljno in dejansko vrednostjo parametrov izbranih segmentov, drugi vsota pa ceno zaradi neujemanja parametrov na mestu spajanja dveh segmentov,
- parametri, ki jih upoštevamo pri računanju ciljne cene so: tip fonema, fonetični kontekst, naglas, pozicija znotraj besede in povedi, tip povedi, f0, trajanje ipd; posamezni parametri so različno uteženi (w_k),

$$C^t(t_i, u_i) = \sum_{k=1}^{p} w_k^t C_k^t(t_i, u_i)$$

- parametri, ki jih upoštevamo pri računanju cene združevanja pa so: ujemanje f0, ujemanje energije, ujemanje formantov in drugih spektralnih karakteristik (MFCC koeficienti); tudi tukaj so posamezni parametri različno uteženi (w_k)

$$C^j(u_{i-1}, u_i) = \sum_{k=1}^{p} w_k^j C_k^j(u_{i-1}, u_i)$$

- uteži pri računanju cene združevanja nastavljamo ročno s poslušanjem,
- uteži pri računanju ciljne cene lahko izračunamo avtomatično [4] s povezavo akustičnih razdalj ter višjenivojskih fonetičnih in prozodičnih parametrov; uporabimo linearno regresijo.

Algoritmi, ki združujejo daljše segmente, se izkažejo za boljše, zato k temu »teži« večina sodobnih algoritmov. Optimizirajo se predvsem fonetični in prozodični parametri, cena zderuževanja zaradi akustičnih parametrov je bolj v ozadju oz. se sploh ne upošteva.

4 SKLEP

Izdelali smo prototip kvalitetnega, naravno zvenečega, razumljivega in široko sprejemljivega, korpusnega sintetizatorja slovenskega govora. Zaenkrat je implementiranih le nekaj osnovnih algoritmov; naprednejši algoritmi so še v razvoju in se testirajo. Sintetizator trenutno uporablja le četrtino govornega korpusa (ca. 1000 stavkov na glas).

Uporabljena govorna zbirka pokriva skoraj vse možne kombinacije difonov in trifonov na katere smo naleteli pri analizi besednega korpusa s preko 7 milijoni povedi. Snemanje govorne zbirke (moški in ženski glas) je potekalo več mesecev. Za vsak glas je bilo prebranih preko 4.000 povedi povprečne dolžine 11 besed. Za lažje označevanje zbirke smo poleg govornega signala posneli še signal Laryngographa, ki prikazuje nihanje glasilk. Sledil je ročni pregled posnetega gradiva in grobo samodejno označevanje; temu sledi še fino popravljanje napak. Gre za najobsežnejšo izdelano govorno zbirko namenjeno sintezi slovenskega govora do sedaj [6,7,8].

Novi Govorec je že na začetku razvoja presegel naša pričakovanja. V veliko delih je umetno generirani govor tako dober, da ga marsikateri poslušalec težko oz. sploh ne loči od običajnih posnetkov (še posebej, če so ti predvajani preko mobilnih komunikacijskih naprav ipd.). Naravnost in razumljivost govora sta povsem primerljiva s sintetizatorji govora za druge večje jezike. Poslušanje takšnega govora ni naporno, zato je sintetizator primeren za najširši krog potencialnih uporabnikov.

Z nadaljnjim razvojem lahko upravičeno pričakujemo še dodatno občutno izboljšanje sintetiziranega govora. Do konca leta bo v novega Govorca vključen celotni govorni korpus, osnovni algoritmi pa bodo nadgrajeni z naprednejšimi in kompleksnejšimi. Govorni korpus bo dodatno pregledan in »očiščen« vseh zaznanih napak. Pri izbiri govornih enot bo uporabljena večkriterijska optimizacija glede na akustične, fonetične in prozodične kriterije. Uporabnik si bo sam izbral ali mu je ljubše, da sintetizator govora govori s čim bolj naravno prozodijo, ali pa mu je pomembnejša razumljivost in zveznost akustičnih parametrov, izbiranje čim daljših govornih enot ter njihovo lepljenje na fonetično najprimernejših mestih. Z drsniki ali izbiro na Pareto fronti bo na preprost način podal svoje preference.

Literatura

[1] A. Mihelič, J. Gros, N. Pavešič, M. Žganec: Pridobivanje govorne zbirke za korpusni sintetizator govora Phonectic, Zbornik konference Jezikovne tehnologije, str. 45-49, 2000.

[2] I. Amdal, T. Svendsen: Unit selection Synthesis Database Development Using Utterance Verification, Zbornik INTERSPEECH 2005, str. 2553-2556, 2005.

[3] T. Šef, Analiza besedila v postopku sinteze slovenskega govora, *doktorska disertacija*, Fakulteta za računalništvo in informatiko, Univerza v Ljubljani, 2001.

[4] A. Hunt, A. Black: Unit selection in a concatenative speech synthesis system using a large speech database Proceedings of ICASSP 96, vol 1, pp 373-376, 1996.

[5] P. Taylor: Text-to-Speech Synthesis, Cambridge University Press, 2009.

[6] T. Šef, M. Romih: Zasnova govorne zbirke za sintetizator slovenskega govora Amebis Govorec, Zbornik 14. mednarodne multikonference Informacijska družba, zvezek A, str. 88-91, 2011.

[7] M. Rojc, Z. Kačič: Design of optimal Slovenian speech corpus for use in the concatenative speech synthesis system, Proceedings of the Second international conference on language resources an evaluation, Athens, Greece, str. 321-325, 2000.

[8] J. Žganec Gros, A. Mihelič, N. Pavešič, M. Žganec, S. Gruden: AlpSynth – Concatenation-based Speech Synthesis for the Slovenian Language, 47th International Symposium ELMAR-2005, Zadar, Hrvaška, str. 213-216, 2005.

CLOUD-BASED RECOMMENDATION SYSTEM FOR E-COMMERCE

Gašper Slapničar[1,2]*, Boštjan Kaluža*[2]
[1]Faculty of Computer and Information Science, Večna pot 113, 1000 Ljubljana, Slovenia
[2]Department of Intelligent Systems, Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
E-mail: slapnicar.gasper@gmail.com, bostjan.kaluza@ijs.si

ABSTRACT

This paper leverages cloud-based machine learning platform to implement an item-based recommendation system for an e-commerce application. The solution is based on Prediction.IO platform, which offers a full-stack architecture based on MongoDB database, Hadoop framework for distributed processing, Apache Mahout scalable machine learning library, and RESTful API. We implemented an item-based recommendation engine for product suggestions in an online retail store using real-world data. Preliminary results are quite promising achieving Mean Average Precision of 6 %.

1 INTRODUCTION

A challenge that many retailers are facing today in the saturated online market is how to gain a competitive advantage and obtain cost-effective recommendation features without large investment into machine learning research and development. This challenge is difficult as we are getting large amounts of data even in smaller web applications. Therefore, it is expected that the solution is simple to implement, fast, distributed and scalable [1].

As a solution to this challenge, cloud-based recommendation systems offer recommendation-as-a service. These solutions are an emerging trend lately, with an open source server solution just recently raising $2.5 million in seed funding [2]. The main idea is to provide a recommendation engine in a cloud as a solution, whereas a retailer provides data in utilizes the output form the cloud.

The aim of this paper is twofold. The first goal is to review, compare and evaluate several cloud-based recommendation systems. The second goal is to implement a reference item-recommendation system that can be used in a real-world application. As a result, the paper aims at a fully working solution that is able to address the major challenges – scalability together with simple API access and implementation.

Preliminary results show the level of accuracy we can obtain by using the default set of algorithms and parameters on real-world data. These results can be later used as a baseline orientation point in further comparison with custom solutions.

2 CLOUD-BASED MACHINE LEARNING PLATFORMS

Recently, a wide variety of cloud-based recommendation systems emerged. All of them offer various machine-learning algorithms, while the implementations of the prediction model generation and supported algorithms differ. In our study, we focused on the currently available systems, which offered distributed, scalable, full-stack architecture, simple API access and were documented to a degree that allowed us a basic understanding of the whole platform. Available options for development and implementation of custom algorithms were also a priority and significant advantage of a platform.

BigML [3] is a commercial solution exclusively based on decision trees and therefore very suitable for classifications. It can be tested freely with smaller amounts of data and has a streamlined process for creating decision tree models. The interface is highly intuitive and the model creation is a matter of a few clicks. The main strength of it is the simplicity of usage and useful visualization of the generated decision tree model. It is however limited to decision trees, which are not suitable for recommendation problem.

QMiner [4] is an open source platform that offers high levels of customization, a decent amount of available algorithms, and is well documented. It is implemented in C++ and offers a JavaScript API. The architecture is distributed and customizable. However, the platform is still in development phase and faces some difficulties with deployment in production environment.

PredictionIO [5] is an open source solution implemented in Scala and offers APIs in most of the popular languages. It shares similarities with QMiner's distributed architecture and consists of four layers. At the bottom, there is a MongoDB database, followed by Hadoop framework for scalable distributed computing. The third layer is the heart of the prediction model generation process – Apache Mahout, a scalable machine-learning library with many popular algorithms already implemented. The top layer contains an API, which offers simple access to the prediction server.

Due to simple setup process, its open source nature and many available machine-learning algorithms, we chose PredictionIO for further steps.

Other solutions such as SensePlatform [6], Google Prediction API [7] and Microsoft Azure ML [8] are also available commercial solutions that mainly focus on cloud-based implementation of core machine-learning algorithms.

3 DATA

We obtained real-world data containing orders in an online retail store in a period of one year. The data was described by item (product) data, user (consumer) data and user-item interactions (item bought) data, for example, *user U1* made an *order O1* in which *items A,B,C,D* were bought.

In total, there were around 10.000 items, 36.000 users and 300.000 user-item interactions. There were some minor occurrences of missing data as well as some duplicated entries, which varied for a single character, yet represented the same product. Those entries were simply filtered with basic manual corrections. The only preprocessing included replacing the encoding of the original dataset with utf-8 encoding.

4 ITEM-BASED RECOMMENDATION

Our goal was to develop a recommendation system by leveraging the PredictionIO built-in engines. Figure 1 shows the PredictionIO architecture comprising several engines. Each engine processes data and constructs a predictive model independently. There can be several engines within a single application, where each of them will serve its own prediction results based on corresponding predictive model. Each engine can be configured with a variety of options and parameters for fine tuning, such as preference for newer items, preference for surprising discovery, custom attributes and most notably the goal to be maximized through predictions. Based on the selected goal, which can be any action (*like, view, conversion*) or a rating threshold (e.g. *rating* >= *3*), it is possible to evaluate the available algorithms using built-in interface.

4.1 Recommendation engines

PredictionIO server offers two recommendation engines: item recommendation engine and item similarity engine. Each engine builds a prediction model based on the underlying algorithm. In both cases, prediction model is generated using Mahout's Collaborative Filtering [9] methods.

Mahout's kNN (k-nearest neighbors) Item Based Collaborative filtering was used in Item recommendation engine and Mahout's Item Similarity Collaborative Filtering was used in Item similarity engine. Both are implemented to be run either on a single machine or multiple machines in a distributed/scalable setting.

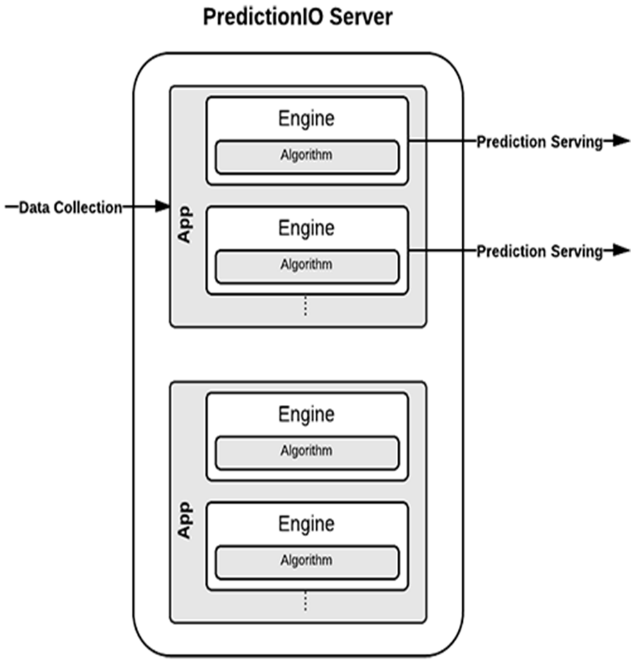


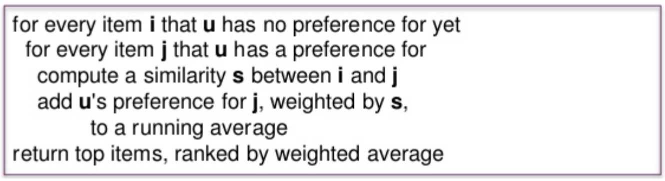
Figure 1: PredictionIO server architecture [14].

Collaborative filtering [11, 12] is among the most used and successful methods for this type of recommendation systems. Collaborative filtering finds the users with similar preferences (user based) in such a way that it finds items, which were similarly rated by other users (item based).

User based approach has some issues, especially with scalability, since computation grows with both the number of users and the number of items; hence, item-based approach is more common due to its simplicity and better scalability.

The basic idea of the item based approach is to take the items some user has rated/bought and computes how similar they are to other items. Based on this similarity it then selects *k* most similar items.

Mahout's item based collaborative filtering implementation is based on the pseudo-code shown in Figure 2. The algorithm computes similarity between pairs of items, where one item of the pair is an item already preferred by a user, and the other item of the pair is not.



```
for every item i that u has no preference for yet
  for every item j that u has a preference for
    compute a similarity s between i and j
    add u's preference for j, weighted by s,
         to a running average
return top items, ranked by weighted average
```

Figure 2: Pseudo-code for recommendation algorithm.

To demonstrate the general algorithm from Figure 2 on an e-commerce example, it tells us that for each product that is not in a user's shopping cart yet, it takes each product already in the shopping cart and computes the similarity between these pairs of products. It then weighs the computed preference with frequency of common occurrences.

4.2 Mean average precision as evaluation criteria

For result evaluation, PredictionIO offers built-in evaluations where we can regulate the size of the learning and testing set and also some other parameters such as the number of predictions and number of iterations. These simulated evaluations use Mean Average Precision (MAP) to measure accuracy [10].

The critical step is the evaluation of similarity. PredictionIO offers several similarity evaluation options, such as Pearson's correlation, Cosine similarity, Jaccard coefficient, Log-likelihood ratio etc. This offers a simple option to compare the results each of them gives. In our case, Log-likelihood ratio was chosen as it fits our problem of product recommendation best.

This similarity measure is based on finding and counting the cases, where two items appear together [15] and is similar to an expanded co-occurrence matrix.

If we want to measure the precision of the prediction for *n* "best recommendations" for some user, we use *MAP@n*:

$$\text{MAP}@n = \sum_{i=1}^{N} ap@n_i / N,$$

where N is the number of all users, while ap@n_i is the average precision for user *i* and is defined as:

$$\text{ap}@n = \sum_{k=1}^{n} P(k)\Delta R(k),$$

where P(k) is the precision with recommendation of *k* items, which means the number of correctly recommended items from the first *k* recommendations. *ΔR (k)* is the change of recall in step *k*, which means *1/n* if the *k*-th item was recommended correctly, and *0* otherwise. [10]

5 EXPERIMENTAL RESULTS

We ran the evaluation in three iterations, each prediction offering 20 items, with 70% of total data being the training set and 30% being the testing set. The evaluation was first ran for a random recommendation algorithm in order to get a baseline with which we can compare our results. We then ran it for item recommendation engine and item similarity engine, using the above-mentioned Collaborative filtering algorithms with Log-likelihood as a similarity measure.

The best result was 6% MAP accuracy using item similarity engine as seen on Figure 3. This is notably better than the 0.1% MAP accuracy using baseline random recommendation. For our el-commerce example this means that given 100 recommended products, 6 of these would actually be chosen by a customer and added in their shopping cart.

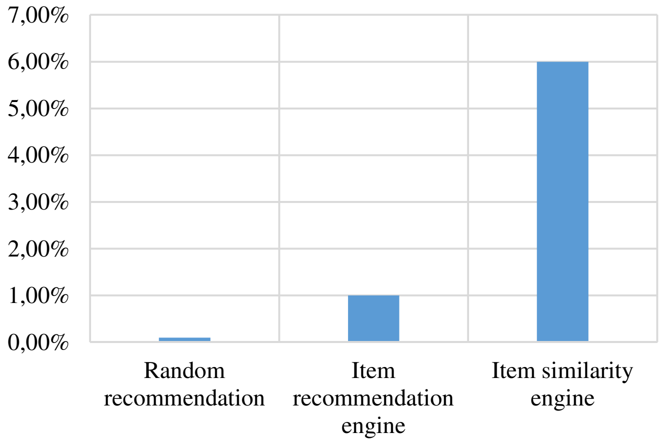
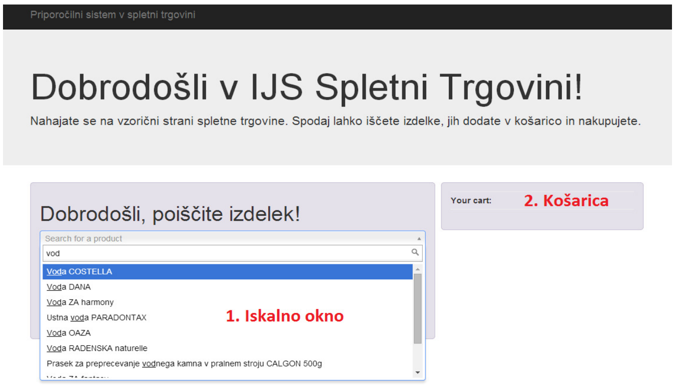


Figure 3: Comparison of MAP prediction accuracy.

6 REFERENCE IMPLEMENTATION

Based on the comparison presented in the previous section, we reused the best model and implemented PredictionIO within an Ajax based Django application, which implements a demo e-commerce application. This allowed us to evaluate the response time and application behavior and reliability in a test user environment.



Dobrodošli v IJS Spletni Trgovini!

Nahajate se na vzorični strani spletne trgovine. Spodaj lahko iščete izdelke, jih dodate v košarico in nakupujete.

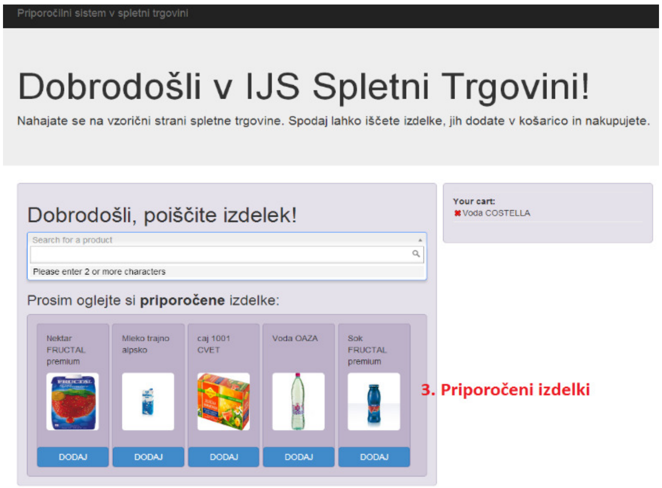
Dobrodošli, poiščite izdelek!

1. Iskalno okno

2. Košarica

Figure 4: Entry site of our demo e-commerce application [13].

The application offers available products to a user dynamically upon input as seen on Figure 4 (*1. Iskalno okno*), which allows us to prevent invalid entries and improve the user experience. Upon selection a set of five products is returned with minimal response time as seen on Figure 5 (*3. Priporočeni izdelki*).



Priporočilni sistem v spletni trgovini

Dobrodošli v IJS Spletni Trgovini!

Nahajate se na vzorični strani spletne trgovine. Spodaj lahko iščete izdelke, jih dodate v košarico in nakupujete.

Dobrodošli, poiščite izdelek!

Search for a product

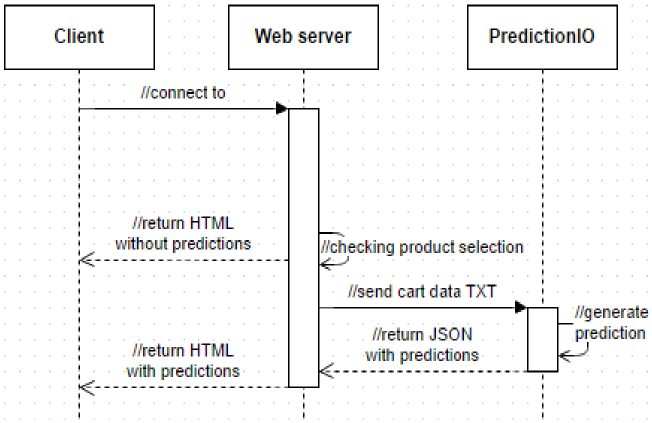
Please enter 2 or more characters

Prosim oglejte si **priporočene** izdelke:

Your cart:
✖ Voda COSTELLA

Figure 5: Application offering 5 products to a user who selected a random product [13].

As displayed on Figure 6, client connects to a web server running on a remote machine. If no products are chosen, the server returns a simple html site. When a client selects a product from the list, the server detects it and collects the data from the shopping cart in form of strings. This data is sent to PredictionIO. PredictionIO processes the data, applies a prediction model and then returns a JSON response containing recommendations to the web server. The webserver parses the response and displays it to the client. It is important to note that minimal coding is required for interaction with PredictionIO.



Client
Web server
PredictionIO
//connect to
//return HTML without predictions
//checking product selection
//send cart data TXT
//generate prediction
//return JSON with predictions
//return HTML with predictions

Figure 6: UML diagram of client - web server - PredictionIO interaction.

A single machine implementation offered good response time for a small set of users trying to access the application simultaneously. In order to achieve the scalability and serve multiple requests per second, we would have to deploy multiple processing machines.

7 CONCLUSION

We overviewed and compared a set of distributed, scalable cloud-based platforms for machine learning. The goal we achieved is a fully functional demo e-commerce application with product recommendation service based on real world data. The application is capable of taking a set of products chosen by the customer and returning a set of *n* products, which are predicted to be the most likely to be bought by this customer. The prediction model utilizes Prediction.IO platform using Mahout's Item Based Collaborative Filtering, using Log-likelihood ratio as a similarity measure.

The preliminary experiments showed that we can expect up to 6% MAP accuracy of predictions. This is an important result, which can be used as a baseline value in future work. It allows us to compare a custom-developed recommendation algorithms with a basic out-of-the-box solution.

The experiment established grand basis for further research, that is, full-stack architecture, modeling, evaluation, and testing. Further work will include implementation of a custom engine for prediction and model generation, which could be easily used in the prebuilt architecture.

References

[1] Very Large Data Bases Endowment Inc., A. Labrinidis, H.V. Jagadish, *Challenges and Opportunities with Big Data*. http://vldb.org/pvldb/vol5/p2032_alexandroslabrinidis_vldb2012.pdf, 2014-09-05

[2] Techcrunch, S. O'Hear, *PredictionIO Raises $2.5M For Its Open Source Machine Learning Server*. http://techcrunch.com/2014/07/17/predictionio/, 2014-09-05

[3] BigML, Inc. https://bigml.com/, 2014-09-05

[4] QMiner. http://qminer.ijs.si/, 2014-09-05

[5] PredictionIO. http://prediction.io/, 2014-09-05

[6] Sense. https://senseplatform.com/, 2014-09-05

[7] Google Developers, Google Prediction API. https://developers.google.com/prediction/?hl=sl, 2014-09-05

[8] Microsoft Azure Machine Learning. http://azure.microsoft.com/en-us/services/machine-learning/, 2014-09-05

[9] The Apache Software Foundation, Apache Mahout. https://mahout.apache.org/users/recommender/recommender-documentation.html, 2014-09-05

[10] Kaggle Inc. https://www.kaggle.com/wiki/MeanAveragePrecision

[11] T. Segaran, *Programming Collective Intelligence*, 2007

[12] B. Sawar, G. Karypis, J. Konstan, J. Riedl, *Item-Based Collaborative Filtering Recommendation Algorithms*. http://www.ra.ethz.ch/cdstore/www10/papers/pdf/p519.pdf, 2014-09-05

[13] Demo e-commerce application. http://prediction-io.ijs.si:8001/

[14] Basics of PredictionIO. http://docs.prediction.io/current/concepts/basics.html

[15] T. Dunning, *Surprise and Coincidence*. http://tdunning.blogspot.com/2008/03/surprise-and-coincidence.html, 2014-09-05

NOVEL IMAGE PROCESSING METHOD IN ENTOMOLOGY

Martin Tashkoski, Ana Madevska Bogdanova
Ss. Cyril and Methodious University, Faculty of Computer Sciences and Engineering,
Rugjer Boshkovikj 16, 1000 Skopje, Macedonia
Tel: +389 2 3070377; fax: +389 2 3088222
e-mail: tashkoskim@yahoo.com, ana.madevska.bogdanova@finki.ukim.mk

ABSTRACT

Image processing and machine learning together offer powerful methods for image classification. In this paper we present novel way of processing microscopic images for automatic classification of two similar insects belonging in the family Aleyrodidae, superfamily Aleyrodoidea (whiteflies). They are very similar and can be distinguished only in a certain stage of their development (fourth larval stage or "pupae stage"). Following our previous work, we propose a novel image processing method for automatically removing images' background noise. We also present results of a classification process using the processed images with the proposed method.

1 INTRODUCTION

Microscopic image processing dates back half a century when it was realized that some of the techniques of image capture and manipulation, first developed for television, could also be applied to images captured through the microscope.

Since agriculture has been important to people thousands of years, we focus on microscopic images of pests that harms agricultural crops.

In this paper we present one of the pests' problems – distinguinshing between two whiteflies lat. Trialeurodes vaporariorum (less dangerous) and lat. Bemisia tabaci (more dangerous). They are small insects (about 1mm long) with wings and bodies all covered with white, powdery wax [1]. They are fed with the plants juice and as a consecuence, they may reduce vigour and growth of the plant [2]. Bemisia tabaci is more dangerous than Trialeurodes vaporariorum because their larvae can inject some enzymes into the plant and those enzymes cause chlorosis or uneven ripening (depending on the plant), and induce physiological disorders [2].

In Macedonia, these whiteflies can be found in the southern parts and their occurrence has been observed 5 – 6 years ago [3]. It is assumed that they are transferred from neighboring countries by importing various plants that causes problems with their spreading.

The border custom control has developed interest in preparing an intelligent system for dangerous pest recognition that will stop the transfer of contaged plants and fruits at the border.

Following our previous work [CIIT 2014], in this paper we propose a new method - Self – removing Noise Method (SRNM) for automatic image background cleansing. We shortly describe the method for solving the problem for distinction of the whiteflies using image processing and machine learning techniques, and give some results gained from different tests. The rest of the paper is organized as follows. In Section 2 we present some of the work related to our problem. In Section 3 we present the entomology problem and the basic idea for solving this problem. The results of the different test are presented in Section 4, and the conclusion and future work are presented in the final Section 5.

2 RELATED WORK

In this section we briefly present some work related to the problem of these two whiteflies and some techniques and existing software for image processing.

The authors in [3] discovered the Bemisia tabaci and Trialeurodes vaporariorum in Macedonia. They explain the danger of these pests and guess that these whiteflies were transferred from other neighboring countries.

Authors in [6] explain all of the stages and the physical look, that can be useful for understanding the differences of the whiteflies and choosing the best indicator for their distinction.

The first step for solving this problem is undertaken in [5]. The authors proposed the algorithm Symmetrical Self – Filtration that can extract the important (vasiform orifice) part from the microscopic images in pupae stage. This is the insect part we use for the classification process in our work.

Authors in [6] present an important insight in explaining various methods for processing microscopic images. Their work on this specific area of images is very useful, because often there are dust and particles that appears as noise in the image and the extraction of some information is very difficult task.

The authors in [7] and [8] are using Wolfram Mathematica to solve some biological problems with image processing. The author in [7] presents a solution for measuring the locations of particles of microscopic images. It is about needle – free injection devices that fire powdered drug particles into the skin on a supersonic gas shock wave. Thin slices of a target are photographed by microscope, and on that images are applied different filters and the noise of the images is removed to find the locations of the particles. The author in [8] analyze a microscopic image of red blood cells and count them, using morphological operations and measurement tools.

3 THE METHODOLOGY

In this section we present the entomology problem at hand and the methodology used for creating a descriptor, the image processing process, removing the background of the images, and the classification process.

Although the whiteflies look similar, they have differences in each stage of their development. But the problem is that these whiteflies can physically change their look depending on the plant they feed themselves with, their environment and its temperature. According to [2], the accurate indicator for distinction is based almost entirely on their fourth larval stage or "puparial stage".

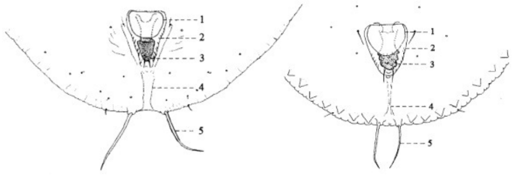


Figure 1: Bemisia tabaci and Trialeurodes vaporariorum, closer look to "vasiform orifice"

On Figure 1 we can see the two larvae of both whiteflies with their characteristic parts. The parts of the larvae are: operculum (1), vasiform orifice (2), lingula (3), caudal furrow (4), caudal seta (5). The characteristic part that is the best indicator for distinction of these two whiteflies is vasiform orifice (2). Vasiform orifice of Bemisia tabaci is thinner than the one of Trialeurodes vaporariorum.

3.1 Obtaining the training and test set

In order to develop an Entomology classification System, the substantial part of the solution was to find appropriate images of these two whiteflies. The images of Bemisia tabaci were found on the Internet database. Since there is no available data base for the second whitefly, we had to make an image collection of ourown. The whitefly Trialeurodes vaporariorum was found in greenhouse in the southern parts of Macedonia, on the plants cucumber and tomato. There are several steps of taking these images: preparation of the biological samples, awareness of the mechanical damage of the samples, type and quality of the microscope. The most difficult problem was that each larva was covered with dust which appeared as a noise at the image. At Figure 2 there are microscopic images with different zoom levels of the larva. The top image is larva captured before removing the dust.

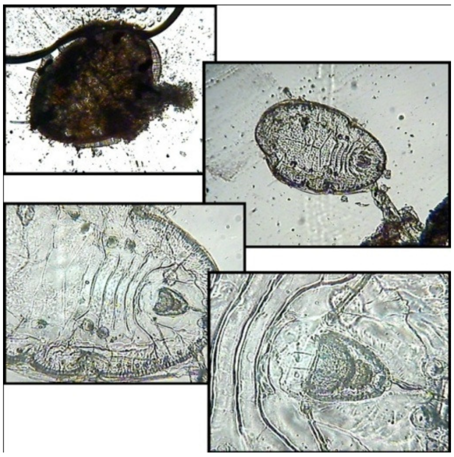


Figure 2: Microscopic images of larva of Trialeurodes vaporariorum

3.2 Image Preprocessing

First of all, the captured images were cropped by applying the algorithm for Symmetrical Self – Filtration [5], and as an output we obtained the images with the characteristic part (vasiform orifice). These images were used as input to our method for automatically removing the background.

3.3 Image Processing - Self – removing Noise Method (SRN Method)

The new method proposed in this paper, the Self – removing Noise Method, removes the noise and the background of the processed microscope images obtained as output in 3.2. in order to prepare the images for the classification process. The vasiform orifice part is not clearly presented in all of the images. Some of them are really noisy. For solving this problem we developed a method that proceeds all of the images, applies some filters, and after that as an output returns the images with the vasiform orifice part in a white background.

We will present the basic steps (1-6) of the method applied on all of the images - Figure 3.

The SRN Method is as follows:

Step 1: importing the images with the vasiform orifice part;

Step 2: converting each of the images to "grayscale" image;

Step 3: producing 3 categories for each image and classifying the pixels according to their color intensity (black, grey and white);

Step 4: applying bilateral filter to the images;

Step 5: morphological image processing;

Step 6: converting each of the images to binary image;

Step 7: deleting the isolated small groups of black pixels.

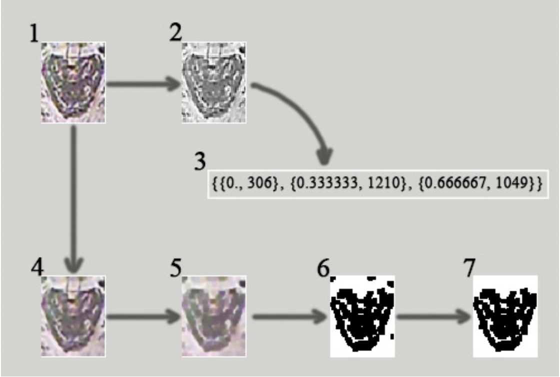


Figure 3: The basic steps in the Self – removing noise Method for removing the background

In Step 1, we import the images with the vasiform orifice part. The second step converts the images to “grayscale” style.

In Step 3, for each image we form 3 dimensional vector of pairs, in order to find the parameters that characterize the image the best way according their color intensity. Because the images were converted to “grayscale” style in Step 2, the pixels values variy in scale of 0 to 1 (0 for black and 1 for white). First pair represents the number of pixels that has values 0 – 0.33 , the second pair – number of pixels with values 0.33 – 0.67, and the third pair – number of pixels with values 0.67 – 1 (Figure 3). This information is used in the following steps to adjust the appropriate filters.

The fourth step is about applying bilateral filter on each of the images, as it is shown at Figure 3. The bilateral filter [9] is a non – linear, edge – preserving and noise – reducing filter for images. The intensity value at each pixel in an image is replaced by a weighted average of intensity values from the nearby pixels. This weight can be based on a Gaussian distribution.

In Step 5 we made some morphological image processing. We used the method of closing as the basic rule for removing noise. Closing removes small holes and it tends to enlarge the boundaries of bright regions in the image and shrink background color holes in such regions.

The sixth step is converting each of the images to binary images, and after that in the last step we use function for deleting the small groups of black pixels.

Figure 4 presents some rejected images. The method counts the pixels at the borders (up, down, right and left). If there are more black pixels than a half of the border length, these images are rejected for the classification process as undistinguishable.

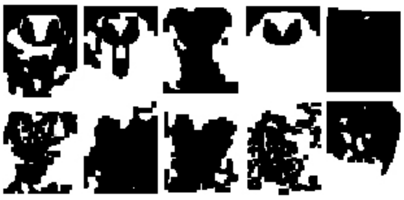


Figure 4: Rejected images

Figure 5 present some of the accepted images that were obtained as outputs of the SRN Method.

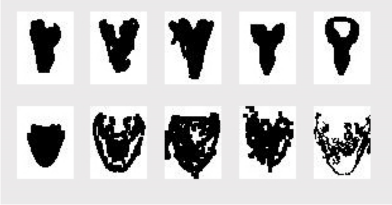


Figure 5: Accepted images (the first row are images of Bemisia tabaci, the second row are images of Trialeurodes vaporariorum)

4 TESTS AND RESULTS

The following tests we undertaken using the code written in Visual Studio 2010 in C# presented in [4]. The processed images obtained with the proposed SRN Method are used as input for the code, and the output is a file with parameters formatted properly for Weka and SVM light. As we presented in paper [4], the best descriptor contains the following parameters: five different widths of the image (number of pixels from the image in one row, taken consecutively from top to bottom through the image on equal distances), height, and ratio (height/bottom_width). The SRN Method was used on 346 images total, 49 of Bemisia tabaci and 297 of Trialeurodes vaporariorum. The images of Trialeurodes vaporariorum were gathered from different sources and contained different background colors. The script as output obtained 326 images total (39 of Bemisia tabaci and 287 of Trialeurodes vaporariorum) ready for classification and 20 rejected images.

4.1 Classification in Weka and SVM light

We made 3 groups with 10 tests and used different classifiers in Weka and SVM light [11]. For the first two groups (presented in [4]) we had 109 images total with manually removed background (48 images of Bemisia tabaci and 61 of Trialeurodes vaporariorum). In the first group we maintain even ratio in the test folder (we use 10 instances of both classes for testing), and in the second group we maintain even ratio in the training folder (we use 40 instances of both classes for training). For the third group we had 326 images total with automatically removed background (39 images of Bemisia tabaci and 287 images of Trialeurodes vaporariorum). In this group we maintain the even ratio in the test folder (we use 10 instances of each class for testing).

4.2 Results

In this section we present results of the classification process using the images obtained with the proposed SRN Method i.e. with automatically removed background. We maintain the even ratio in the test folder.

In the paper [4] we have published the classification results of manually cleaned images where we maintained the even ratio in the test folder. According the average best results in Weka, for Bemisia tabaci and Trialeurodes vaporariorum were obtained with the classifier lazy.IBk (90% correctly recognized instances of Bemisia tabaci, and 95% correctly recognized instances of Trialeurodes vaporariorum). According to the average best results in SVM light for correct recognition of Bemisia tabaci were obtained with the RBF kernel (for gamma=0.01) 89%, and best results for correct recognition of Trialeurodes vaporariorum were obtained with all of the kernels except the RBF kernel (for gamma=0.01) 97%.

For the new group of tests with the SRN method, we have produced 10 tests, and for each test we have taken 10 instances of Bemisia tabaci and 10 instances of Trialeurodes vaporariorum for testing of the total set with 326 instances. For every test we performed classifications in Weka (for different classifiers) and in SVM light (for different kernels).

According the average best results (Figure 6) in Weka for Bemisia tabaci, were obtained with the classifiers Bayes.BayesNet (**78%** correctly recognized), and best results for correctly recognizing Trialeurodes vaporariorum were obtained with the classifier trees.J48 - **89%**. According to the average best results (Figure 6) in SVM light for correctly recognizing Bemisia tabaci were obtained with the RBF kernel (for gamma=0.001) **70%**, and best results for correct recognition of Trialeurodes vaporariorum were obtained with all of the kernels except the RBF kernel (for gamma=0.001) **85%**.

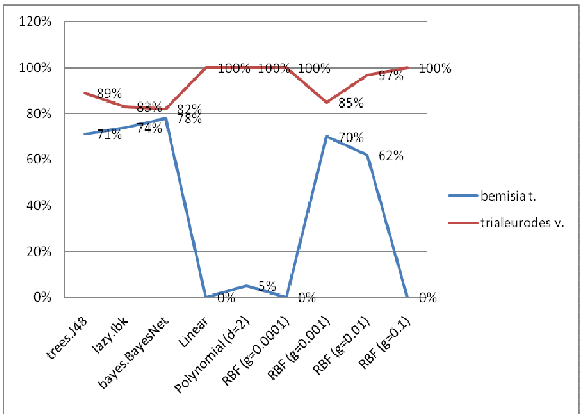


Figure 6: Average of the results in Weka and SVM light with even ratio of the both classes in the testing set (for automatically cleaned images)

6 CONCLUSION

In this paper we proposed a new method for automatically removing background of the images of Bemisia tabaci and Trialeurodes vaporariorum - Self – Removing Noise Method in order to develop an automatic system for classification. We tested with several classifiers, using the images that we obtained with this method and we presented the results. According the new results we can conclude that the Self – removing noise method proved as a good solution for automatically cleaning the images. The SRN Method that we used for filtering the images will be improving in the future in order to obtain results approximate to the results when using manually cleaned images in the classification process. We will extend our set of images of both pests and the final step will be connecting all of these methods into one integrated system for fast recognition of the pests with an user - friendly interface. We also believe that the database of the whitefly Trialeurodes vaporariorum we created, will be usefull to the other entomology researchers.

References

[1] G. S. Hodges, G. A. Evans. An identification guide to the whiteflies (Hemiptera: Aleyrodidae) of the southeastern United States. Florida Dept. Agriculture, Division of Plant Industry, Gainesville, FL 32614. 2005.

[2] C. Malumphy. Protocol for the diagnosis of quarantine organism Bemisia tabaci (Gennadius). Central Science Laboratory, Sand Hutton, York YO41 1LZ, UK.

[3] С. Банџо, Р. Русевски. Bemisia tabaci Genad. Присуство и распространување во Република Македонија, XXXI - во традиционално советување за заштита на растенијата на Република Македонија, Охрид, 2006. In Macedonian.

[4] M. Tashkoski, A. M. Bogdanova. Image classification in Entomology. In: Proceedings of 11th International Conference on Informatics and Information Technologies, CIIT 2014.

[5] М. Киндалов. Препознавање на слики во ентомологијата и практична имплементација со Bemisia tabaci и Trialeurodes vaporariorum. Дипломска работа, ментор: проф. Д-р Ана Мадевска Богданова, Скопје (2009). In Macedonian.

[6] Q. Wu, F. A. Merchant, K. R. Castleman. Microscope image processing. ISBN: 978-0-12-372578-3. Elsevier Inc. 2008.

[7] J. McLoone. Buiding a microscope application in Mathematica, 2011. http://blog.wolfram.com/2011/09/09/building-a-microscopy-application-in-mathematica/

[8] S. Ashnai. How to count cells, annihilate sailboats, and warp the Mona Lisa, 2012. http://blog.wolfram.com/2012/01/04/how-to-count-cells-annihilate-sailboats-and-warp-the-mona-lisa/

[9] C. Tomasi, R. Manduchi. Bilateral Filtering for Gray and Color Images. Print ISBN: 81-7319-221-9. Proceedings of the 1998 IEEE International Conference on Computer Vision, Bombay, India, 1998.

[10] M. H. Malais, W. J. Ravensberg. Knowing and recognizing - The biology of glasshouse pests and their natural enemies. Publisher: Koppept Biological Systems. ISBN 90 5439 126 X. 2003.

[11] T. Joachims. Optimizing Search Engines Using Clickthrough Data, Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD), ACM, 2002.

ARHITEKTURA SISTEMA OpUS

Aleš Tavčar[1,2]*, Jure Šorn*[1]*, Tea Tušar*[1]*, Tomaž Šef*[1]*, Matjaž Gams*[1,2]
Odsek za inteligentne sisteme, Institut »Jožef Stefan« [1]
Jamova cesta 39, 1000 Ljubljana, Slovenija

Mednarodna podiplomska šola Jožefa Stefana[2]
Jamova cesta 39, 1000 Ljubljana, Slovenija
e-mail: {ales.tavcar, tea.tusar, tomaz.sef, matjaz.gams}@ijs.si

POVZETEK

Obstoječi sistemi hišne avtomatizacije oz. pametnih stavb ne omogočajo naprednih funkcionalnosti, ki bi jih od takih sistemov pričakovali. Trenutne rešitve omogočajo zgolj spremljanje stanja sistemov in okolja v hiši ter krmiljenje hišnih naprav preko mobilnih naprav in spleta. Definiranje urnikov delovanja je prepuščeno samim uporabnikom, kar se običajno odraža v večji porabi in neučinkovitem delovanju. V pričujočem prispevku so predstavljeni načini za nadgradnjo sistemov hišne avtomatizacije z inteligentnimi metodami učenja navad uporabnikov in metodami optimizacije delovanja. Tak sistem je zmožen spremljanja obnašanja uporabnikov, se učiti njihovih navad in prilagajati delovanje glede na spreminjajoče se življenjske navade in potrebe.

1 UVOD

Trenutne komercialne rešitve hišne avtomatizacije ne ponujajo naprednih funkcionalnosti, ki bi jih pričakovali v takih sistemih. Namesto pametnega predvidevanja navad in potreb uporabnika obstoječi sistemi ponujajo zgolj krmiljenje hišnih naprav ter nadzor stanja v zgradbi. Slednji poteka preko različnih pametnih naprav in v redkih primerih preko spletnih vmesnikov. Nastavljanje režima delovanja in scenarijev je prepuščeno samim uporabnikom, iz česar običajno sledi, da so urniki nastavljeni površno in zato neučinkovito. Poleg tega se navade uporabnikov neprestano spreminjajo. V določenem obdobju lahko npr. uporabniki začnejo prihajati domov kasneje; ob nespremenjenem urniku to pomeni, da se začne ogrevanje hiše prezgodaj, kar se odraža v večji porabi energentov in posledično v višjih obratovalnih stroških. Poleg tega se vedno bolj uveljavlja uporaba fotovoltaike kot komplementaren vir energije. Predpostavljamo, da lahko z inteligentnim kombiniranje različnih virov energije zmanjšamo stroške obratovanja in hkrati ohranimo zadovoljivo stopnjo udobja uporabnikov.

Za reševanje zgornjih nalog predlagamo sistem, ki je sposoben sprotnega spremljanja dogajanja v hiši, učenja navad uporabnikov, prilagajanja delovanja glede na spremembe v navadah uporabnikov in optimiziranja delovanja celotnega sistema hišne avtomatizacije.

Sistem OpUS, ki je bil razvit v okviru projekta e-storitve za gospodarstvo, je samostojna in robustna rešitev, ki se lahko vgradi v široko paleto obstoječih sistemov hišne avtomatizacije. Združuje več inovativnih komponent, kjer vsaka skrbi za določen vidik inteligentnega upravljanja pametne stavbe.

V nadaljevanju prispevka najprej opravimo kratek pregled sorodnega dela. Nadaljujemo s predstavitvijo arhitekture sistema OpUS in na kratko opišemo naloge posameznih komponent sistema.

2 SORODNI SISTEMI

Obširen pregled sodobnih sistemov vodenja v pametnih zgradbah je opisal Dounis et.al [1]. Za razliko od klasičnih načinov vodenja, je za optimalno, prediktivno ali adaptivno vodenje potrebno imeti model zgradbe. Temu pristopu sledimo v tem prispevku.

Vodenje sistemov z uporabo podatkov in znanj o uporabnikih in okolju predstavlja nove smernice raziskav in razvoja tako imenovanega vseprisotnega in prodornega računalništva (ang. ubiquitous computing, pervasive computing), saj sodobne naprave, senzorji in aktuatorjih, ki se vse bolj množično pojavljajo v zgradbah (senzorji prisotnosti, senzorji gibanja, senzorji odprtosti oken, senzorji na mobilnih telefonih, osebne vremenske postaje itd.) omogočajo beleženje najrazličnejših informacij in kopičenje znanj tako o obnašanju posameznega uporabnika, kot o obnašanju sistema. Uporaba takšnih znanj se izkorišča v sistemih, ki spodbujajo uporabnike k zmanjšanju porabe energije s spodbujanjem k na primer nižanju želenih temperatur ogrevanja ali pa k izbiri primernih prostorov v službi za potrebe sestankov (manj ljudi - manjši prostor - manj energije za ogrevanje) [3]. Znanje o uporabnikih se izkorišča za gradnjo modelov uporabnikovega obnašanja in uporabo le-teh pri vodenju in adaptaciji sistemov ogrevanja, razsvetljave, prezračevanja in ogrevanja sanitarne vode [4,5]. Prihranki energije se gibljejo med 5-30%.

Veliko projektov na temo izvedbe testnih pametnih hiš in stanovanj je bilo že dokončanih. Leta 1990 so izdelali Neural Network House [6], kjer so uporabljali nevronske mreže za inteligentno vodenje sistemov. Sledila sta IHome[7] in MavHome[8], temelječa na inteligentnem več-agentnem pristopu nadzora in vodenja sistemov z uporabo tehnik za modeliranje in napovedovanje uporabnikovega obnašanja in akcij. Gator Tech Smart House [9] je splošno uporaben študijski projekt za raziskavo tehnik vseprisotnega računalništva (ang. pervasive computing). Eden zadnjih projektov - ThinkHome [10] uporablja širok nabor podatkov o okolju, vremenu in uporabniku za namene študije vodenja pametnih domov.

Vsi sistemi se povečini osredotočajo zgolj na določene vidike upravljanja stavbe. ThinkHome, na primer, poskuša predvidevati temperaturo, ki bo za uporabnika najudobnejša. Poleg tega poskuša predvidevati, kdaj bo nek uporabnik prisoten. Sistem OpUS je obširnejši, saj poskuša modelirati večje število parametrov pametne hiše obenem pa v algoritme vodenja vključuje uporabniške navade in optimizacijske algoritme.

3 ARHITEKTURA SISTEMA

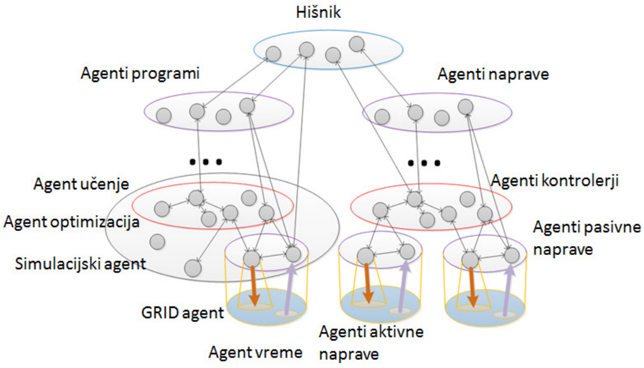
Arhitektura sistema OpUS je definirana s hierarhično urejenim več-agentnim sistemom [11]. Agenti so avtonomne entitete, ki so sposobne zaznavanja in interakcije z okolico skladno z njihovimi preferencami, lastnostmi in aktivnimi cilji. Sposobni so samostojnega razmišljanja in sodelovanja za dosego skupnih ciljev. Vsak agent v OpUS agentni arhitekturi je določen z agentno ovojnico, ki natančno definira posameznega agenta. Ovojnica določi vhodne podatke, ki jih agent zahteva, akcije, ki jih lahko izvaja in izhodne podatke, ki jih lahko posreduje.

3.1 Hierarhično urejena agentna arhitektura

Agentna arhitektura ni samo skupek agentov, ki določa hierarhične odnose med njimi, ampak skrbi za beleženje stanja vsakega agenta, določa način komunikacije med njimi in omogoča podporo za sprotno simuliranje dinamike v prototipnem okolju.

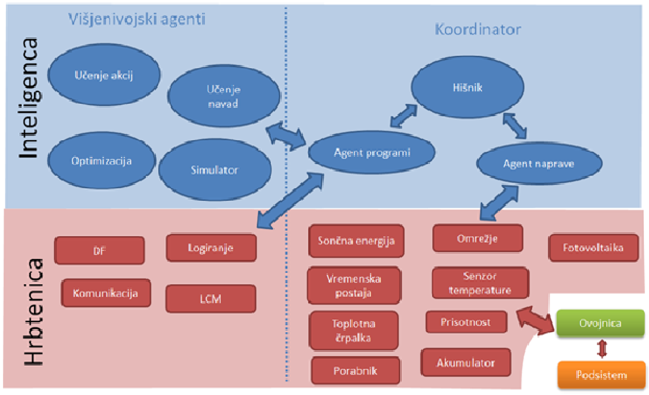
Sistem OpUS sestavlja šest različnih tipov agentov. Agent **pasivna naprava** je naprava, ki zgolj beleži in posreduje določen podatek. Primer takega agenta je senzor temperature zraka. Agent **aktivna naprava** je naprednejši in omogoča upravljanje z napravo. Primer takega agenta je toplotna črpalka, ki se ga lahko vklaplja, izklaplja in nastavlja na določeno temperaturo. Naslednji tip agentov je **program**, slednji združuje agente, ki ponujajo različne servise znotraj arhitekture. V to kategorijo spadajo agent dnevnik, ki beleži spremembe v sistemu, agent upravljanja sistema, agenti za učenje, agenti za optimizacijo, simulacijo ipd. Četrti tip agentov **hišnik** so upravljavski agenti, ki omogočajo nadzor in upravljanje določenih sklopov pametne stavbe. Odvisno od razdelitve in same hierarhične strukture, lahko ti agenti upravljajo sistem ogrevanja, določeno sobo ali pa celotno zgradbo. Agent hišnik, ki je v agentni hierarhiji postavljen najvišje, skrbi tako za koordinacijo med ostalimi višjenivojskimi agenti, kot tudi za komunikacijo z zunanjim svetom. Zgradba je namreč lahko vključena v pametno mesto, ki lahko od nje zahteva aktivno pogajanje o cenah energentov ipd.

Zadnja dva tipa agentov sta še podporni in komunikacijski agenti, ki so lastni arhitekturi in skrbijo za prenašanje sporočil med agenti, beleženje posameznih agentov, posredovanje podatkov, ontologij ipd. Simbolična shema arhitekture je prikazana na sliki 2. Sama arhitektura je zasnovana tako, da je omogočeno enostavno dodajanje in odstranjevanje agentov v sistem.



Slika 1: *Hierarhična več-agentna arhitektura.*

Sistem je logično razdeljen na dva dela, kot je prikazano na sliki 3. Zgornji nivo vsebuje tako imenovano inteligenco, torej agente, ki skrbijo za generiranje novih, za uporabnika ustreznejših urnikov. Spodnji nivo (t. i., hrbtenica) vsebuje agente vodenja, ki skrbijo za izvajanje urnikov in sprotno sinhronizacijo z realnim okoljem. Pomembna predpostavka v sistemu je ta, da lahko sistem ob morebitnem izpadu zgornjega dela še vedno nemoteno deluje in skrbi za upravljanje z napravami v zgradbi.



Slika 2: *Povezovanje agentov znotraj arhitekture.*

3.2 Učenje navad uporabnika

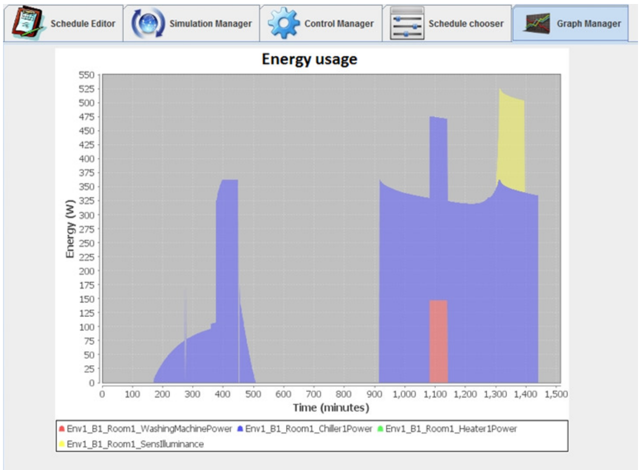
V sistem OpUS je vključen modul za učenje navad uporabnika. Slednji poskuša zgraditi model obnašanja uporabnika na podlagi opazovanja prisotnosti v hiši in v posameznih sobah. Zgrajeni model lahko za določeno obdobje napove verjetnost, da je uporabnik prisoten ali odsoten. Napoved prisotnosti uporabnika omogoči celo vrsto dodatnih storitev, ki so v sodobnem sistemu hišne avtomatizacije nujne. Natančna napoved časa odhoda uporabnika omogoča varčevanje z energijo, saj lahko sistem predčasno izklopi ogrevanje. Podobno lahko natančna napoved časa prihoda omogoči povečanje občutka udobja pri uporabnikih.

Gradnja modela poteka tako, da sistem dlje časa beleži senzorske podatke o prisotnosti in sproti gradi verjetnostni model navad uporabnika. Prednost sprotne gradnje je v tem, da omogoča prilagajanje na spremembe v navadah uporabnika. Sistem nato dnevno uporabi zgrajeni model za napoved časa odhoda in prihoda uporabnika. S tem optimizacijskemu agentu zmanjša preiskovalni prostor, saj določi intervale, kjer z večjo verjetnostjo pride do sprememb v režimu delovanja.

3.3 Model in simulacija zgradbe

Sposobnost natančnega simuliranja realnega okolja omogoča predhodno verifikacijo strategij vodenja in izbiro tiste, ki vodi do najustreznejšega stanja sistema. V sistemu OpUS je simulacija ključna za poganjanje optimizacijskih algoritmov in sprotnega izvajanja dinamike sistema.

Razviti simulacijski agent uporablja orodje EnergyPlus[13], ki omogoča simuliranje poljubnih modelov stavb in različnih naprav. Ključna naloga je zgraditi model stavbe, ki kar se da natančno odraža realno prototipno okolje. Orodje omogoča definiranje naprav s poljubnimi specifikacijami, omogoča integracijo fotovoltaike in akumulatorjev. Za čim bolj natančno izvajanje simulacije uporablja informacije o geografski lokaciji modela, realne informacije o vremenu in sončnem obsevanju. Simulacijski agent omogoča sprotno spreminjanje parametrov delovanja in izpis stanja naprav. Na sliki 3 je prikazana zaslonska maska pregleda porabe posameznih naprav znotraj simulacije.



Slika 3: *Prikaz posamezne porabe tekom simulacije.*

3.4 Optimizacija delovanja

Sistem OpUS uporablja optimizacijo za izračun novih urnikov, ki so za uporabnika ustreznejši in omogočajo manjše stroške in večje udobje. Z uporabo simulatorja stavbe, ki za podani urnik vrača stroške in udobje, večkriterijski optimizacijski algoritem poišče množico urnikov, ki predstavljajo kompromisne rešitve glede na obravnavana nasprotujoča si kriterija. Dobljene rešitve predstavljajo približek za t.i. Pareto fronto [12], na osnovi katere nato uporabnik preko uporabniškega vmesnika izbere nov, zanj najustreznejši urnik.

Algoritem gradi množico rešitev iz začetnega urnika, s preiskovanjem celotnega prostora urnikov in uporabo simulatorja za ovrednotenje generirane rešitve iz vidika porabe in udobja. Tekom izvajanja se generirane rešitve izboljšujejo in najboljše rešitve z vidika obeh kriterijev tvorijo Pareto fronto nedominiranih urnikov. Uporabnik lahko preko uporabniškega vmesnika izbere enega od teh urnikov glede na preference o načinu vodenja stavbe. Za izvajanje izbranega urnika poskrbi agentna arhitektura, ki sočasno upravlja simulirano in realno prototipno okolje.

3.5 Komunikacijski vmesnik

Eden pomembnih modulov v arhitekturi je komunikacijski vmesnik, ki skrbi za sinhronizacijo agentne arhitekture z realnim kontejnerskim prototipom. OpUS preko HTTP zahtevkov pridobiva informacije o stanju naprav priključenih na kontrolerje.

Primer zahtevka za branje temperature v prostoru:

```
http://www.exp.com/scgi/?c11025.ts01_temperature_0
```

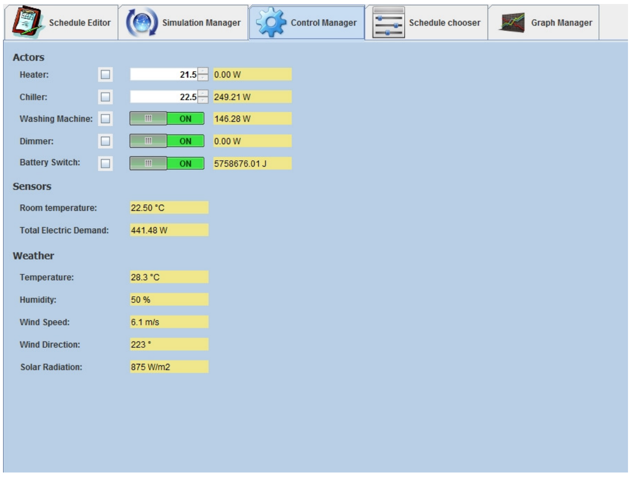
Pri čemer je »c11025« CyBro krmilnik z naslovom 11025, ts01_temperature_o pa je ime spremenljivke v krmilniku.

Strežnik vrne odgovor v formatu XML:

```
<data>
<var>
<name>c11025.ts01_temperature_0</name>
<value>268</value>
<description>
Measured internal temperature, multiplied
by 10 (e.g. 247 means 24.7°C).
</description>
</var>
</data>
```

Dobljene vrednosti se uporabljajo za sinhronizacijo agentne arhitekture z dogajanjem v realnem okolju, tako da se vrednosti v XML odgovoru prepišejo v ustreznega agenta-napravo.

Vodenje sistema poteka na podoben način preko http zahtevkov. Na podlagi vrednosti v izbranem urniku pošlje sistem zahtevek za spremembo delovanja naprave. Prednost takega pristopa je decentraliziranost, saj ni potrebno, da sta sistem OpUS in krmilnik locirana na istem strežniku. S tem tudi ločimo inteligentni del sistema od logičnega dela, ki skrbi zgolj za izvajanje ukazov na krmilniku.



Actors

Heater:	□	21.5	0.00 W
Chiller:	□	22.5	249.21 W
Washing Machine:	□	ON	146.28 W
Dimmer:	□	ON	0.00 W
Battery Switch:	□	ON	5758676.01 J

Sensors

Room temperature: 22.50 °C

Total Electric Demand: 441.48 W

Weather

Temperature: 28.3 °C

Humidity: 50 %

Wind Speed: 6.1 m/s

Wind Direction: 223 °

Solar Radiation: 875 W/m2

Slika 4: *Upravljanje z agenti aktuatorji.*

Na sliki 4 je prikazan vmesnik za upravljanje z napravami v sistemu OpUS in hkrati, preko sinhronizacijskega mehanizma, v realnem prototipnem okolju. Na zaslonski maski je prikazana trenutna poraba naprav ter celotnega sistema, prikazani so podatki iz senzorjev, stanje baterije in trenutna vremenska napoved. Posamezne naprave je mogoče priklapljati in izklapljati ter nastavljati parametre delovanja.

4 ZAKLJUČEK

V pričujočem prispevku smo opisali celotno arhitekturo sistema OpUS in opisali glavne module sistema. Uporaba več-agentne paradigme olajša razvoj kompleksnih sistemov, kjer je pomembna sprotna komunikacija in učinkovito sodelovanje med komponentami.

Vseh pet glavnih agentov sistema je načrtovanih tako, da lahko uporabljajo storitve drugih agentov in hkrati ponudijo na razpolago svoje funkcionalnosti. Tipičen primer je simulacijski agent, ki uporablja storitve vmesnika ter podatke v agentni arhitekturi in ponuja storitve simulacije optimizacijskemu agentu ter omogoča sprotno izvajanje s strani agente arhitekture.

Takšen dinamičen in inteligenten sistem lahko znatno zniža porabo in s tem stroške inteligentne hiše. S tem pa vzpodbuja ekološko ozaveščenost uporabnikov in zmanjšuje negativne vplive na okolje.

Obstoječi tržni sistemi hišne avtomatizacije predstavljenih rešitev ne uporabljajo. Deloma zaradi višjih stroškov razvoja, kar se direktno preslika v ceno takega sistema, deloma pa zaradi trenutno nezanesljivega in nerobustnega delovanja. Sistem OpUS se uspešno spopada z navedenimi težavami in z uporabniškega stališča omogoča prijazno uporabo in veliko mero avtonomnosti.

Reference

[1] Dounis, Anastasios I., and Christos Caraiscos. "Advanced control systems engineering for energy and comfort management in a building environment—A review." Renewable and Sustainable Energy Reviews 13.6 (2009): 1246-1261.

[2] R. Baños, F. Manzano-Agugliaro, F.G. Montoya, C. Gil, A. Alcayde, J. Gómez, Optimization methods applied to renewable and sustainable energy: A review, Renewable and Sustainable Energy Reviews, Volume 15, Issue 4, May 2011, Pages 1753-1766, ISSN 1364-0321, http://dx.doi.org/10.1016/j.rser.2010.12.008

[3] Laura Klein, Jun-young Kwak, Geoffrey Kavulya, Farrokh Jazizadeh, Burcin Becerik-Gerber, Pradeep Varakantham, Milind Tambe, Coordinating occupant behavior for building energy and comfort management using multi-agent systems, Automation in Construction, Volume 22, March 2012, Pages 525-536, ISSN 0926-5805, http://dx.doi.org/10.1016/j.autcon.2011.11.012

[4] Lu, Jiakang, et al. "The smart thermostat: using occupancy sensors to save energy in homes." Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems. ACM, 2010.

[5] Agarwal, Yuvraj, et al. "Occupancy-driven energy management for smart building automation." Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building. ACM, 2010.

[6] M. C. Mozer, The neural network house: An environment hat adapts to its inhabitants, Proc AAAI Spring Symp Intelligent Environments. pp. 110–114, (1998)

[7] V. Lesser, M. Atighetchi, B. Benyo, B. Horling, V. L. M. Atighetchi, A. Raja, R. Vincent, P. Xuan, S. X. Zhang, T. Wagner, P. Xuan, and S. X. Zhang. The intelligent home testbed. In Proceedings of the Autonomy Control Software Workshop, (1999).

[8] D. Cook,M. Youngblood, I. Heierman, E.O., K. Gopalratnam, S. Rao, A. Litvin, and F. Khawaja. Mavhome: an agent-based smart home. In Pervasive Computing and Communications, 2003. (PerCom 2003). Proceedings of the First IEEE International Conference on, pp. 521 – 524 (march, 2003). doi: 10.1109/PERCOM.2003.1192783.

[9] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, and E. Jansen, The gator tech smart house: a programmable pervasive space, Computer. 38(3), 50 – 60 (march, 2005). ISSN 0018-9162. doi: 10.1109/MC.2005.107.

[10] C. Reinisch, M. J. Kofler, F. Iglesias, and W. Kastner, Thinkhome energy efficiency in future smart homes, EURASIP J. Embedded Syst. 2011, 1:1–1:18 (Jan., 2011). ISSN 1687-3955. doi: 10.1155/2011/104617

[11] Multi-Agent system:
https://en.wikipedia.org/wiki/Multi-agent_system

[12] Kalyanmoy Deb. Multi-Objective Optimization Using Evolutionary Algorithms. Wiley, Nw York, 2001.

[13] EnergyPlus.
http://apps1.eere.energy.gov/buildings/energyplus/energyplus_about.cfm

PREDICTIVE PROCESS-BASED MODELING OF AQUATIC ECOSYSTEMS

Nina Vidmar[1], Nikola Simidjievski[2,3], Sašo Džeroski[2,3]
Faculty of Mathematics and Physics, University of Ljubljana, Ljubljana, Slovenia[1]
Jožef Sefan Institute, Ljubljana, Slovenia[2]
Jožef Stefan International Postgraduate School, Ljubljana, Slovenia[3]
e-mail: nina.vidmar@student.fmf.uni-lj.si, {nikola.simidjievski, saso.dzeroski}@ijs.si

ABSTRACT

In this paper, we consider the task of learning interpretable process-based models of dynamic systems. While most case studies have focused on the descriptive aspect of such models, we focus on the predictive aspect. We use multi-year data, considering it as a single consecutive dataset or as several one-year datasets. Additionally, we also investigate the effect of interpolation of sparse data on the learning process. We evaluate and then compare the considered approaches on the task of predictive modeling of phytoplankton dynamics in Lake Zürich.

1 INTRODUCTION

Mathematical models play an important role in the task of describing the structure and predicting the behavior of an arbitrary dynamic system. In essence, a model of a dynamic system consists of two components: a structure and a set of parameters. There are two basic approaches to constructing models of dynamic systems, i.e., theoretical (knowledge-driven) modeling and empirical (data-driven) modeling. In the first, the model structure is derived by domain experts of the system at hand, the parameters of which are calibrated using measured data. In contrast, the later uses measured data to find the most adequate structure-parameter combination that best fits the given task of modeling. In both approaches, models often take the form of ordinary differential equations (ODEs), a widely accepted formalism for modeling dynamic systems, allowing the behavior of the system to be simulated over time.

Equation discovery [1, 2] is the area of machine learning dealing with developing methods for automated discovery of quantitative laws, expressed in the form of equations, from collections of measured data. The state-of-the-art equation discovery paradigm, referred to as process-based modeling [3], integrates both theoretical and empirical approaches to modeling dynamics. The result is a process-based model (PBM) – an accurate and understandable representation of a dynamic systems.

The process-based modeling paradigm has already been proven successful for modeling population dynamics in a number of aquatic ecosystems, such as: lake ecosystems [4, 5, 6, and 7] and marine ecosystems [3]. However, these studies focus on obtaining explanatory models of the aquatic ecosystem, i.e., modeling the measured behavior of the system at hand, while modeling future behavior is not considered. In contrast, Whigham and Recknagel [8] discuss the predictive performance of process-based models in a lake ecosystem. However, either they assume a single model structure and focus on the task of parameter identification, or explore different model structures where the explanatory aspect of the model is completely ignored. The method proposed by Bridewell et.al [9] focuses of establishing robust interpretable process-based models, by tackling the over-fitting problem. Even though this method provides estimates of model error on unseen data, these estimates are not related to the predictive performance of the model, i.e., its ability to predict future system behavior beyond the time-period captured in training data. Most recently, the study of Simidjievski et.al [10] focuses on the predictive performance of process-based models by using ensemble methods. However, while their proposed ensemble methods improve the predictive performance of the process-based models, the resulting ensemble model is not interpretable.

In this paper we tackle the task of establishing an interpretable predictive model of a dynamic system. We focus on predicting the concentration of phytoplankton biomass in aquatic ecosystems. Due to the high dynamicity and various seasonal exogenous influences [6, 7], most often process-based models of such systems are learned using short time-periods of observed data (1 year at most). Note however, this short time-periods of data are very sparse, i.e., consist of very few measured values, thus, most often the measurements are interpolated and daily samples are obtained from the interpolation.

The initial experiments to this end, indicate that the predictive performance of such models is poor: While providing dense and accurate description of the observed behavior, they fail at predicting future system behavior. To address this limitation we propose learning more robust process-based models. We conjecture that by increasing the size of the learning data, more general process-base models will be obtained, thus yielding better predictive performance while maintaining their interpretability.

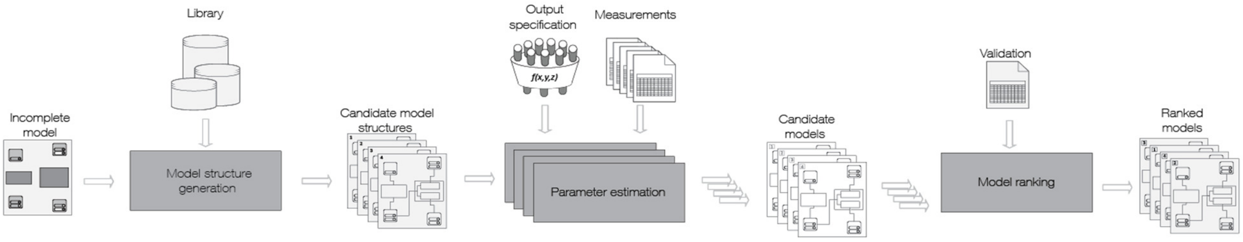


Figure 1: Automated modeling with ProBMoT.

The main contribution of this paper are the approaches to handling the learning data. The intuitive way of increasing the size of the learning data is by sequentially adding predeceasing contiguous datasets, thus creating one long time-period dataset, i.e., learning from sequential data (LSD). In contrast, when learning from parallel data (LPD), the model is learned from all the datasets simultaneously. Figure 2 depicts the both approaches. The two approaches, in terms of learning process-based models, are described in more detail in Section 3.

We test the utility of the two approaches on a series of tasks of modeling phytoplankton concentration in Lake Zürich. We use eight yearly datasets, using six for training, one for validation and one for testing the predictive performance of the obtained models. The aim of this paper is two-fold: besides validating the performance of the two approaches to handling data when learning predictive process-based models, we also test the quality of the training data. For that purpose, we perform additional set of experiments, similar to the previous. However, instead of using the interpolated data for learning the models – we use the original (sparse) measured values, thus examining the influence of the interpolation on the predictive performance of the process-based models.

The next section provides more details of the task of process-based modeling, and introduces a recent contribution to the area of automated process-based modeling, i.e., the ProBMoT [4, 10] platform. Section 3 depicts the task of predictive process-based modeling of aquatic ecosystems. Section 4 describes the data used in the experiments, the design of the experiments, and the task specification. Section 5 presents the results of the experiments. Finally, Section 6 discusses the findings of this paper and suggests directions for future work.

2 PROCESS-BASED MODELING AND PROBMOT

The process-based modeling paradigm, addresses the task of learning process-based models of dynamic systems from two points of view: qualitative and quantitative. The first, provides a conceptual representation of the structure of the modeled system. Still, this depiction does not provide enough details that would allow for simulation of the system's behavior. In contrast, the later, treats the process-based model as a set of differential and/or algebraic equations which allows for simulation.

A process-based model consists of two basic types of elements: entities and processes. Entities correspond to the state of the system. They incorporate the variables and the constants related to the components of the modeled system. Each variable in the entity has its role. The role specifies whether the variable is exogenous or endogenous. Exogenous variables are explanatory/input variables, used as forcing terms of the dynamics of the observed system (and are not modeled within the system). Endogenous variables, are the response/output (system) variables. They represent the internal state of the system and are the ones being modeled. The entities are involved in complex interactions represented by the processes. The processes include specifications of the entities that interact, how those entities interact (equations), and additional sub-processes.

From the qualitative perspective, the unity of entity and processes allows for conceptual interpretation of the modeled system. On the other hand, the entities and the processes provide further modeling details that allow for transformation from conceptual model to equations and therefore simulation of the system, i.e., providing the quantitative abilities of the process-based model. The equations define the interactions represented by the processes including the variables and constants from the entities involved.

The process-based modeling paradigm allows for high-level representation of domain-specific modeling knowledge. Such knowledge is embodied in a library of entity and process templates, which represent generalized modeling blueprints. The entity and process templates are further instantiated in specific entities and processes that correspond to the components and the interactions of the modeled system. These specific model components and interactions define the set of candidate model structures.

The algorithm for inducing models employs knowledge-based methods to enumerate all candidate structures. For each obtained structure, a parameter estimation is performed using the available training data. For this reason each structure is compiled into a system of differential and algebraic equations, which allows for the model to be simulated. In essence, this includes minimizing the discrepancy between the values of the simulated behavior obtained using the model and the observed behavior of the system.

Recent implementations of the PBM approach include Lagrame2.0 [11], HIPM [12] and ProBMoT (Process-Based Modeling Tool) [4, 10], which is next described.

The Process-Based Modeling Tool (ProBMoT), is a software platform for simulating, parameter fitting and inducing process-based models. Figure 1 illustrates the process of automated modeling with ProBMoT. The first input to ProBMoT is a conceptual model of the modeled system. The conceptual model specifies the expected logical structure of the modeled system in terms of entities and processes that we observe in the system at hand. The second input is the library of domain-specific modeling knowledge. By combining the conceptual model with the library of plausible modeling choices, candidate model structures are obtained.

The model parameters for each structure are estimated using the available training data (third input to ProBMoT). The parameter optimization method is based on meta-heuristic optimization framework jMetal 4.5 [13], in particular, ProBMoT implements the Differential Evolution (DE) [14] optimization algorithm. For the purpose of simulation, each model is transformed to a system of ODEs, which are solved using CVODE ODE solver from the SUNDIALS suite [15].

Finally, the last input, is a separate validation dataset. In both cases (LSD and LPD), the model which has best performance on the validation dataset is the output of automated modeling process.

3 PREDICTIVE PROCESS-BASED MODELING OF AQUATIC ECOSYSTEMS

ProBMoT has been used extensively to model aquatic ecosystems [4, 5 and 6]. Most of the case-studies, however, have focused on descriptive modeling – focusing on the content/interpretation of the learned models and not on their accuracy and predictive performance (with the exception of [10]). Predominately, models have been learned from short time-period (one-year) datasets, as considered long time-periods worth of data resulted in models of poor fit. These models, however, had poor predictive power when applied to new (unseen) data.

We use ProBMoT to learn predictive models of aquatic ecosystems from long time-period (multi-year) datasets. ProBMoT supports predictive modeling, as the obtained models can be applied/evaluated on a testing dataset. Taking the input/exogenous variable values from the test dataset, ProBMoT simulates the model at hand, and makes predictions for the values of the output/endogenous (system) variables. Using the output specifications, the values of the output variables of the model are calculated and compared to the output variables from the test set, thus allowing for the predictive performance of the model to be assessed.

Concerning the use of long time-period datasets, ProBMoT supports two different approaches, i.e., learning from sequential data (LSD) and learning from parallel data (LPD). The parameter optimization algorithm uses the available training data from the observed system to estimate the numerical values of the parameters. When learning from sequential data, illustrated in Figure 2a, ProBMoT takes as an input one training dataset. The training dataset is comprised of several contiguous short time-period datasets, thus the parameters are estimated over the whole time-span.

One the other hand, when learning from parallel data, depicted in Figure 2b, ProBMoT takes as an input several short time-period training datasets. The parameter optimization algorithm handles the short time-periods in parallel, i.e., it estimates the optimal model parameters by minimizing the discrepancy between the simulated behavior and each individual training set.

ProBMoT offers wide range of objective functions for measuring model performance such as sum of squared errors (SSE) between the simulated values and observed data, mean squared error (MSE), root mean squared error (RMSE), relative root mean squared error (ReRMSE), which is used in all experiments presented here for when learning the models and evaluating their performance. Relative root mean squared error (*ReRMSE*) [16] is defined as:

$$ReRMSE(m) = \sqrt{\frac{\sum_{t=0}^{n}(y_t - \hat{y}_t)^2}{\sum_{t=0}^{n}(\bar{y} - \hat{y}_t)^2}}, \qquad (1)$$

where n denotes the number of measurements in the test data set, $\hat{y}_t$ and y_t correspond to the measured and predicted value (obtained by simulating the model m) of the system variable y at time point t, and $\bar{y}$ denotes the mean value of the system variable y in the test data set.

The data on the aquatic systems are very sparse (e.g. measure on a monthly basis). In the above mentioned studies, often they have been typically interpolated and sampled at a daily interval. Here, to assess the effect of the interpolation to the performance of the models, we also consider using only the original measured values when establishing the predictive process-based model.

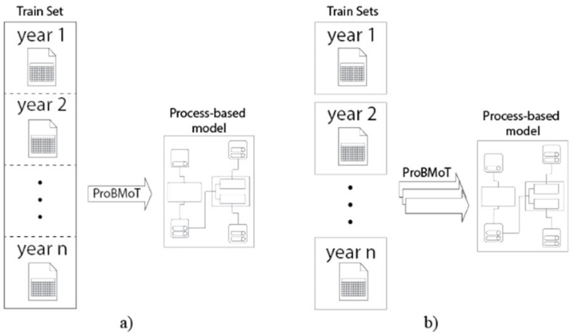


Figure 2: Two approaches to predictive modeling. a) Learning from sequential data (LSD), and b) Learning from parallel data (LPD).

4 EXPERIMENTAL SETUP

In this study, we apply the automated modeling tool ProBMoT to the task of predictive modeling of phytoplankton dynamics in Lake Zürich, Switzerland. We empirically evaluate the two different approaches for learning predictive models, LSD and LPD, on this task. We apply those two on interpolated data (sampled daily) and on the original (sparse) data.

4.1 Data & domain knowledge

The datasets used for our experiments were obtained from the Water Supply Authority of Zürich. Lake Zürich is a lake in Switzerland, extending southeast of the city of Zürich. It has an average depth of 49 m, a volume of 3.9 km^3 and a surface area of 88.66 km^2. The measurements consist of physical, chemical and biological data for the period from 1992 to 1999, taken once a month at 19 different sites, and averaged to the respective epilimnion (upper ten meters) and hypilimnion (bottom ten meters) depths.

The data were interpolated with a cubic spline algorithm and daily samples were taken from the interpolation [17]. Both the original and interpolated data from the first six years were used for training the models (1992-1997), data from year 1998 for validation and data from 1999 to estimate the predictive performance of the learned models.

The population dynamics model considered, consists of one endogenous/output (system) variable and multiple exogenous/input variables structured within a single ODE. The phytoplankton biomass is represented as a system variable, while the exogenous variables include: the concentration of zooplankton, dissolved inorganic nutrients (nitrogen, phosphorus, and silica) and two environmental influences of water temperature and global solar radiation (light).

The library for process-based modeling of aquatic ecosystems used in our experiments, is the one presented by Atanasova [18]. Particularly, to reduce the computational complexity of our experiments, we use a simplified version of the library which results in total of 128 candidate models.

4.2 ProBMoT parameter settings

For the parameter calibration procedure we use Differential Evolution with rand/1/bin strategy, 1000 evaluations over a population space of 50 individuals. For simulating the ODEs we use the CVODE simulator with absolute and relative tolerances set to 10^{-3}. For measuring the model performance we use objective function *ReRMSE,* described in Section 3. To further assess the significance of the differences in performance between the single dataset approach and multiple datasets approach we use Wilcoxon test for statistical significance [19] as presented by Demšar [20].

4.3 Experimental design

In this paper we compare the performance of the two different approaches (LSD and LSP) to learning predictive process-based models. For each approach we learn six process-based models using the available training data of six successive years (1992-1997). For both cases, we start with one short time-period training dataset (year 1997), and continue for five steps adding one preceding year to the training data set. At each step we learn the process-based models accordingly to the two approaches described in the previous section.

First, we apply this two approaches on the interpolated data, or more precisely, daily samples of interpolated data. Second, we apply the two learning approaches to the original (sparse) training data. In all of the experiments the validation dataset (year 1998) and the test dataset (year 1999) remain the same.

5 RESULTS

Table 1 summarizes the performance comparison between models learned from sequential data (LSD) and models learned from parallel data (LPD), using both interpolated (left-hand side) and original (right-hand side) training data. Note that, in both cases, learning from sequential data, yields better predictive performance than learning from parallel data. The results of the Wilcoxon test (in Table 1 below) shows that using LSD is better than using LPD, however, the difference in performance is not substantial nor significant (p-value=0.11).

Table 1: Comparison of the predictive performances (ReRMSE on test data) of models learned from sequential data (LSD) and models learned from parallel data (LPD), from both interpolated and original samples. The numbers in bold indicate the best result for the given years.

Train data (years)	Interpolated		Original	
	LSD	LPD	LSD	LPD
‘97	**1.398**	**1.398**	1.074	1.074
’96-’97	**1.099**	1.391	1.381	1.469
’95-’97	1.006	1.044	**0.984**	1.084
’94-’97	**0.986**	1.094	1.004	1.112
’93-’97	**1.075**	1.109	1.105	1.085
’92-’97	**0.934**	0.998	1.074	0.974

Wilcoxon test	LSD > LPD;
	p-value = 0.11
			LSD > LPD;
			p-value = 0.11	

Next, as shown in Table 1, using the original measured values when learning the models, did not improve their predictive performance.

Finally, most importantly, from both experiments performed, we can conclude that using large amounts of training data (even interpolated) improves the overall predictive performance of the learned process-based models. Note however, that for one case (’93-97) the performance of the models does not improve. Further investigations are required to determine whether this phenomena is due to the quality of the data of that particular dataset (‘93), or to the dynamics of the system at that particular period significantly differing from the rest.

6 CONCLUSION

In this paper, we tackle the task of learning predictive interpretable process-based models of dynamic systems. In the process of establishing general and robust predictive models, we investigate learning from parallel data (LPD), in contrast to the state-of-the-art approach of learning from sequential data (LSD). We apply the both approaches to the task of modeling phytoplankton dynamics in Lake Zürich, using ProBMoT, a platform for simulating, parameter fitting and inducing process-based models. Additionally, besides validating the performance of the approaches to learning predictive process-based models, we also test the quality of the training data by learning models from the original measured values, in contrast to learning models from daily samples of interpolated data.

The general conclusion of this paper is that using larger amounts of training data for learning process-based models yields improved predictive performance for tasks of modeling aquatic ecosystems. Both, Atanasova et al [5] and Taškova et al. [6] clearly state that one-year datasets produce models with poor predictive performance. We show that using data from a longer period, considered either consequently (LSD) or parallel (LPD) helps in deriving more general models, and therefore, better predictive models.

Even though the statistical significance comparison shows that the LSD approach has better performance than the LPD approach, the difference in performance is neither substantial nor significant. Nevertheless, when learning from sequential data, due to the mater of simulation and parameter optimization, the available training data considered for learning process-based models should be contiguous. On the other hand, one useful feature of the LPD approach is that can handle missing data (e.g. intermediate period with no measurements) for establishing robust process-based models.

Our empirical evaluation of learning from the original uninterpolated and sampled interpolated data, showed that the interpolation does not affect the performance of the learned process-based models. On the contrary, the models learned using the interpolated values yielded better performance than the ones learned using the original values. We conjecture that this is due to the sparsity of the original measured values (~ 12 time-points per year), which is insufficient to capture the dynamics of such a system. Moreover, considering the relative performance between the two approaches, the LSD approach performed insignificantly better than the LPD approach

Taken all together, some new questions arise for further investigation. How strongly the quality of measurements affects the results? Would the results change significantly in the case of ideal measurements? Considering this, possible directions for further work are as follows. First, performing more experiments using multiple parallel sets of data from different periods and, data from various different lake ecosystems should be used. In order to achieve more controlled experiments, we consider testing the presented approaches on synthetic data, that is, data obtained by simulating a well-established model of an arbitrary aquatic ecosystem. Finally, we would like to extend our approach to different ecosystems and other domains.

Acknowledgements

We would also like to acknowledge the support of the European Commission through the project MAESTRA - Learning from Massive, Incompletely annotated, and Structured Data (grant number ICT-2013-612944).

References:

[1] P. W. Langley, H. A. Simon, G. Bradshaw, J. M. Zytkow. Scientific Discovery: Computational Explorations of the Creative Processes. MA: The MIT Press, Cambridge, MA, USA. 1987.

[2] S. Džeroski, L. Todorovski. Learning population dynamics models from data and domain knowledge. *Ecological Modelling 170*, pp. 129–140. 2003.

[3] W. Bridewell, P. W. Langley, L. Todorovski, S. Džeroski. Inductive process modeling. *Machine Learning 71*, pp. 1–32. 2008.

[4] D. Čerepnalkoski, K. Taškova, L. Todorovski, N. Atanasova, S. Džeroski. The influence of parameter fitting methods on model structure selection in automated modeling of aquatic ecosystems. *Ecological Modelling 245* (0), pp. 136–165. 2012.

[5] K. Taškova, J. Šilc, N. Atanasova, S. Džeroski. Parameter estimation in a nonlinear dynamic model of an aquatic ecosystem with meta-heuristic optimization. *Ecological Modelling 226*, pp. 36–61. 2012.

[6] N. Atanasova, F. Recknagel, L. Todorovski, S. Džeroski, B. Kompare. Computational assemblage of Ordinary Differential Equations for Chlorophyll-a using a lake process equation library and measured data of Lake Kasumigaura. In: Recknagel, F.(Ed.), Ecological Informatics. Springer, pp. 409–427. 2006a.

[7] N. Atanasova, L. Todorovski, S. Džeroski, R. Remec, F. Recknagel, B. Kompare. Automated modelling of a food web in Lake Bled using measured data and a library of domain knowledge. *Ecological Modelling 194* (1-3), pp. 37–48. 2006c.

[8] P. Whigham, F. Recknagel, F. Predicting Chlorophyll-a in freshwater lakes by hybridising process-based models and genetic algorithms. *Ecological Modelling 146* (13), pp. 243–251. 2001.

[9] W. Bridewell, N. B. Asadi, P. Langley, L. Todorovski. Reducing overfitting in process model induction. In: Proceedings of the 22nd International Conference on Machine learning. (ICML '05). ACM, pp. 81–88. 2005.

[10] N. Simidjievski, L. Todorovski, S. Džeroski. Learning ensembles of population dynamics models and their application to modelling aquatic ecosystems. *Ecological Modelling (In Press)*. 2014.

[11] L. Todorovski, S. Džeroski. Integrating domain knowledge in equation discovery. In: Džeroski, S., Todorovski, L. (Eds.), Computational Discovery of Scientific Knowledge. Vol. 4660 of Lecture Notes in Computer Science. Springer Berlin, pp. 69–97. 2007.

[12] L. Todorovski, W. Bridewell, O. Shiran, P. W. Langley. Inducing hierarchical process models in dynamic domains. In: Proceedings of the 20th National Conference on Artificial Intelligence. AAAI Press, Pittsburgh, USA, pp. 892–897. 2005.

[13] J. J. Durillo, A. J. Nebro. jMetal: A Java framework for multi-objective optimization. *Advances in Engineering Software 42*, pp. 760–771. 2011.

[14] R. Storn, K. Price. Differential Evolution – A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization 11* (4), pp. 341–359. 1997.

[15] S. D. Cohen, A. C. Hindmarsh. CVODE, a stiff/nonstiff ODE solver in C. Computers in Physics 10 (2), pp. 138–143. Mar. 1996.

[16] L. Breiman. Classification and Regression Trees. Chapman & Hall, London, UK. 1984.

[17] A. Dietzel, J. Mieleitner, S. Kardaetz, P. Reichert. Effects of changes in the driving forces on water quality and plankton dynamics in three swiss lakes long-term simulations with BELAMO. *Freshwater Biology 58* (1), pp. 10–35. 2013.

[18] N. Atanasova., L. Todorovski, S. Džeroski, B. Kompare. Constructing a library of domain knowledge for automated modelling of aquatic ecosystems. *Ecological Modelling 194* (13), pp. 14–36. 2006b.

[19] F. Wilcoxon. Individual comparisons by ranking methods. Biometrics, 1:80–83. 1945.

[20] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research 7*, pp. 1–30. Dec. 2006.

RECOGNITION OF BUMBLEBEE SPECIES BY THEIR BUZZING SOUND

Mukhiddin Yusupov[1]*, Mitja Luštrek*[2]*, Janez Grad, Matjaž Gams*[2]
Czech Technical University in Prague – Computer Science Department[1]
Jozef Stefan Institute – Intelligent Systems Department[2]
e-mail: yusupmuk@fel.cvut.cz, {mitja.lustrek, matjaz.gams}@ijs.si, janez.grad@siol.com

ABSTRACT

The goal of our work is to help people to automatically identify the species and worker/queen type of bumblebee based on their recorded buzzing. Many recent studies of insect and bird classification based on their sound have been published, but there is no thorough study that deals with the complex nature of buzzing sound characteristic of bumblebees. In this paper, a database of recorded buzzings of eleven species were preprocessed and segmented into a series of sound samples. Then we applied J48, MLP and SVM supervised classification algorithms on some predetermined sets of feature vectors. For five species the recognition rate was above 80% and for other six species it was above 60%. At the end we consider how to further improve the results.

1 INTRODUCTION

Bumblebees are important pollinators of many plants and their colonies are now used extensively in greenhouse pollination of crops such as tomatoes and strawberries. Some bumblebee species are declining and it is a cause for concern in Europe, North America and Asia. In Europe, around one quarter of species are threatened with extinction, according to recent studies. This is due to a number of factors, including land clearing and agricultural practices. There is a lot of research devoted to keep some bumblebee species from such decline.

Until now over 250 species are known. There are about 19 different species of bumblebee found in the UK, 68 in Europe, 124 species in China, 24 in South America and 35 in Slovenia. The colonies of bumblebees are composed of a queen and many workers. Since only experts can identify the species by looking at or listening to them and their sound, we decided to make this identification easy for all. One needs to record the buzzing and provide it to the system (program) that will process it and then tell to which species and worker/queen type this buzz corresponds to. The program is accessible from the homepage of the Department of intelligent systems at the Jozef Stefan Institute - http://dis.ijs.si/. More information can be provided from janez.grad@siol.com.

Although there are generalizations of the type of problem we are solving here, such as a system for classifying many types of insects [4], relatively little has been done previously on automatic recognition on bumblebee species with a detailed analysis of their buzzing sound. Several internet applications provide sounds and images and images of different species of birds, frogs etc., but they rely on human pattern-recognition skills to identify the species at hand and do not provide active help. Our study is related to active system help in recognizing a particular (sub)species, and in particular to other audio data classification problems like classification of general audio content [8], auditory scene recognition, music genre classification and also to the speech recognition, which have been studied relatively extensively during last few years also in our department. We here try to take advantage from these previous studies.

Some studies like [7], where they also tried to classify bee species, used different approaches. We can view these systems as solving a pattern recognition problem. In [7] the recognition of bee species is performed visually, based on its image. The task was to find relevant patterns from the image and identify similarities to specific species. However, pictures vary a lot based on different factors, and often a picture does not represent well what we see in nature. In our work the patterns are buzzing sound events produced by bumblebees. The chosen approach is recognition based on a parametric (feature) representation of the sound events. Features should be selected so that they are able to maximally distinguish sounds that are produced by different bumblebee species. Most of the recognition systems based on audio and especially human voice recognition uses Mel-frequency cepstrum coefficients (MFCC) as a feature vector. There are also works where feature vectors are Linear Predictive Coding coefficients (LPC) or a set of low-level signal parameters like in [1].

This paper uses MFCC and LPC to extract the features. For the extraction of features and for other processing of audio records we used jAudio package [9]. Before feature extraction we preprocess and segment the sound recordings. We found that the segmentation is as important as the extraction of features with a strong influence on the prediction accuracy. Then we constructed the model separately with three different classification algorithms: J48, MLP and SVM. Training and evaluation of a model were performed on a stored database of fifteen species of bumblebees. The experiments were carried out using WEKA open source machine learning software. Results show that SVM has better performance than other two systems.

2 PREPROCESSING

Each sound record preprocessing consists of three steps: normalization, pre-emphasis and segmentation. First the normalization is applied to the record by dividing it with maximum value:

$$\tilde{a} = x(i)/max\,x(i), 0 \square i \square n-1 \qquad (1)$$

where $x(i)$ is the original signal, $\tilde{x}(i)$ is the normalized signal and n is the length of the signal.

After that pre-emphasis is performed in order to boost only the high-frequency components, while leaving the low-frequency components in their original state. This approach is based on observations that sound data comes with a high frequency and low magnitude whereas the parts of the recording that we are not interested in (noise, gaps) incorporate low frequency and much higher energy. The pre-emphasis factor α is computed as

$$\alpha = e^{-2\pi F t} \qquad (2)$$

where t is the sampling period of the sound. The new sound is then computed as:

$$H(z) = 1 - \alpha z^{-1} \qquad (3)$$

The last step of the pre-processing is segmentation. In this step we separate sound record into a number of samples which represent only the buzzing. Each sound record is 45 to 60 seconds long. Extracting features from the whole sound record, firstly, increases the computational complexity and, secondly, affects the accuracy of the recognition. We do not need to calculate the features for the silent, noisy and other irrelevant parts of the record.

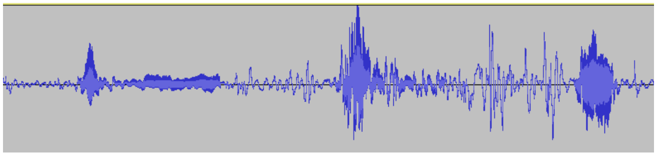


Figure 1: Representation of audio record of humilis queen species in time domain

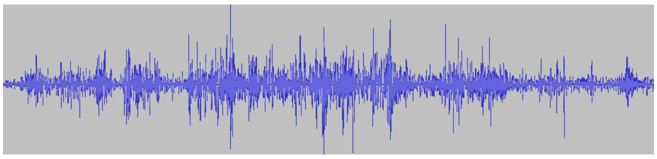


Figure 2: Representation of audio record of sylvarum worker species in time domain

However, spectral changes of signal parts are rather diverse and detection of boundaries of such samples is difficult because adjacent samples of separate buzzings can overlap in time and frequency. Moreover, due to the starting point of buzz is being slow it may occur below the background noise level. In Figure 1 we can see the representation of the sound record of humilis queen. It is difficult to recognize there are three separate relevant parts and everything in between with low frequency components as not relevant.

In Figure 2 it is even more problematic to say where exactly buzzing of the sylvarum worker starts and only in 20% of the recording there is the buzzing signal we are interested in. During the investigation of spectrum of each bumblebee species we also found out that buzzing of the same species can vary based on the state of the bumblebee or during the buzzing of one species some other ones can join and as a result we will have a combination of buzzes. Same species makes one buzz when for example it is working and has some other different buzz when it is angry.

We have to take into account various factors in devising a segmentation method, since unsuccessful separation of a record would result in unsuitable candidate samples and subsequently parametric representation would be different than for real signal data. That is why for the current version of our work we decided to segment the audio recordings manually by an audio editor program so that we could see the result of recognition based purely on real signal data. On one hand, this decision of manual separation obliges us to use in a testing phase of the model only noisiness records where most parts of the record consists of signal data. But on the other hand we analyzed how recognition accuracy changes when we change the strategy for segmentation, since by visually looking at the spectrum of record it is easier to segment it. In this current state of the work we segmented the recording manually into samples of 1-4 seconds of length and the parts which have less than 1 seconds of buzzing duration we combined with adjacent samples.

3 FEATURE EXTRACTION AND MODEL CONSTRUCTION

For each sample segment we calculated MFCC and LPC features. These are features that are mostly used in audio based classification tasks. Samples are processed in a window-by-window manner. The size and the overlapping factor of windows are the two key parameters in feature extraction in general in any audio/signal processing task. We found that the window size of 2014 and the overlapping factor of 30% gives us the feature vectors, which subsequently resulted in the best recognition model. For each window we have several MFCC or LPC values. It is better to represent each window with one feature value by aggregating all the values in a window, so we applied the aggregation by computing the mean value for each window.

In this work we considered three classification algorithms for building the model and these are the J48 Decision Tree, MLP Neural Network and SVM algorithms. Models were built by these algorithms to classify among the 15 different bumblebee cases, most common in Slovenia, i.e. central Europe. Classifying between a worker and a queen is not difficult and on average 90% for all species are easily identified, but we are more interested in knowing the exact type of species like hortoum, hypnorum and pratorum in addition to the status of a bumblebee in colony. So for each species we have two cases, either queen or worker, altogether 15 classes. The fact that the number of records for each class of species in our testing and training dataset are not evenly distributed caused slight inconvenience for us to build a good model. Also, this is one of the reasons why we have different rates of accuracy for all species. 5 of the bumble species we recognized with above 80% of accuracy and 2 of them had a rate of 95%. In Table 1 we provide the rates of recognition for each model built separately on MFCC and LPC feature values with the three algorithms.

	LPC	MFCC
J48	56%	56%
MLP	56%	60%
SVM	57%	64%

Table 1: Evaluation of the rates of recognition accuracy for each built model

In practical terms, when the system proposed three most-probable classes, the accuracy rose to over 90% overall, enabling users to distinguish between the three proposed potential solutions visually. This is the way the system works at the moment.

4 CONCLUSION

In future we want to make the segmentation step to separate record of samples automatically in a system by incorporating all we learned from recordings and patterns of the 11 bumblebee species of both types. Also, we are going to build model using HMM and deep learning, because in many works related to audio classification HMM and deep learning produce best results. Then we intend to compare its result with the ones we obtained from SVM.

References:

[1] Seppo Fagerlund. Bird Species Recognition Using Support Vector Machines. EURASIP Journal on Advances in Signal Processing, Volume 2007, Article ID 38637. doi:10.1155/2007/38637

[2] Zhu Leqing, Zhang Zhen. Insect sound recognition based on SBC and HMM. 2010 International Conference on Intelligent Computation Technology and Automation.

[3] Forrest Briggs, Raviv Raich, and Xiaoli Z. Fern. Audio Classification of Bird Species: a Statistical Manifold Approach.

[4] Diego F. Silva, Vinícius M. A. de Souza, Eamonn Keogh, Daniel P. W. Ellis. Applying Machine Learning and Audio Analysis Techniques to Insect Recognition in Intelligent Traps.

[5] Ruben Gonzalez. Better than MFCC Audio Classification Features.

[6] J. F. Bueno, T. M. Francoy, V. L. Imperatriz-Fonseca, A. M. Saraiva. Modeling an automated system to identify and classify stingless bees using the wing morphometry – A pattern recognition approach.

[7] H. Fujisaki, S. Ohno. Analysis and Modeling of Fundemental Frequency Contour of English Utterances. Proc. EUROSPEECH'95. Vol. 2. pp. 634-637. Philadelphia. 1996.

[8] Erling Wold, Thom Blum, Douglas Keislar and James Wheaton. Content Based Classification, Search, and Retrieval of Audio.

[9] Daniel McEnnis, Cory McKay, Ichiro Fujinaga, Philippe Depalle. jAudio: A feature extraction library.

THE LAS VEGAS METHOD OF PARALLELIZATION

Bogdan Zavalnij
Institute of Mathematics and Informatics
University of Pecs
Ifjusag utja 6, H-7634 Pecs, Hungary
e-mail: bogdan@ttk.pte.hu

ABSTRACT

While the methods of parallelizing Monte Carlo algorithms in engineering modeling very popular, these methods are of little use in discrete optimization problems. We propose that the variance of the Monte Carlo method, the Las Vegas method can be used for these problems. We would like to outline the basic concept and present the algorithm working on a specific problem of finding the maximum clique.

1 INTRODUCTION

The Monte Carlo methods have been powerful tools in scientific and engineering modeling for the last half century [1]. Their usage become even more intense in the era of computers. The easiness of parallelization made these methods useful in supercomputer environments as well. But apart from the original idea the more recent versions of these methods, in which category the Las Vegas method is falling [2], found little usage in algorithms, and even less in parallel programs. The few exceptions are the primality tests and the quicksort algorithm, although some research was made earlier in this field [7][8].

The problem we concentrate on is the maximum clique problem [4], although the concept described in this paper applies to other problems in the field of discrete optimization as well. The maximum clique problem can be formulated in the following way. Given a finite simple graph *G=(V,E)*, where *V* represents the nodes and *E* represents the edges. We call Δ a clique of *G* if the set of vertices of Δ is subset of *V*; Δ is an induced subgraph of *G*; and Δ is an all connected graph, thus all its nodes connected to all the other nodes. We call Δ a maximum clique if no other clique of *G* is bigger than it. The maximum clique problem is to find the size of a maximum clique, and it is a well known NP-hard problem. A simplified variation of this problem is the *k*-clique problem, which is a problem of the NP-complete class. The question is, that if given a graph *G*, and a positive integer *k*, is there a clique of size *k* in the graph. To answer the question we either must present a *k*-clique of the graph, or either prove that there is none in the graph.

2 THE FAMILY OF MONTE CARLO METHODS

The family of the Monte Carlo methods can be distinguished by the nature of their error. We can speak about two sided, one sided and zero sided error methods. In the case of two sided errors we approximate the solution step by step. In the analysis of the method we can measure the distance of the approximation from the real solution, and find that we can get closer and closer in each step. This method is mostly useful in engineering modeling of problems with real number solutions. This two sided method can be programmed in parallel environment with ease, as the steps usually independent.

In the case of one sided error, which takes place mostly in decision problems, in each step we either get a final solution, or get no answer. This is the idea behind many primality tests, where we can find the composite numbers, but get uncertain answer for primes. The algorithms make a few dozen steps, and the uncertainty of being wrong decreases to minimum. These algorithms usually are very fast and need no parallelization.

The last method, which is called the Las Vegas method, is the case of zero sided error. The famous quicksort algorithm falls into this category. With these algorithms we always get the right answer – as the quicksort sorts the sequence in the end –, but the running time of the algorithm can be described by a probability variable. In other words sometimes the algorithm is very fast, and sometimes it can be very slow. (Luckily the later case is very-very rare in the case of the quicksort.)

Formally, we call an algorithm a Las Vegas algorithm if for a given problem instance the algorithm terminates returning a solution, and this solution is guaranteed to be a correct solution; and for any given problem instance, the run-time of the algorithm applied to this problem is a random variable. [13]

From this description it is clear, that the Monte Carlo method, on one hand, can be easily used for engineering problems as we are looking for real number answers with certain correctness. On the other hand, in the case of discrete and combinatorial optimization problems we usually need exact answers, so the Las Vegas method can prove itself of more use.

3 PARALLELIZATION WITH THE AIM OF THE LAS VEGAS METHOD

The variance in the running time of a Las Vegas algorithm led Truchet, Richoux and Codognet to implement an interesting way of parallelization the algorithms for some NP-complete discrete optimization problems [13]. The authors note that the algorithm implementation for those problems heavily depends on the "starting point" of the algorithm, as it starts from a random incorrect solution and constantly changes it to find a real solution. Depending on the starting incorrect solution the convergence of the algorithm may be very fast or slow as well. The idea behind the Las Vegas parallel algorithm was to start several instances of the sequential algorithm from different starting points and let them run independently. The first instance which finds the solution shuts down all the other instances and the parallel algorithm terminates. As the running time of the different instances vary, some will terminate faster, thus ending the procedure in shorter time. The article describes the connection of the variance of the running times and the possible speed-up when using k instances and found that for some problems a linear speedup could be achieved.

This approach can be useful in several ways. For example one can use different solvers for a given problem, and/or use different preconditioning techniques. Starting these solvers concurrently will lead the most suitable one to finish in the shortest time, thus leading us to a fast solution. (Note, that for different problems different versions of the solvers may be the fastest.)

While the previous approach is simple and extremely elegant as well, it lacks something. First, the different instances cannot help each other in finding the solution. Second, each of the instances trying to solve the whole problem and no division into subproblems appears in this proposal.

I propose a different approach, which includes these notions. If we divide an NP-hard problem into parts, then the arising subproblems falls into the same category as described: these are also NP-hard problems, and have great variance in solution time. But we have a problem of constructing the sequence of the divided subproblems. As a solution of one subproblem can be helpful in the solution for the other the sequence of these subproblems have great importance: we would like to solve the easier first to help the more complex ones later. Here we can use some heuristics, but more often we proceed in the order the subproblems are already given, which leads to an inefficient algorithm.

Instead we can use the proposed Las Vegas technique starting the instances of the solver for the arisen subproblems parallely. Thus we may overcome the question of the sequence construction. As we seen, some problems will run much faster and terminate with the desired answer. These answers the can be feed to the other instances and help them to solve their subproblem faster. This way each instance solving a partial problem instead of the whole, and can help other instances to solve their subproblems faster. This method resembles the BlackBoard technique known well in the field of Artificial Intelligence.

This approach can be used to parallelize several different discrete optimization algorithms. Namely, we can use it in any Branch-and-Bound technique instead of the branching rule. As it happens at a branching we have the problem of choosing the sequence of the branches. The speed of the algorithm heavily depends on this sequence, as the result in one branch may help us in an other branch – as a new, better bound for example.

4 AN APPLICATION

In order to demonstrate the described method we choose a more simple algorithm than a general Branch-and-Bound. Instead we used an algorithm from Sandor Szabo [10], which answers the *k*-clique problem by dividing the original problem into thousands of subproblems. These subproblems then can be processed parallely with a sequential program. Obviously this algorithm needs proper number of subproblems in order to achieve proper speed-ups, which this algorithms achieves well. The proposal starts with a quasi coloring with *k-1* colors, and then examines each disturbing edge, whether that edge can be an edge of a *k*-clique. If yes, than we found a positive solution, if no, then the edge can be deleted from the graph. After all the disturbing edges are deleted, we get a proper *k-1* coloring, which forbids the *k*-clique, thus we solved the problem. I have implemented this proposal and measured the running times for several different problems [14].

The measurements compares three version of the algorithm. In the first there is no information given from one subproblem to another to help it in the solution. The program instances run totally independently. In the second I constructed a sequence where the helping information is the consequence of this sequential ordering, thus the help given in advance. This means that we can delete the edges in the sequence of the subproblems in advance, proposing that no *k*-clique can contain them. There is no actual communication between the program instances and they also run independently. These two versions are detailed in the paper of Sandor Szabo [10]. The third version is the Las Vegas method, where the program instances starts parallely, and when one is finished, this information is given to others thus speeding up their solution time as the subproblems can be reduced with the aim of this information. In our case if the algorithm for a given subproblem reports that there is no *k*-clique that contains that edge, then we delete this particular edge from all the subproblems including those that are already running. For this purpose we obviously need a sequential clique search program where an edge can be deleted during the runtime.

5 RESULTS

I used three sets of graph problems. The first set is consists of random graph with given probability of the edges. The second set is taken from the DIMACS challenge website

[5][6]. In the third set two extremely hard problem represented, one is from coding theory [3][12], the other is from combinatorial optimization [11]. For all problem we know the clique size, so I run the algorithm to prove that there is no clique bigger by one than the known clique number. This step is important, because finding the maximum clique depends only on luck, thus shows little about the goodness of an algorithm. While for proving that there is no clique which is bigger by one as the known one needs extended search through the whole search tree, and thus provides a good comparison for different algorithms and implementations.

The tables show the name of a problem instance, the size (N), the density (%), the maximum clique size (clique) and the running time for the sequential algorithm on the same computer (seq). I also noted the number of subproblems that arise in the algorithm (parts).

The tests were run on 4+1, 16, 64 and up to 512 processor cores (one core doing the distribution and not taking part in calculation itself), and I show the running times in seconds for those core numbers. The "noopt" results are from the first version with no help between the problems, the "opt" stands for the more optimal version with help from one instance to an other by the original sequence, and the "lv" represents the Las Vegas method of parallelization. If the running time exceeded the time limit the table continues no data (*). The produced results seems to prove the idea interesting. The running times of the third algorithm in most of the cases were close to the running times of the second algorithm with help to other subproblems, which is an interesting fact by itself. But even more interesting, that for some cases it surpassed the second algorithm. These were the most difficult cases, thus this method could perhaps be useful for the solution of the most difficult problems.

Table 1: Random graph problems

N	200	300	500	500	500	1000	1000	1000
%	90	80	60	70	80	40	50	60
clique	40	29	17	22	32	12	15	20
parts	152	540	2478	2231	1664	10918	10955	9823
seq	**623**	**898**	**67**	**3453**	*****	**136**	**447**	**15k**
5-nopt	376	466	431	*	*	1268	*	*
5-opt	109	231	420	1401	*	1156	*	*
5-lv	126	242	424	1444	*	1168	*	*
16-nopt	123	135	119	584	*	350	*	*
16-opt	33	64	116	387	*	319	*	*
16-lv	66	71	118	407	*	329	*	*
64-nopt	49	45	29	142	*	84	368	1236
64-opt	27	16	28	93	18k	76	345	1064
64-lv	39	23	29	99	21k	79	355	1100
512-nopt	38	23	4	25	14k	11	48	158
512-opt	27	15	4	14	6595	10	45	135
512-lv	39	23	4	19	4189	10	46	142

Table 2: Problems from the DIMACS challenge

	brock 800_3	latin_square _10	keller5	MANN _a45	p_hat 1500-1	p_hat 500-3
N	800	900	776	1035	1500	500
%	65	76	75	99	25	75
clique	25	90	27	345	12	50
parts	4888	380	420	45	14918	657
seq	**7302**	**4902**	**4531**	**3666**	**278**	*****
5-nopt	*	*	*	1340	894	*
5-opt	*	1423	*	719	814	*
5-lv	*	1504	*	1051	824	*
16-nopt	*	531	986	402	247	*
16-opt	*	403	672	205	225	*
16-lv	*	430	686	388	232	*
64-nopt	472	150	318	183	60	*
64-opt	413	105	173	140	54	11k
64-lv	425	128	228	174	56	7165
512-nopt	64	82	138	183	9	41k
512-opt	55	62	137	140	8	11k
512-lv	59	83	138	183	8	5300

Table 3: Problems of monotonic matrices and deletion codes

	monoton-7	monoton-8	monoton-9	deletion-9
N	343	512	729	512
%	79	82	84	93
clique	19	23	28	52
parts	313	590	932	375
seq	**7**	**2347**	*****	*****
5-nopt	76	*	-	-
5-opt	74	1282	-	-
5-lv	74	1292	-	-
16-nopt	23	959	-	-
16-opt	21	408	-	-
16-lv	22	385	-	-
64-nopt	8	475	150k	-
64-opt	6	409	150k	-
64-lv	6	195	44k	-
512-nopt	4	405	150k	*
512-opt	2	409	150k	*
512-lv	4	243	31k	255k

ACKNOWLEDGEMENTS

Author would like to thank the HPC Europe grant for the fruitful visit to Helsinki, to the Finish Computer Science Center which hosts the supercomputer Sisu on which the computations was performed.

Refernces

[1] Hammersley, J.M. and Handscomb, D.C. *Monte Carlo Methods.* London. 1975. (1964.)

[2] Babai, L. Monte-Carlo algorithms in graph isomorphism testing. *Université de Montréal, D.M.S.* No.79–10. http://people.cs.uchicago.edu/~laci/lasvegas79.pdf

[3] Bogdanova, G.T., Brouwer, A.E., Kapralov, S.N. and Ostergaard, P.R.J. Error-Correcting Codes over an Alphabet of Four Elements. *Designs, Codes and Cryptography.* August 2001, Volume 23, Issue 3, pp 333–342.

[4] Bomze, I.M., Budinich, M., Pardalos, P.M. and Pelillo, M. (1999). The Maximum Clique Problem. In D.-Z. Du and P.M. Pardalos (Eds.) *Handbook of Combinatorial Optimization.* (pp. 1–74.) Kluwer Academic Publishers.

[5] DIMACS. ftp://dimacs.rutgers.edu/pub/challenge/graph/ (May 30, 2014)

[6] Hasselberg, J., Pardalos, P.M. and Vairaktarakis, G. Test case generators and computational results for the maximum clique problem. *Journal of Global Optimization.* (1993), 463–482.

[7] Luby, M. and Ertel, W. Optimal Parallelization of Las Vegas Algorithms. In Enjalbert, P at all. (Eds.), *Lecture Notes in Computer Science.* (1994). (pp. 461–474.) Springer Berlin Heidelberg.

[8] Luby, M., Sinclair, A. and Zuckerman, D. Optimal Speedup of Las Vegas Algorithms. In: *Proceedings of the 2nd Israel Symposium on Theory of Computing and Systems.* Jerusalem, Israel, June 1993.

[9] Ostergaard, P.R.J. A fast algorithm for the maximum clique problem. *Discrete Applied Mathematics.* (2002), 197–207.

[10] Szabo, S. Parallel algorithms for finding cliques in a graph. *Journal of Physics: Conference Series Volume 268, Number 1.* 2011 J. Phys.: Conf. Ser. 268 012030 doi:10.1088/1742-6596/268/1/012030

[11] Szabo, S. Monotonic matrices and clique search in graphs. *Annales Univ. Sci. Budapest., Sect. Comp.* (2013), 307–322.

[12] Sloan, N. http://neilsloane.com/doc/graphs.html (May 30, 2014)

[13] Truchet, C., et al. Prediction of Parallel Speed-ups for Las Vegas Algorithms. http://arxiv.org/abs/1212.4287 2012.

[14] Zavalnij, B. Three Versions of Clique Search Parallelization. *Journal of Computer Science and Information Technology.* June 2014, Vol. 2, No. 2, pp. 09–20.

RESOURCE-DEMAND MANAGEMENT IN A SMART CITY

Jernej Zupančič^1, *Damjan Kužnar*^{1}, *Boštjan Kaluža*^{1}, *Matjaž Gams*^{1}

^{1}Department of Intelligent Systems, Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana
e-mail: jernej.zupancic@ijs.si

ABSTRACT

Due to the increasing demand and limited amount of natural resources the costs of resource supply is increasing. Reasons for the increasing demand are also the increasing needs of urban life in the city and the lack of mechanism that would encourage proper resource consumption. In the paper we present a hierarchical architecture for resource-demand management in Smart city. The proposed architecture enables distributed computing and robust resource-demand management. Further, we present a two-stage mechanism that encourages the reduction of resource consumption. In first stage it ensures that all the consumers are satisfied with the resource per-unit price and in the second stage rewards are offered to consumers who are prepared to further lower their consumption.

1 INTRODUCTION

The demand of resources such as electricity, water, natural gas and oil is on the rise. Together with limited amount of natural resources they contribute to the increasing costs of resource supply [1]. Consumers usually pay the same per-unit price for a resource, although large consumers are responsible for the rising cost. With the prevalence of metering technology and increasing sustainability awareness of consumers different levels of consumption are urged to be priced differently.

Using proper mechanism one could lower resource demand peaks that are usually expensive (additional less efficient means of resource production have to be enabled) by shifting the consumption of the resource to a part of the day when the resource is in low demand. This way resource can be used much more efficiently and investment into resource production units that are required to meet peak demands can be postponed.

A resource-demand management mechanism that sets the same price for every consumer in advance does not fully exploit demand-management capabilities. Dynamic-pricing mechanisms (the prices change every so often) were already proposed (for instance in [6]). Most of such mechanisms require the consumers either to report their utility function (which raises privacy concerns) or to leave the decision about consumption to the consumers themselves (which is not as efficient as it could be). A dynamic-pricing mechanism that negotiates prices with every individual was proposed in [3]. However, their approach has to assume that the consumers will report their consumption truthfully, which is not always the case. Truth-incentive demand management was proposed in [4]. The mechanism described in [4] has several desirable properties besides being truth-incentive: it is proved to converge to a Nash equilibrium, it is budget balanced (the total cost of resource provision equals the total cost the consumers pay), and it reaches the global optimum (minimal peak-to-average ratio). However, it still raises some issues: the resource consumption of every consumer is known to everyone, consumers are only encouraged to shift their load to different parts of the day and not to reduce consumption, smaller consumption does not result in smaller per-unit price, and real-time pricing requires price prediction capabilities.

In this paper we give a short presentation of a mechanism [7] that addresses some of those shortcomings. It changes the prices dynamically and adapt them to each consumer individually, it is truth-incentive, it encourages lower resource consumption and it preserves privacy of the consumer. Further, we present the architecture that enables the application of the proposed mechanism.

The rest of the paper is structured as follows. In section 2 we present the envisaged architecture for resource-demand management in Smart city. In section 3 we give general description of demand-management mechanisms and in section 4 the negotiation protocol that encourages consumption reduction of convexly priced resources is presented. Section 5 summarizes and concludes the paper.

2 ARCHITECTURE FOR RESOURCE-DEMAND MANAGEMENT IN SMART CITY

Every city is hierarchically structured into districts then further into streets and then even further into individual buildings with devices and appliances. Therefore, it is only natural that resource-demand management architecture acknowledges this hierarchical structure. Although hierarchical structure has some disadvantages against star formation (Figure 1) that is typically used for resource-demand management (hierarchical structure requires more communication nodes and more messages to be transmitted, which could result in slower response) it also possesses some desirable properties: it is distributed, it is more resistant against failures and it can better represent the reality and real world decision making.

Decision nodes in the architecture have to communicate among themselves, they need some computational ability and they have to take into account real-world decision maker preferences when taking decision by themselves. They are agents forming a multi-agent system. Hierarchical organizational

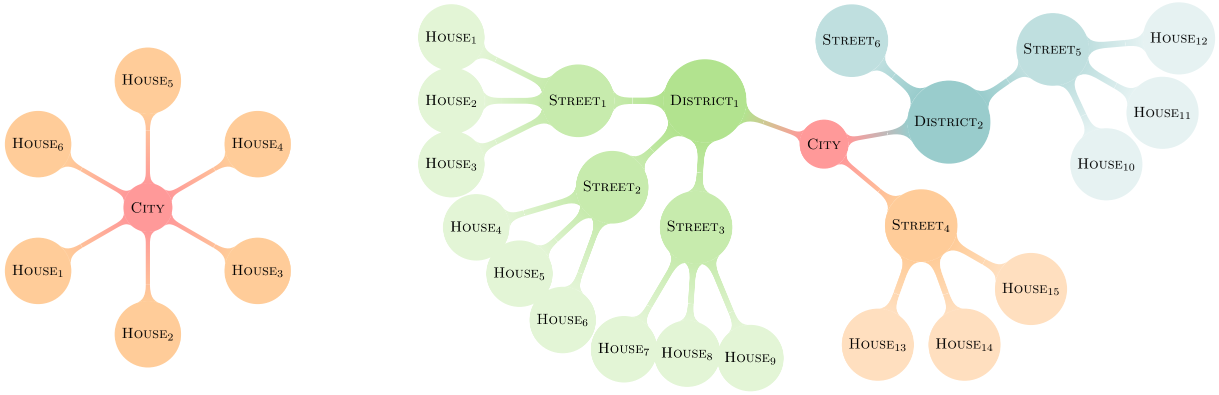


Figure 1: Hierarchical structure on the right and star formation on the left

structure of multi-agent system can be described as follows. At the root of the structure is the top decision entity in the city and it is responsible for setting the cost of resource provision and global price computation of the resource. We call the root node a *Resource negotiator agent*. At lower levels there are *Aggregation agents* that propagate the price information from higher nodes to lower nodes and resource consumption information from lower to higher nodes. They could be independent since every district could have its own policies on resource consumption. At the lowest level there are *House coordinator agents* that can operate a group of appliances and devices. Every parent node can either negotiate with its child node/agent (when child agent does not reveal its information) or optimize it (when child agent reveals its information and allows the parent agent to control it).

3 RESOURCE PRICING MECHANISM

In previous section we presented an underlying architecture that defines the network of nodes and connections between them. In this section we will give some general information about mechanisms or protocols. Mechanisms define how the communication is carried out over the proposed architecture, what is the content of communication and what are the rules for the interaction between the nodes. We can classify mechanisms according to their properties.

A mechanism is **strategy proof** or **truthful** when the best the agents can do when the mechanism is in action, is to tell the truth. There is no incentive for any of the agent to lie about its information.

A mechanism **converges** to some final distribution of prices and consumptions in finite number of steps. Convergence property guarantees that the mechanism ends, however, the speed of convergence is important as well and should be addressed.

A mechanism is **budget-balanced** when the cost of resource provision to the consumers should be the same as the total cost the consumers pay for the consumed resource. No agent can benefit from the mechanism, be it a resource producer or resource consumer.

All mentioned mechanism properties are preferred in a mechanism that is to be implemented in reality.

Mechanisms can be further divided according to the length of the time period for which they determine consumption. Resource consumption over the day can be divided into finitely many time intervals. A **dynamic mechanism** can determine the resource consumption for the next short time interval (on the scale of an hour), in that case the price and consumption are set dynamically every time a new time interval approaches. This type of mechanism is used when it is difficult to determine the cost of resource production and distribution far in advance. With the dynamic mechanism the agents cannot schedule the operation of the devices or appliances for the whole day. The resource demand dynamically matches the price of the resource.

A **scheduling mechanism** can determine the price and the resource consumption for every time interval for the whole day. Appliances operation is scheduled in a way that optimizes the cost or energy efficiency and meets the requirements of the user. When using a scheduling mechanism a resource negotiator needs a good knowledge of the resource provision cost for every time interval for the whole day. Unexpected events can greatly disturb the schedule set by the agents.

4 A DYNAMIC NEGOTIATION MECHANISM FOR CONVEXLY PRICED RESOURCES

In this section we give a quick overview of a negotiation mechanism that encourages the reduction of resource consumption of convexly priced resource. A more detailed description can be found in [7]. The mechanism consists of two stages: a negotiation stage and a renunciation stage. In the first stage the goal is to reduce resource consumption to a level where every consumer is satisfied with the price it has to pay. In second stage rewards are offered to consumers that are prepared to further lower their consumption.

In **negotiation stage** many rounds take place with consumers reporting their desired consumption to the negotiator and negotiator computing prices for every consumer. The prices are computed individually for every consumer using serial cost sharing mechanism [5]. The serial cost sharing mechanism determines fair price for every consumer – lower consumption results in lower price. Further, since the resource is convexly priced it also results in lower per-unit pricing.

Individual price is computed by a function $price(i, f, c)$ that takes as inputs the following parameters: consumer i, a sorted consumption vector c and resource cost function f. Function $price$ is defined recursively in Eq. (1).

$$\begin{aligned} &price(0, f, c) = 0 \\ &price(i, f, c) = price(i-1, f, c) + \qquad (1) \\ &+ \frac{f\left(\sum_{j=1}^{i-1} c[j] + (N+1-i) \cdot c[i]\right) - f\left(\sum_{j=1}^{i-1} c[j]\right)}{N+1-i}, \end{aligned}$$

where N is the number of consumers participating in negotiation.

If any of the consumers does not agree with the price it receives and wants to further reduce its consumption anticipating a lower per-unit price, another round of negotiation takes place. In the following round the desired consumption of individual consumer can be the same or lower. Negotiation stage ends when the demand is the same for every consumer in two consecutive rounds.

In **renunciation stage** the consumers that further reduce their consumption are rewarded under the condition that the reward they are offered is sufficient to compensate their consumption reduction. There is only one round in renunciation stage. Consumers are addressed one by one starting with the consumer with the lowest demand. Algorithm for the renunciation stage outputs new prices while taking into account further reductions offered by consumers and rewards demanded. Negotiator computes the reduction in the resource total supply cost (the cost is lowered due to lower demand), which could be offered as a reward to a consumer. To ensure that the consumers who had lower resource demand after the negotiation stage receive lower final per-unit price we may have to adjust the reward. If the consumer agrees with the reward it lowers its consumption and receives a discount.

4.1 Mechanism properties

The presented mechanism has several desirable properties. In this section we will list them and present the sketches of the proofs.

Negotiations converge in a finite number of steps. The renunciation stage ends in one round, therefore, we only have to show that the negotiation stage of the protocol ends in a finite number of rounds. Since there is a finite number of consumers, and every consumer has a finite number of appliances and devices, and every appliance or device has a finite number of operating states, there ia a finite number of consumption level combinations. Since the consumers cannot increase their desired consumption in two consecutive rounds, a repetition of demand will surely occur. Therefore, negotiation will end.

The protocol is strategy-proof when a consumer is risk-avert. A consumer could insist on a large desired consumption in the negotiation stage and then try to obtain large reward when reducing the resource consumption in the second stage. However, rewards are first given to the small consumers and those rewards are also the largest, since the pricing of the resource is convex. Further, the rewards are adjusted so that the consumer with a higher demand after the negotiation stage cannot obtain a lower per-unit price in the renunciation stage. Therefore, by lying an agent risks getting a price it is not prepared to pay. Truth-telling is therefore the best strategy for a risk-averse agent.

Consumers are motivated to reduce resource consumption by the protocol itself. Due to serial cost sharing mechanism, which is used for pricing in the negotiation stage, lower demand results in a lower resource price. Further, since the resource is convexly priced, per-unit price is reduced as well.

The protocol is budget-balanced. Serial cost sharing mechanism defines prices in a way that the cost of resource supply equals to the total cost the consumers have to pay. In the renunciation stage, when a consumer reduces its resource consumption, a budget imbalance (a surplus) occurs in the favor of the negotiator (seller of the resource), since larger consumers pay a higher price than needed to obtain the desired amount of the resource. The surplus is then offered to the consumer who reduced its consumption and sometimes even to the consumers with lower consumption, due to the per-unit price equalization. Therefore, no surplus is generated at the end of the renunciation stage.

4.2 Experiments

We tested the negotiation mechanism on the multi-agent system implemented in JADE (Java Agent DEvelopment) framework [2]. We implemented a resource negotiator agent and house representative agents in star formation. Each house representative agent possesses the information about the electrical energy consumption of the devices and the information about consumer preferences (maximal per-unit price for operating each device and required reward for not operating the device). This information is private to the house representative agent and sent neither to resource negotiator nor to other house representative agents. Linear piecewise function was chosen for convex resource cost function.

In typical simulation run reduction of energy consumption level can be observed (Figure 2). In the first round of the negotiation (up to the dashed line), every house representative achieves the price it is willing to pay. A further reduction in resource consumption occurs in the renunciation stage (between the dashed and the solid line), where per-unit price reduces as well for low consuming houses H1 and H2. Large consuming house H3 does not receive the reward large enough to further reduce its consumption.

In scalability test we gradually increased the number of agents involved in the experiment up to 100 000. Linear scalability of the mechanism is observed (Figure 3).

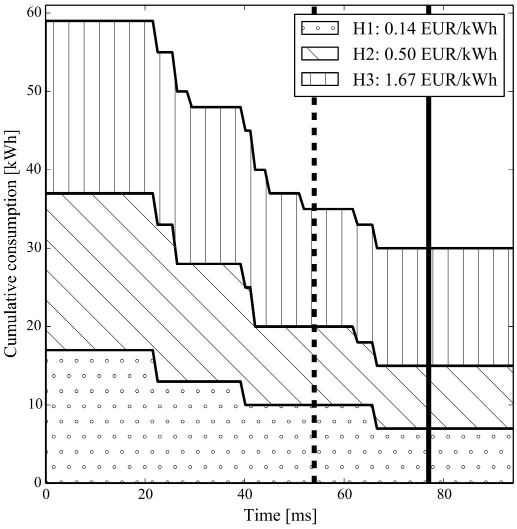


Figure 2: Cumulative consumption observed during negotiation, with final prices added

5 CONCLUSION

In the paper we presented a hierarchical architecture for resource-demand management in smart cities, which is a natural representation of the city and enables distributed computation and robust control. We defined classes of the mechanisms that can be applied to the proposed architecture. In second part of the paper we presented a dynamic mechanism that encourages the reduction of resource consumption when resource cost function in convex. It is a two-stage mechanism that ensures consumer satisfaction in the first stage when consumer is truthful and encourages further reduction of consumption in second stage by offering rewards. The mechanism has several desirable properties: it is budget-balanced, converges in finite number of steps, it is strategy proof and scales linearly with the number of agents that participate in the mechanism.

Further work will include the analysis and modelling of the consumer behaviour. The mechanism will then be applied to the real world models. The goal of this research is to provide a modular mechanism that will incorporate dynamic demand-response together with scheduling. Further, it will be able to deal with: different types of architectures, agents that have hidden information, and agents that reveal all their information. That kind of mechanism will combine optimization and negotiation in an efficient and universal way.

Acknowledgements

We thank Gregor Grasselli, Matej Krebelj and Jure Šorn for the help with the implementation of the experiments in JADE environment. The research was sponsored by ARTEMIS Joint Undertaking, Grant agreement nr. 333020, and Slovenian Ministry of Economic Development and Technology.

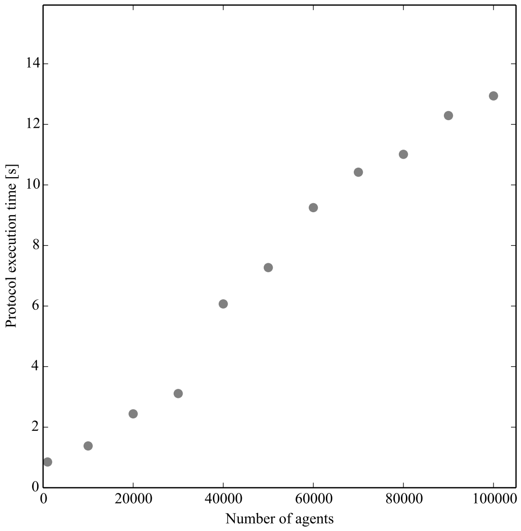


Figure 3: Scatter plot for the scalability experiment

References

[1] E. B. Barbier. Economics, natural-resource scarcity and development: conventional and alternative views. *Routledge*, 2013.

[2] F. L. Bellifemine, G. Caire, D. Greenwood. Developing multi-agent systems with JADE. *John Wiley & Sons*, 2007.

[3] F. Brazier, F. Cornelissen, R. Gustavsson, C. M. Jonker, O. Lindeberg, B. Polak, J. Treur. Agents negotiating for load balancing of electricity use. *Distributed Computing Systems, 1998. Proceedings. 18th International Conference on*, pp. 622–629, 1998.

[4] A-H. Mohsenian-Rad, V. W. S. Wong, J. Jatskevich, R. Schober, A. Leon-Garcia Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid. *Smart Grid, IEEE Transactions on* **1**(3), pp. 320–331, 2010.

[5] H. Moulin, S. Shenker. Serial cost sharing. *Econometrica: Journal of the Econometric Society*, pp. 1009–1037, 1992.

[6] P. Samadi, H. Mohsenian-Rad, R. Schober, V. W. S. Wong Advanced demand side management for the future smart grid using mechanism design. *Smart Grid, IEEE Transactions on* **3**(3), pp. 1170–1180, 2012.

[7] J. Zupančič, D. Kužnar, B. Kaluža, M. Gams. Two-stage negotiation protocol for lowering the consumption of convexly priced resources. *IAT4SIS '14, Workshop proceedings on, to be published*, 2014.

Indeks avtorjev / Author index

Konferenca / Conference
Uredili / Edited by

Inteligentni sistemi /
Intelligent Systems
Rok Piltaver, Matjaž Gams