Longbing Cao · Yifeng Zeng
Andreas L. Symeonidis · Vladimir Gorodetsky
Jörg P. Müller · Philip S. Yu (Eds.)

Agents and Data Mining Interaction

LNAI 8316

9th International Workshop, ADMI 2013
Saint Paul, MN, USA, May 6–7, 2013
Revised Selected Papers

Springer

Lecture Notes in Artificial Intelligence 8316

Subseries of Lecture Notes in Computer Science

For further volumes:
http://www.springer.com/series/1244

Longbing Cao · Yifeng Zeng
Andreas L. Symeonidis · Vladimir Gorodetsky
Jörg P. Müller · Philip S. Yu (Eds.)

Agents and Data Mining Interaction

9th International Workshop, ADMI 2013
Saint Paul, MN, USA, May 6–7, 2013
Revised Selected Papers

Springer

Editors
Longbing Cao
University of Technology Sydney
Sydney, NSW
Australia

Vladimir Gorodetsky
St. Petersburg Institute for Informatics
St. Petersburg
Russia

Yifeng Zeng
Teesside University
Middlesbrough
UK

Jörg P. Müller
Technische Universität Clausthal
Clausthal-Zellerfeld
Germany

Andreas L. Symeonidis
Aristotle University of Thessaloniki
Thessaloniki
Greece

Philip S. Yu
University of Illinois Chicago
Chicago, IL
USA

ISSN 0302-9743 ISSN 1611-3349 (electronic)
ISBN 978-3-642-55191-8 ISBN 978-3-642-55192-5 (eBook)
DOI 10.1007/978-3-642-55192-5
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014938480

LNCS Sublibrary: SL7 – Artificial Intelligence

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

We are pleased to welcome you to the proceedings of the 2013 International Workshop on Agents and Data Mining Interaction (ADMI-13), held jointly with AAMAS 2013.

In recent years, agents and data mining interaction (ADMI, or agent mining) has emerged as a very promising research field. Following the success of ADMIs in the past nine years, the ADMI-13 event provided a premier forum for sharing research and engineering results, as well as potential challenges and prospects encountered in the coupling between agents and data mining.

The ADMI-13 workshop encouraged and promoted theoretical and applied research and development, which aims to:

- Exploit agent-enriched data mining and demonstrate how intelligent agent technology can contribute to critical data mining problems in theory and practice
- Improve data mining-driven agents and show how data mining can strengthen agent intelligence in research and practical applications
- Explore the integration of agents and data mining toward a super-intelligent system
- Discuss existing results, new problems, challenges and the impact of the integration of agent and data mining technologies as applied to highly distributed heterogeneous, including mobile, systems operating in ubiquitous and P2P environments
- Identify challenges and directions for future research and development on the synergy between agents and data mining, especially in agents for big data

This volume comprises ten papers contributed by authors across eight countries. ADMI-13 submissions cover regions from North America, Europe to Asia, indicating the booming of agent mining research globally. The workshop also included two invited talks by two distinguished researchers.

Following the tradition of ADMI, the papers accepted by ADMI-13 were further revised and then published as LNAI conference proceedings by Springer. We appreciate Springer, in particular Alfred Hofmann, for the continuing publication support.

ADMI-13 was sponsored by the Special Interest Group: Agent-Mining Interaction and Integration (AMII-SIG: www.agentmining.org). We appreciate the guidance of the Steering Committee. More information about ADMI-13 is available from the workshop website: http://admi13.agentmining.org/.

Finally, we appreciate the contributions made by all authors, Program Committee members, invited speakers, and AAMAS 2013 workshop and local organizers.

May 2013

Jörg P. Müller
Philip S. Yu
Longbing Cao
Yifeng Zeng
Andreas L. Symeonidis
Vladimir Gorodetsky

Organization

Program Co-chairs

Longbing Cao	University of Technology Sydney, Australia
Yifeng Zeng	Teesside University, UK
Andreas L. Symeonidis	Aristotle University of Thessaloniki, Greece
Vladimir Gorodetsky	Russian Academy of Sciences, Russia

General Co-chairs

Jörg P. Müller	Clausthal University of Technology, Germany
Philip S. Yu	University of Illinois at Chicago, USA

Organizing Co-chairs

Yiling Zeng	University of Technology Sydney, Australia
Yingke Chen	Aalborg University, Denmark

Program Committee

Ahmed Hambaba	San Jose State University, USA
Andreas L. Symeonidis	Aristotle University of Thessaloniki, Greece
Balaraman Ravindran	Indian Institute of Technology Madras, India
Bo An	Chinese Academy of Sciences, China
Bo Liu	University of Massachusetts Amherst, USA
Daniel Kudenko	University of York, UK
Daniel Zeng	Arizona University, USA
Deborah Richards	Macquarie University, Australia
Dionisis Kehagias	Informatics and Telematics Institute, Greece
Eduardo Alonso	University of York, UK
Elizabeth Sklar	City University of New York, USA
Fan Yang	Xiamen University, China
Hua Mao	Aalborg University, Denmark
Janusz Sobecki	Wroclaw University of Technology, Poland
Joerg Mueller	Technical University Clausthal, Germany
Katia Sycara	Carnegie Mellon University, USA
Ladjel Bellatreche	National Engineering School for Mechanics and Aerotechnics, France
Longbing Cao	University of Technology Sydney, Australia

Seunghyun Im	University of Pittsburgh at Johnstown, USA
Simon Parsons	City University of New York, USA
Sviatoslav Braynov	University of Illinois at Springfield, USA
Tapio Elomaa	Tampere University of Technology, Finland
Valérie Camps	Paul Sabatier University, France
Vladimir Gorodetsky	Russian Academy of Sciences, Russia
William Cheung	Hong Kong Baptist University, SAR China
Xudong Luo	Sun Yat-sen University, China
Yan Wang	Macquarie University, Australia
Yifeng Zeng	Teesside University, UK
Yingke Chen	Aalborg University, Denmark
Yuqing Tang	Carnegie Mellon University, USA
Yves Demazeau	CNRS, France
Zbigniew Ras	University of North Carolina, USA
Zhi Jin	Peking University, China
Zili Zhang	Deakin University, Australia
Zinovi Rabinovich	University of Southampton, UK

Steering Committee

Longbing Cao	University of Technology Sydney, Australia (Coordinator)
Edmund H. Durfee	University of Michigan, USA
Vladimir Gorodetsky	St. Petersburg Institute for Informatics and Automation, Russia
Hillol Kargupta	University of Maryland Baltimore County, USA
Matthias Klusch	DFKI, Germany
Jiming Liu	Hong Kong Baptist University, China
Michael Luck	King's College London, UK
Pericles A. Mitkas	Aristotle University of Thessaloniki, Greece
Joerg Mueller	Technische University Clausthal, Germany
Ngoc Thanh Nguyen	Wroclaw University of Technology, Poland
Gerhard Weiss	Software Competence Center Hagenberg, Austria
Xindong Wu	University of Vermont, USA
Philip S. Yu	University of Illinois at Chicago, USA
Chengqi Zhang	University of Technology Sydney, Australia
Andreas L. Symeonidis	Aristotle University of Thessaloniki, Greece

Contents

Agent Mining

Using Dynamic Bayesian Networks to Model User-Experience

Paul van Schaik[1(✉)], Yifeng Zeng[2], and Iain Spears[1]

[1] School of Social Sciences and Law, Teesside University, Middlesbrough, UK
[2] School of Computing, Teesside University, Middlesbrough, UK
{p.van-schaik,y.zeng,i.spears}@tees.ac.uk

Abstract. This paper presents a new approach to modelling the time course of user-experience (UX). Flexibility in modelling is essential: to select or develop UX models based on the outcome variables that are of interest in terms of explanation or prediction. At the same time, there is potential for (partial) re-using UX models across products and generalisation of models. As a case study, an experience model is developed for a particular consumer product, based on a time-sequential framework of subjective well-being [13] and a theoretical framework of flow for human-computer interaction [23]. The model is represented as a dynamic Bayesian network and the feasibility and limitations of using DBN are assessed. Future work will empirically evaluate the model with users of consumer products.

1 Introduction

Parallel to the spread of personal computing, user-experience (UX) has become a major area in HCI research. Sutcliffe [20] provides a useful definition of UX: users' judgment of product quality arising from their experience of interaction, and the product qualities which engender effective use and pleasure. UX stresses that interactive products do not only deliver functional benefits, they promote experiences too, and users intention to (re)live positive experiences is an important driver of technology use [7]. Because most modern interactive products, such as laptop computers, hand-held devices (e.g. smart phones) and tablets, can be used both for work and leisure, utilitarian aspects (e.g., ease of use and learnability) are widely regarded as important, but insufficient by themselves to give a complete account for the acceptance, use and success of these technologies [3]. Indeed, the main idea behind the concept of UX is that the success of interactive products is fundamentally connected to their ability to promote high-quality experiences, but usability remains important. It is helpful to distinguish between instrumental and non-instrumental factors in relation to UX [22]. Usability of a product, as an instrumental factor, may strongly contribute to negative experiences, if it does not reach a satisfactory level expected by users. However, in order to achieve positive experiences, high levels of non-instrumental factors (e.g. positive and negative affect) are needed.

L. Cao et al. (Eds.): ADMI 2013, LNAI 8316, pp. 3–13, 2014.
DOI: 10.1007/978-3-642-55192-5_1,

Models that represent HCI knowledge are useful to summarize data, formalize relationships between variables and to make predictions, even if or precisely because they possess a degree of incompleteness and falseness. Indeed, HCI models can have theoretical and practical value as long as they fit data well, and make theoretical and practical sense, without actually being entirely truthful in their description of a particular phenomenon or process. Flexibility in modeling is therefore essential: to select or develop UX models based on the outcome variables that are of interest in terms of explanation or prediction, instead of using a single *one-size-fits-all* approach. Usually outcome variables are seen as indicators of success of a particular product, for example satisfaction or overall evaluation of experience. Outcome variables can be derived from, for instance, defined user-requirements (e.g. health improvement) or marketing objectives (e.g. satisfied customers). After UX has been measured it is possible to establish (*a*) to which extent requirements or objectives of the product have been met and (*b*) which other variables mostly contribute to explaining variance in the outcomes, as a basis for potential product improvement. Products that share the same outcome variables may share the same or similar models, thereby facilitating potential (partial) re-use UX models for new products and generalization of models.

With a change in emphasis from usability to experience, it is increasingly important that products promote a high-quality experience. This is particularly important for new technology that users may be unfamiliar with, such as augmented reality (AR). AR systems could promote high-quality UX, but there is a lack of UX research to underpin the design of such systems. Research to inform the design of such products is expected to benefit both product users and product manufacturers.

Existing models of UX have been formulated and tested with techniques based on the general linear model. In particular, multiple regression analysis, variance-based structural equation modeling (partial least-squares path modeling) and covariance-based structural equation modeling have been used. In this paper we explore the use of dynamic Bayesian networks, with the following contributions: (*a*) a flexible, but theory-driven, approach to UX modeling, (*b*) the specification of a particular well-grounded theory-based UX model and (*c*) the representation of the model as a dynamic Bayesian network and analysis of the modeling work. Section 2 presents related work. Section 3 presents the modeling approach, followed by conclusions and future work in Sect. 4.

2 Related Work

Existing research on UX modeling distinguishes instrumental and non-instrumental aspects of experience. However, in this work UX outcomes are usually non-instrumental. In Hassenzahl's user-experience [7,8] model perceptions of product characteristics (pragmatic quality and hedonic quality) are antecedents of global product evaluations (goodness and beauty). In Porat and Tractinsky's environmental-psychology UX model [18], environmental stimuli (classical aesthetics, expressive aesthetics and usability) are antecedents of emotional states

(pleasure, arousal and dominance); in turn, these are antecedents of attitudes towards service. In Thüring and Mahlke's [22] CUE model, system properties, user-characteristics, and task/context are antecedents of interaction characteristics; in turn, these are antecedents of perceptions of instrumental qualities and perceptions of non-instrumental qualities, both of which lead to emotional reactions; all three are antecedents of appraisal of the system. In Hartmann *et al.*'s model of user-interface quality assessment [8], three stages are involved in users' judgment of quality assessment. First, users assess an interactive system based on their goals and the task domain. Second, users select decision-making criteria based on their goals and task. Third, users evaluate the system using these criteria.

Tests of these four UX models were in empirical studies used analysis of variance [7,21,22], partial correlation [7], covariance-based structural modeling [18]. Recent work has proposed the use of dynamic Bayesian networks (DBNs) for modeling quality of experience [11]. In their approach, context attributes are antecedents of the context state; in turn, context-state variables are antecedents of the situation state. The approach is illustrated with simulation results. Limitations of this work include the following. The work has no apparent credible theoretical justification; it does not build on existing theory of UX. Furthermore, it does not account for experience of a particular episode of interaction as it happens and global judgment of interaction with a product. Instead it only accounts for the memory of interaction episodes. Moreover, it does not account for causal relations between these three aspects of experience and, in a gross simplification, reduces the measurement of technology acceptance to a Boolean.

A major shortcoming of existing UX research on AR (and interactive products more generally) is that often actual product use and long-term use are not studied [24]. Furthermore, the role of task performance is not addressed; moreover, most research is not experimental, so cause (design) and effect (UX) cannot be established. Therefore, our program of research aims to conduct experimental research that models UX over time to inform the design of AR systems to sustain high-quality UX. We use a time-sequential framework of subjective well-being [13], our methods for modeling UX [23] and our hybrid real-time motion measurement system [19]; this work is expected to lead to new applications and improvements in product design.

3 Modeling Approach

Models of UX specifying determinants of positive experiences have been tested with a range of interactive devices and technologies. However, several challenges in UX modeling remain, in terms of UX theory, research design, technical solution and application of modeling to product design. Based on Kim-Prieto *et al.*'s time-sequential framework of subjective well-being [13], a time-sequential framework of UX can be framed as a sequence of stages over time: from the experience of a particular episode of interaction as it happens (**Level 1**) to the memory of interaction episodes (**Level 2**) to global judgment of interaction with a product (**Level 3**).

The approach taken here to model UX with a product over time uses DBNs [11]. This is illustrated with a consumer product (shaver), but the approach applies without loss of generality to any product. Based on existing work with industry by the research team, a sensor-embedded shaver will be developed. The shaver will communicate with a users existing smart phone to record the users behavior and measure the users experience in terms of memory of experience episodes (shaves) and global judgment of shaving experience.

The use of DBNs for UX modeling over time has several advantages over other techniques such as multiple regression analysis, structural equation modeling (SEM, in particular PLS path modeling and covariance based SEM), multilevel modeling and time series analysis are. DBNs are a dynamic version of probabilistic graphical models - Bayesian networks-that represent cause-effect relations embedded in a domain. They are able to structure the relations over time and provide an intuitive tool for conducting various inference tasks in the domain. To make a functional DBNs, it is always quite tedious to construct the DBNs manually, which requires a large amount of knowledge input from domain experts. Considering the availability of data in our domain, we are using automatic methods to learn DBNs from the accumulated data over subject study. However, it remains important that modeling results are grounded in theoretical understanding in order to build cumulative knowledge; therefore, as a starting point, we derive a well-argued theoretical model by integrating existing theoretical frameworks.

A recent theoretical framework of flow for HCI will be used [23] because the crucial role of task performance in modeling UX and the theory of flow experience (the degree to which a person feels involved in a particular activity) uniquely addresses this performance. In this framework, characteristics of person (user), artifact (product) and task are antecedents of flow experience. Flow experience consists of two main components: preconditions of flow and the dimensions of flow proper. Consequents of flow include objective, subjective and behavioral out-comes. The concept of flow is linked to that of effortless(ness of) performance [2]: the more flow people experience, the more effortless/less effortful their task performance is.

3.1 Data Capture and Variables

At **Level 1**, experience as it happens is inferred from captured sensor data and secondary-task data collected during each interaction episode. The three-dimensional position of the shaver is recorded continuously as well as the force a user applies to the shaver (and, optional, muscle activity). From these, measures of effortlessness are computed, including accuracy of motor performance, frequency and size of (motor) corrections and speed of action (variability) [5]. Performance is more effortless with more accurate motor performance, more frequent and smaller (motor) corrections and greater speed of action [5].

From a secondary task (for example, a reaction-time task where people respond to specific [sound] signal), response time is recorded. Attentional demand

is measured as speed of secondary-task performance (timing of response relative to signal). Performance is more effortless with reduced attentional demand (faster secondary-task performance). In sum, online UX variables for DBN-modeling include: (U1)Accuracy of motor performance; (U2)Frequency and size of (motor) corrections; (U3)Speed of action; (U4)Speed of secondary-task performance.

Rather than using these variables as nodes in a model, we use a hidden (latent) variable to represent *effortlessness* inferred from these (indicator) variables, with reflective measurement. This is because the latent variable is the *cause* of the variables. At **Level 2**, memory of interaction episode is inferred from captured psychometric-questionnaire data collected immediately after each interaction episode. Flow experience is measured using Guo and Pooles 30-item inventory (or a subset or a similar instrument) [6]. From the 30 items, nine dimensions as hidden (latent) variables are inferred. The first three dimensions are preconditions of flow and the remaining six are dimensions of flow proper. For simplification, from the six dimensions of flow one high-order flow dimension may be inferred, but autotelic experience (Dimension 9) may also be used in the modeling as a variable on its own as it captures most clearly the intrinsically motivational value of flow experience. Visual attractiveness is measured using a single item from Tractinsky *et al.* [17], using a 10-point semantic differential. Affect is measured using PANAS with 10 items for positive affect and 10 for negative affect [1]. From the 20 items, two dimensions (positive and negative affect) are inferred. In sum, online interaction-memory variables for DBN-modeling include: (U5)Balance of challenge and skill; (U6)Goal clarity; (U7)Feedback; (U8)Autotelic experience; (U9)Visual attractiveness; (U10)Positive affect; (U11)Negative affect.

Quality of task result (quality of shave) is assessed from a photograph taken of a particular save and satisfaction with task result from psychometric-questionnaire data collected immediately after each interaction episode. Quality of task result is rated by an independent judge or through image interpretation software. Items(5) to measure satisfaction from result are developed, based on existing instruments. From the items the satisfaction with task result as hidden (latent) variables is inferred. In sum, task result variables for DBN-modeling include: (U12)Quality of task result; (U13)Satisfaction with task result.

At **Level 3**, global judgment of interaction is inferred from captured psychometric questionnaire data collected after a number of interaction episodes. Items to measure utility(4), appearance(4), positive memories(3), pleasure of interaction(2), product attachment(5) and product satisfaction(4) are from Mugge et al. [16]. Items to measure intention(4) of future purchase are adapted from Kowatsch and Maass [14]. From the items the three global-judgment constructs as hidden (latent) variables are inferred. In sum, global-judgment variables for DBN-modeling include: (U14)Utility; (U15)Appearance; (U16)Positive memories;

(U17)Pleasure; (U18)Product attachment; (U19)Product satisfaction; (U20) Intention of future purchase.

The role of person characteristics in relation to flow experience is that they moderate the effect of preconditions of flow experience on flow proper [4]. These characteristics are inferred from captured psychometric-questionnaire data collected once at the start of a trial. The constructs of achievement motivation and in particular action orientation (volatility subscale of the Action Control Scale, 12 items) from Diefendorff *et al.* [10] and perceived importance (3 items) from [4] are measured. From the items each of these constructs are inferred as hidden (latent) variables. In sum, person variables for DBN-modeling include: (U21)Action orientation (volatility); (U22)Perceived importance.

3.2 Static BN

To structure a potential BN, we proceed to specify relations among variables by exploiting their description in the existing literatures.

Level 1. Experience during interaction. The hidden variable *effortlessness* is modeled reflectively as a cause. This is because all measured variables at Level 1 are indicators of and 'caused' by effortless attention. We may abuse the functional relations as follows.

- Accuracy of motor performance = F(Effortlessness)
- Frequency and size of (motor) corrections = F(Effortlessness)
- Speed of action = F(Effortlessness)
- Speed of secondary-task performance = F(Effortlessness)

Level 2. Memory of interaction episode. Peoples memory of flow immediately after an interaction episode reflects the degree of effortlessness of the activity [2]. Therefore,

- Balance of challenge and skill = F(Effortlessness)
- Goal clarity = F(Effortlessness)
- Feedback = F(Effortlessness)

According to the staged model of flow experience [6] preconditions of flow are causes of flow experience proper; according to Engeser and Rheinberg [4] and Keller and Bless [12], achievement motivation is a moderator of the effect of the preconditions of flow on flow proper; according to Engeser and Rheinberg [4], importance is a moderator of this effect. Therefore,

- Autotelic Experience = F(Balance of challenge and skill, Goal clarity, Feedback, Achievement motive, Importance)

Because of cognitive (attention-enhancing) and motivational facilitation [4,23] task performance and the result of task performance are enhanced. Therefore,

– Quality of task result = F(Effortlessness, Balance of challenge and skill, Goal clarity, Feedback)

Level 3. Global judgment of interaction. Consistent with Kim-Prieto *et al.* [13], (immediate) memories of task result provides extrinsically motivational value that contributes to the global judgment of perceived utility. Therefore,

– Utility = F(Satisfaction with task result)

Brief (immediate) judgment of visual attractiveness contributes to elaborate (reflective) judgment of aesthetics [17]. Therefore,

– Appearance = F(Visual attractiveness)

(Immediate) memories of experience contribute to global judgment of experience [13]. Therefore,

– Positive memories = F(Positive affect, Negative affect)

Autotelic experience in a particular interaction episode is an 'intrinsically rewarding experience' [9] and therefore produces pleasure that contributes to a global judgment of pleasure of interaction. Utility and appearance contribute to pleasure [16]. Therefore,

– Pleasure = F(Autotelic experience, Utility, Appearance)

Pleasure partially mediates the effect of utility on satisfaction and fully mediates the effect of appearance [16]. Therefore,

– Product Satisfaction = F(Pleasure, Utility, Appearance)

The effects of utility and appearance on product attachment are fully mediated by pleasure [16]. Positive memories have a positive effect on pleasure [16]. The effects of utility and appearance on product attachment are moderated by positive memories [16]. Therefore,

– Product Attachment = F(Pleasure, Utility, Appearance, Positive Memories)

Satisfaction is an antecedent of intention [15]. Therefore,

– Intention of Future purchase = F(Satisfaction)

By combining the relations specified above, we may present the static BN for the UX in Fig. 1.

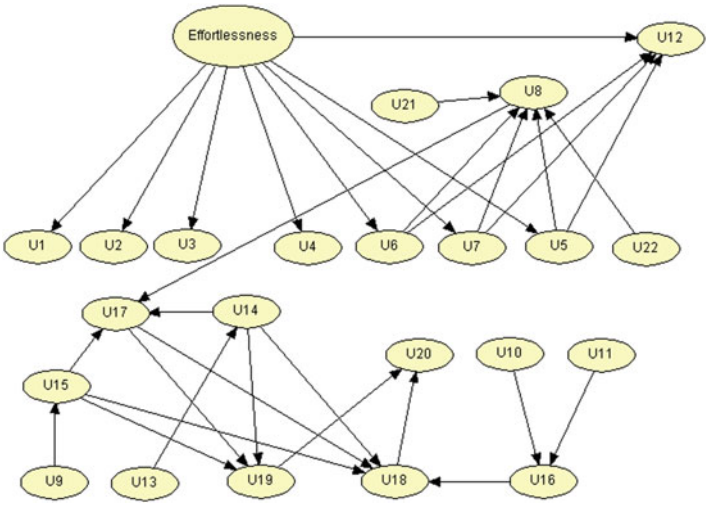


Fig. 1. A static BN represents the UX.

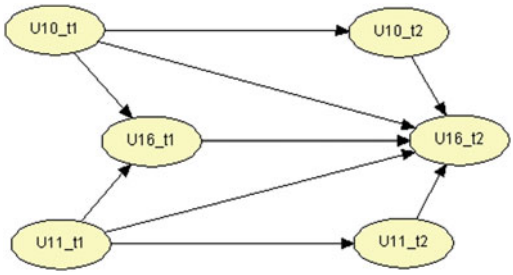


Fig. 2. A dynamic BN represents one relation over time.

3.3 Dynamic BN

To construct a time-dependent framework, we assume a first-order Markov process: the previous experience is an antecedent of next experience. Therefore, each experience variable at the previous time ($t1$) is treated a cause of the same variable at the next time($t2$). The relations (indicated by **F**) can also turn into a first-order Markov process. For example, we may represent the relation below over time in Fig. 2.

– Positive memories = F(Positive affect, Negative affect)

Obviously, the complete DBN will be a very complicated model where all relations are expanded over time. We will further test whether any descendants of the antecedents in the relations are statistically significant. By doing this, we expect to simplify the model by reducing the connectivity over time.

3.4 Discussions

Complicated UX relations always puzzle both domain experts and practitioners as the dimensions grow over time. Resorting to probabilistic graphical models, we intend to provide a more intuitive representation to describe UX over time. Particularly, the model becomes an easy way to convey UX to product designers who can understand the studied domain through a formal language.

By exploiting the previous study on UX, we structure UX variables into one BN and expand the BN into DBN assuming a first-order Markov process. The remaining thing is to specify DBN parameters (conditional probability tables) that normally can be done in an automatic way. Currently, we are gathering domain data from the field study and expect to estimate the parameters through a proper learning method.

4 Conclusions and Future Work

Modeling the time course of UX is important, but an under-researched field of study. The use of DBN is a promising approach to modeling UX over time, but this work needs to be informed by and account for existing theoretical frameworks and new ideas. This will allow existing theories to be refined or replaced by new theories. We have demonstrated the feasibility and limitations of using DBN to model UX. Most important findings were that UX relations can be explicitly represented through BN and can be intuitively understood by researchers and practitioners without different knowledge background; however, learning BN parameters could be a potential issue as a sufficient amount of data shall be gathered. We will exploit domain knowledge to develop a more reliable and efficient learning process.

Future work will include testing the UX flow model that has been presented here in experiments with different products where (the result of) task performance is essential. Furthermore, it is important to realize that modeling needs to be flexible to select or develop UX models based on the outcome variables that are of interest in terms of explanation or prediction, instead of using a single 'one-size-fits-all' approach. Depending on target UX outcomes and the role of task performance for particular products, different models need to be formulated and tested for theoretical understanding and as a basis for design improvement.

References

1. Watson, D., Clark, L.A., Tellegen, A.: Development and validation of brief measures of positive and negative affect: the panas scales. J. Pers. Soc. Psychol. **54**(6), 1063–1070 (1988)
2. de Manzano, O., Theorell, T., Harmat, L., Ullén, F.: The psychophysiology of flow during piano playing. Emotion **3**(10), 301–311 (2010)
3. Dillon, A.: Beyond usability: process, outcome and affect in human-computer interactions. Can. J. Libr. Inf. Sci. **4**(26), 57–69 (2001)

4. Engeser, S., Rheinberg, F.: Flow, performance and moderators of challenge-skill balance. Motiv. Emot. **32**(3), 158–172 (2008)
5. Wulf, G., Lewthwaite, R.: Effortless motor learning? an external focus of attention enhances movement effectiveness and efficiency. In: Bruya, B. (ed.) A New Perspective in Attention and Action, pp. 75–101. MIT Press, Cambridge (2010)
6. Guo, Y.M., Poole, M.S.: Antecedents of flow in online shopping: a test of alternative models. Inf. Syst. J. **19**(4), 369–390 (2009)
7. Hassenzahl, M.: The thing and i: understanding the relationship between user and product. In: Blythe, M.A., Overbeeke, K., Monk, A.F., Wright, P.C. (eds.) Funology, pp. 31–42. Kluwer Academic Publishers, Norwell (2003)
8. Hassenzahl, M.: The interplay of beauty, goodness, and usability in interactive products. Hum. Comput. Interact. **19**(4), 319–349 (2004)
9. Jackson, S.A., Marsh, H.W.: Development and validation of a scale to measure optimal experience: the flow state scale. J. Sport Exerc. Psychol. **18**(1), 17–35 (1996)
10. Diefendorff, J.M., Hall, R.J., Lord, R.G., Strean, M.L.: Action-state orientation: construct validity of a revised measure and its relationship to work-related variables. J. Appl. Psychol. **85**(2), 258–263 (2000)
11. Mitra, K., Zaslavsky, A., Ahlund, C.: Dynamic bayesian networks for sequential quality of experience modelling and measurement. In: Proceedings of the 11th International Conference and 4th International Conference on Smart Spaces and Next Generation Wired/Wireless Networking, Anonymous Smart Spaces and Next Generation Wired/Wireless Networking, pp. 135–146 (2011)
12. Keller, J., Bless, H.: Flow and regulatory compatibility: an experimental approach to the flow model of intrinsic motivation. Pers. Soc. Psychol. Bull. **34**, 196–209 (2008)
13. Kim-Prieto, C., Diener, E., Tamir, M., Scollon, C., Diener, M.: Integrating the diverse definitions of happiness: a time-sequential framework of subjective well-being. J. Happiness Stud. **3**(6), 261–300 (2005)
14. Kowatsch, T., Maass, W.: In-store consumer behavior: how mobile recommendation agents influence usage intentions, product purchases, and store preferences. Comput. Hum. Behav. **26**(4), 697–704 (2010)
15. Lin, W.-S.: Perceived fit and satisfaction on web learning performance: is continuance intention and task-technology fit perspectives. Int. J. Hum. Comput Stud. **70**(7), 498–507 (2012)
16. Mugge, R., Schifferstein, H.N.J., Schoormans, J.P.L.: Product attachment and satisfaction: understanding consumers' post-purchase behavior. J. Consum. Mark. **27**(3), 271–282 (2010)
17. Tractinsky, N., Cokhavi, A., Kirschenbaum, M., Sharfi, T.: Evaluating the consistency of immediate aesthetic perceptions of web pages. Int. J. Hum. Comput. Stud. **64**(11), 1071–1083 (2006)
18. Porat, T., Tractinsky, N.: Its a pleasure buying here: the effects of web-store design on consumers' emotions and attitudes. Hum. Comput. Interact. **27**(3), 235–276 (2012)
19. Spears, I.: Development and evaluation of an 'exergaming' intervention to target cardio-vascular and quality of life outcomes in a deprived area of the north-east. Grant Ref: EP/I001891/1. Research Councils UK Digital Economy Programme (2010)
20. Sutcliffe, A.: Designing for User Engagement: Aesthetic and Attractive User Interfaces. Morgan and Claypool, San Rafael (2009)

21. Hartmann, J., Sutcliffe, A., De Angeli, A.: Towards a theory of user judgment of aesthetics and user interface quality. ACM Trans. Comput. Hum. Interact. **15**(4), 15:1–15:30 (2008)
22. Thüring, M., Mahlke, S.: Usability, aesthetics and emotions in human-technology interaction. Int. J. Psychol. **4**(42), 253–264 (2007)
23. van Schaik, P., Ling, J.: A cognitive-experiential approach to modelling web navigation. Int. J. Hum. Comput Stud. **9**(70), 630–651 (2012)
24. Bai, Z., Blackwell, A.F.: Analytic review of usability evaluation in ismar. Interact. Comput. **24**(6), 450–460 (2012)

Multi-Agent Joint Learning from Argumentation

Junyi Xu(✉), Li Yao, Le Li, and Jinyang Li

Science and Technology on Information Systems Engineering Laboratory,
National University of Defense Technology, Changsha, China
xujunyi0923@163.com

Abstract. Joint learning from argumentation is the idea that groups of agents with different individual knowledge take part in argumentation to communicate with each other to improve their learning ability. This paper focuses on association rule, and presents MALA, a model for argumentation based multi-agent joint learning which integrates ideas from machine learning, data mining and argumentation. We introduce the argumentation model Arena as a communication platform with which the agents can communicate their individual knowledge mined from their own datasets. We experimentally show that MALA can get a shared and agreed knowledge base and improve the performance of association rule mining.

Keywords: Argumentation · Data mining · Association rule · Multi-agent learning

1 Introduction

With the rapid development of data mining and knowledge discovery technology, people can get potential knowledge in large amount of data through data mining techniques. However, the knowledge gained by mining is too lengthy and jumbled, so it is difficult for users to filter and apply the knowledge in problem solving. As an important branch of data mining, association rule mining also has this bottleneck in practical application. In order to solve this problem, some researchers have integrated argumentation theory in artificial intelligence with data mining technology to improve the quality of data mining [1,2].

As the experience knowledge mined by individual Agent is incomplete and maybe defective, thus Multi-Agent Joint Learning or agent mining [12] can optimize the experience knowledge to obtain high-quality experience rules for groups to share. From the perspective of joint learning, this paper attempts to apply argumentation theory to distributed association rule mining problem using the idea of "joint learning from argumentation" and proposes an argumentation based multi-agent learning approach MALA. Our experiments show that: argumentation-based joint learning method can effectively achieve reasonable knowledge assessment and optimization in association rule mining and enhance the quality of data mining.

L. Cao et al. (Eds.): ADMI 2013, LNAI 8316, pp. 14–25, 2014.
DOI: 10.1007/978-3-642-55192-5_2,

The paper is organized as follows. Section 2 provides a quick overview of related work. Section 3 formally proposes the new idea of "joint learning from argumentation". Next, Sect. 4 presents MALA-Arena, a model of multi-agent joint learning from argumentation using Arena. After that, Sect. 5 introduces a dialectic analysis model Arena which is used for multi-agent argumentation in MALA. Finally, Sect. 6 presents an experimental evaluation of our model. The paper closes with conclusions.

2 Related Works

Recent years, a number of different approaches have been proposed to integrate argumentation and machine learning. Governatori and Stranieri investigate the feasibility of KDD in order to facilitate the discovery of defeasible rules for legal decision making [3]. In particular they argue in favor of Defeasible Logic as an appropriate formal system in which the extracted principles should be encoded in the context of obtaining defeasible rules by means of induction-based techniques.

The idea that argumentation might be useful for machine learning was discussed in [4], since argumentation could provide a sound formalization for both expressing and reasoning with uncertain and incomplete information. Since the possible hypotheses induced from data could be considered an argument, and then by defining a proper attack and defeat relation, a sound hypotheses can be found.

Ontan and Plaza in [5] research concept learning, and put forward a multi-Agent inductive learning framework A-MAIL, which integrates inductive learning, case-based reasoning and argumentation theory. In this framework, Multi-Agent Inductive Learning consists of three stages: individual induction; argumentation process; and belief revision. The proposed method is different from ours. In A-MAIL, each Agent just use argumentation based inductive learning to revise their own knowledge and multi-Agent system do not form a shared knowledge base. Moreover, A-MAIL focus-es on inductive learning while MALA focuses on association rules.

Maya proposes argumentation from experience in [6], and combines argumentation theory with data mining techniques. Agent gets association rules as their arguments in the library of their own experience through data mining. PADUA argumentation model is designed to achieve two party argumentation processes and resolve uncertainties classification problems. Later, PISA model is designed in [7] in order to solve the multi-classification problem. However, PISA has complicated strategy and complex argumentation process, so that the model does not have general applicability. Subsequently, the concept of collaborative group of Agents is proposed for arguing from experience in [8].

In order to enhance the versatility of PISA, Maya simplifies the speech acts and removes a complex strategy in argumentation in [9]. The improved model can be used to solve the following problem in classification: multi-agent classification, ordinal classification and imbalance classification. Although the simplified model improves the versatility, its classification accuracy is decreased.

In this paper, Multi-Agent joint learning from the argumentation model MALA is different from PISA model. PISA model focuses on classification problem and its goal is to improve the classification accuracy through multi-Agent argumentation, while the purpose of MALA is to realize knowledge sharing in distributed data mining. Argumentation in PISA is driven by the target of classification while MALA is driven by association rule.

3 Joint Learning from Argumentation

As guided by the "Knowledge spiral" model, this paper will apply argumentation theory to distributed association rule mining issues and propose a new method of "joint learning from argumentation". This section briefly describes the principle of the method.

3.1 A Knowledge Spiral Mode

Nonaka designed a knowledge spiral model (see Fig. 1) in knowledge management area [10]. The knowledge spiral shows how organizations extract shared explicit knowledge from individual tacit knowledge. Organizations develop tools and models to accumulate and share knowledge from individuals. The knowledge spiral is a continuous activity of knowledge flow, extraction, and sharing by individuals, groups, and organizations. Knowledge spiral starts at the individual level and moves up to the organizational level through expanding communities of interaction. Nonaka argues that an organization has to promote a facilitating context in which the organizational knowledge-creation process can easily take place. Learning jointly from argumentation can achieve the organizational knowledge-creation process.

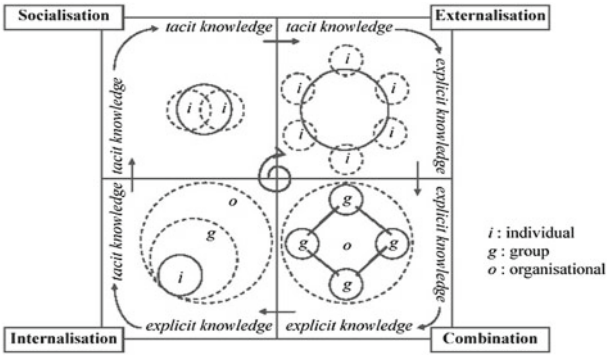


Fig. 1. The knowledge spiral model [10]

From the knowledge spiral model we can find: Individuals with the same task in an organization can obtain group knowledge with consensus through

mutual communication. These common knowledge as explicit knowledge will further enhance the individuals' ability to solve new tasks. Then new individual knowledge will be exchanged again to form higher quality consensus knowledge, so as to achieve further knowledge sharing and application. Knowledge spiral model indicates the mutual transformation of individual knowledge and group knowledge, explicit knowledge and tacit knowledge, as well as spiral development process of knowledge evolvement.

3.2 An Approach to Joint Learning from Argumentation

In Multi-agent system, the local knowledge of single Agent is limited; as a result their problem-solving ability is limited. In order to effectively organize and optimize knowledge of multi-agent system to enhance the overall capacity of multi-Agent system, we need to optimize and share individual knowledge. However, individual Agent has different knowledge, and such knowledge is likely tacit, which led to difficulties in knowledge extraction and sharing.

In response to the problem, this paper proposes the idea of joint learning from argumentation (Multi-Agent Learning jointly from Argumentation, MALA) guided by "Knowledge spiral" model. MALA method divides the learning process into three stages: the individual association rule mining, multi-agent argumentation and the group knowledge extraction, as shown in Fig. 2.

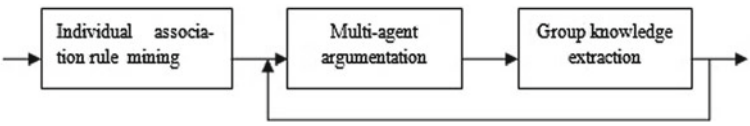


Fig. 2. Learning process in MALA

In the stage of individual association rule mining, each Agent first perform the ex-tended association rule mining in local experience dataset, and form the local experience Knowledge Base in the form of Experience Argument Schema (EAS) [11]. Through data mining technology, we can find the potential knowledge of individuals and realize externalization of tacit Knowledge in individual Agent, and use EAS to represent experience knowledge.

In the stage of Multi-agent argumentation, we use argumentation techniques to achieve mutual learning between Agents. For the same case, Agent uses EAS as the main form of the argument on argumentation platform to express their views and to communicate and compare their local experience knowledge with the other Agents. Through argumentation, experience knowledge of high quality with consensus can be formed. So argumentation model can provide a platform for Multi-Agent System to communicate and discuss individual experience knowledge. Agents can analysis and discuss a specific topic to reach the consensus.

In the stage of Knowledge extraction, the outcome of argumentation is clearly represented to form the shared explicit knowledge, and stored in the shared global knowledge base. In the following argumentation, each Agent will use shared know-ledge and local experience knowledge to argue.

The method of "joint learning from argumentation" can effectively merge the local experience knowledge of individual Agent: Individual Agent can realize the function of individual knowledge externalization by association rule mining; through the process of multi-Agent argumentation, individual Agents with different knowledge can interact and communicate with each other so as to reach consensus, and realize the transformation of individual knowledge into organizational knowledge; Ultimately, the shared knowledge of multi-Agent System further guide individuals of following problem solving and continue accumulation and refinement to form the spiral evolution process.

4 Realizing MALA Using ARENA

According to the above approach of joint learning from argumentation, we design a model of multi-agent joint Learning from Argumentation using Arena, called MALA-Arena.

In MALA-Arena, multi-agent system first performs association rule mining on distributed datasets and individual Agents form their independent local knowledge bases. Given a set of Agent $A = \{A_1, \ldots, A_m\}$, acquisition of Agents local knowledge is built on the basis of association rule mining. Each Agent A_i has a separate example dataset $D = \{d_1, \ldots, d_n\}$. In order to achieve a unified knowledge form, each Agent A_i uses the same association rule mining algorithm in each example dataset, and takes the support and confidence measure to assess the pros and cons of association rules. By association rule mining, each Agent forms their local knowledge base EAS. Agent's local knowledge base can be expressed as a set of Experience Argument Schema (EAS) $EAS = \{eas_1, \ldots, eas_n\}$.

There are inconsistencies between datasets D_i of each Agent which result in inconsistent knowledge in each Agents local knowledge bases EAS_i. In order to effectively integrate the inconsistent knowledge, we can use the method of multi-Agent argumentation. On this basis, we design a multi-agent argumentation model Arena, which transforms the multi-party argumentation process into two-party argumentation processes to achieve assessment and screening of association rules. To a specific topic t_i, Agent can use their own Experience Argument Schema (EAS) on the Arena to construct arguments and attack relations to argue with other Agents. After the end of argumentation, the main argument of winner becomes the valuable knowledge k_i.

For the valuable knowledge k_i get from the current argumentation, multi-agent system needs feedback. According to the correct classification result of current case t_i, system will determine whether the valuable knowledge is consistent with the correct result. If the result is consistent, the valuable knowledge k_i will be stored in the global knowledge base K; Otherwise, Multi-Agent

System will discard the knowledge. Through a large number of training cases, Multi-Agent System can accumulate a focused set of association rules by using "learning from argumentation" and eventually form a shared global knowledge base $K = \{k_1, \ldots, k_n\}$.

The main process of MALA-Arena is as follows:

1. Agent A_i gets local knowledge base EAS_i by association rules mining on his own dataset D_i. Knowledge in local knowledge base is in the form of Experience Argument Schema (EAS);
2. For a specific input case t , each Agent uses their own EAS to generate argument eas_i in the current argumentation on Arena;
3. After the end of current argumentation, multi-agent system can get a valuable rule k;
4. Feedback process: to determine whether the current case t can be correctly classified by the valuable rule k according to the known result of classification;
5. If correctly classified, the valuable rule k will be stored in the global knowledge base K as a multi-agent shared knowledge; if classification is not correct, it means this rule is flawed, not to join the global knowledge base;
6. Repeat the learning process 2–5, and the shared knowledge in the global knowledge base K continue to accumulate, eventually converge to a stable state.

The brief algorithm of MALA-Arena is as follows:

```
Algorithm  MainControl of  MALA-Arena
Input: Training Set T
For each(Ai)do
EASi = Associasion_Rule_Mining(Di);
While (ti in T) do // there are still other input data
k = Arena(ti, EAS) //argumentation in Arena
{ Broadcast ti;
  Get_Participants (Qp); // getting participants from queue of
  Agents
  Initial (grid of dialectical analysis trees);
  For each participant Pi do
    Propose_Argument (Pi, easi);
    Change the speak token;
  End for
  If Pi == silence then
    select next participant Pi+1;
  end if
  If only Pi == active then
    Pi == winner;
    Return (k);
  End if
} // The argument game is over
```

```
bool i = Verify(k, ti);
if i == true then
  Add_To_Knowledge_Base(k);
else if i == false then
//do nothing
End if
End while
K = Get_Knowledge_Base();
Return (K);
Output: Knowledge base K
```

5 Argumentation Model Arena

Arena is a dialectic analysis model for multiparty argument games (more details in [11]). In Arena, we designed four roles: Referee, Master, Challenger and Spectator. In Arena, all the arguments between the Master and the Challenger are about the association rules. The whole process of argumentation is stored in the grid of dialectic analysis trees.

In Arena, the Referee doesn't participate in argumentation but manages the argumentation process according to the dialogue rules of Arena. And there can be only one Master and one Challenger to take part in the argumentation, while other participants are not allowed to speak when they are just Spectator.

The Referee is a neutral agent which manages a variety of tasks to facilitate multi-party dialogues from experience. It has following responsibilities: Starting a dialogue; Identifying the roles of Master, Challenger, and Spectator along with the change of the game situation; Monitoring the dialogue; Maintaining the dialectic analysis tree to reflect the moves made by the masters or the challengers; Terminating the dialogue once a termination condition is satisfied; Announcing the games winner, his opinion and the valuable experience rule.

Participant Agents can produce arguments in form of Experience Argument Schema EAS from local knowledge base. Suppose that x represents the case under discussion. EAS is defined as follows: Conclusion: $w(x)$; Premises: $l_1(x), l_2(x), \ldots, l_n(x)$; Confidence: c; Conditions: $u_1(x), u_2(x), \ldots, u_s(x); \neg v_1(x)$, $\neg v_2(x), \ldots, \neg v_t(x)$; Exceptions: $e_1, \ldots, e_k$. Such argument schema for experience can be read as follows: In my experience, if anything x doesnt belong to $\{e_1, \ldots, e_k\}$, with features $u_1, u_2, \ldots, u_s$ and not with features $v_1, v_2, \ldots, v_t$, then x with features $l_1, l_2, \ldots, l_n$, are Ws (or have feature W) with probability c.

In Arena, all the participating agents will play a role of Master, Challenger and Spectator. During an argumentation, the participating agents need to compete for Master or Challenger continually with his own set of EASs. Once Master and Challenger are identified, the agents can use one of the six speech acts, which collectively form the basic building blocks for constructing Master-Challenger dialogues in Arena.

In Arena, there are also six speech acts in Arena: ProposeOpinion, Distinguish, Counter Rule, BeInapplicable, BeAnException, and Defeated. These speech acts fall under three basic types: stating a position, attacking a position and conceding defeated, as follows (Table 1):

Table 1. Speech acts in Arena

Type	Speech acts	Content
Stating position	ProposeOpinion	Proposing the opinion about the case under discussion according to a new EAS with highest confidence from his local knowledge
Attacking position	Distinguish	Addition of new premise(s) to a previously proposed EAS, so the confidence of the new rule is lower than the original one
	CounterRule	Using a new EAS with higher confidence to attack the conclusion or the confidence of the adversarys EAS
	BeInapplicable	Stating that the EAS of the adversarys argument is inapplicable to this case in his own knowledge
	BeAnException	Stating that the case under consideration is an exception of the EAS in his own knowledge
Conceding defeated	Defeated	Stating that the player concedes defeated

At the beginning of the argumentation, the Referee broadcast the discussion topic, and the first agent who proposes its opinion about the current topic becomes the Master of Arena. All the other participants whose option is different from the Master can challenge the Master and form the queue of challengers, and the first participant in the queue is selected to be the Challenger of Arena. All the other participant agents except Master and Challenger are Spectator of Arena.

Noted that during the argumentation the Spectator can apply for Master or Challenger at any moment, and the Referee just put its argument in the application queue. Since the defeated argument of the old Master cant be used to apply for Master again, the old Master may produce another argument for the instance under discussion, and uses this new argument to apply for Master once more. In addition the defeated Challenger has no chance to challenge the same Master again.

If the Master is defeated by the Challenger, this Challenger will become the new Master, and he can propose his opinion about the current topic from his own knowledge base. All the other participants decide whether or not to challenge this new option. Otherwise, if the Challenger is defeated, the next participant in the queue is selected to be the Challenger, and the argumentation between

Master and Challenger continues. If the Master can defeat all the challengers, the Master wins the argumentation and the Masters association rule will be considered a valuable rule.

There is a termination condition: the queue of Challenger is empty or Master is empty. When Master isn't empty, the Match has a winner. Otherwise, the Match is tie. Since the number of the arguments produced by a participant is finite and the defeated arguments cant be allowed to use repeatedly, the termination of the game is thus guaranteed (Fig. 3).

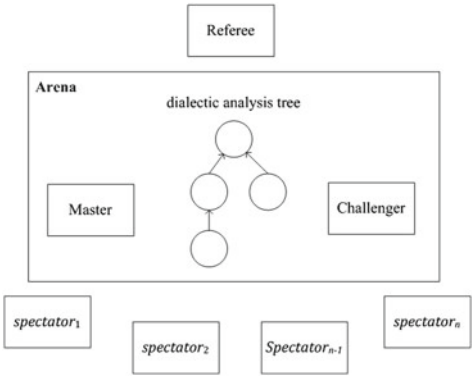


Fig. 3. The basic structure of Arena model

6 Experiments

In order to empirically evaluate MALA-Arena we use three machine learning data-sets: nursery, scale and Tie-Tac-Toe from the UCI Machine Learning Repository[1]. The nursery dataset contains 12960 examples belonging to 5 different classes. The scale dataset contains 625 examples belonging to 3 different classes. The Tie-Tac-Toe dataset contains 958 examples belonging to 2 different classes. In the experiment, we use 4 Agents to take part in MALA-Arena. And all records of each datasets are divided into four parts equally, which belongs to four Agents respectively. We use Agent1, Agent2, Agent3 and Agent4 to represent these agents. Each Agent produces his association rules in the form of EAS with the confidence level to 50 % and the support level to 1 % using Apriori-TFP data mining Algorithm [13].

To evaluate MALA-Arena, we used 10 fold cross validation (TCV) test on each dataset. In an experimental run, we use the training set to form the sharing knowledge base, which will be evaluated using the test set. For each dataset, we report the average results for each group of TCV test.

We compared the results of MALA-Arena with respect to the result of centralizing all the examples and performing centralized association rule mining algorithm TFPC [14]. Thus, the difference between the results of TFPC

[1] UCI machine learning repository: http://archive.ics.uci.edu/ml/datasets.

and agents using MALA-Arena with Apriori-TFP should provide a measure of the benefits of MALA-Arena, whereas comparing with centralized association rule mining algorithm gives a measure of the quality of MALA-Arena outcome. Table 2 shows a row for each of the data sets we used in our evaluation. Performance is measured using accuracy in classification. Analyzing the results in Table 2 we can see that accuracy of MALA-Arena is more than 80 %, while TFPC is below 70 %. MALA-Arena can greatly increase the accuracy over the TFPC algorithm in three datasets. This shows that MALA-Arena successfully integrates argumentation and association rule mining, and allows agents to learn highly accurate knowledge without requiring the centralization of all data.

Moreover, from Table 3 we can see that the number of valuable rules generated by MALA-Arena is much smaller than the number of association rules mined by individual Agents from their own example bases. The average number of rules in knowledge base generated by MALA-Arena is almost lower than 100, while there are thousands of rules of each Agent in nursery and Tie-Tac-Toe datasets. So MALA-Arena can be a filter to control the size of knowledge from association rule mining and increase the quality of knowledge base.

Table 2. Accuracy of MALA-Arena and TFPC in different datasets

Accuracy	Nursery (%)	Scale (%)	Tie-Tac-Toe (%)
MALA-Arena	94	81.1	86.2
TFPC	63.53	65.26	60.96

Table 3. Number of association rules (ARs) of different knowledge bases in different datasets

Number of ARs	Agent1	Agent2	Agent3	Agent4	MALA-Arena
Nursery	1769	1802	1765	1781	79.5
Scale	318	297	280	295	102.7
Tie-Tac-Toe	9238	9590	9346	9396	70.3

In summary, we can conclude that MALA-Arena successfully achieves multi-agent joint learning from argumentation, since performance is outstanding from the TFPC approach. Moreover, this is achieved extract a small size of knowledge from individual Agents to get a high accuracy. Additionally, on average, the number of rules of MALA-Arena is much lower than that of individual Agents, which is interesting since it could be used to improve the quality of data mining, by distributing the task among several agents, later arguing about their local knowledge and finally forming a focused sharing knowledge base.

7 Conclusion

In this paper, we have proposed the theory of joint learning from argumentation which provides a new way to evaluate and share the knowledge mined from different databases and demonstrates a fact that a combined analytical and inductive machine learning method could overcome the pitfalls in each separate approach.

This paper has presented MALA, an approach to Multi-Agent Learning jointly from Argumentation. The key idea is that argumentation can be used as a formal learning framework to exchange and discuss the local knowledge learnt by agents using association rule mining. In our experiments, we designed and realized MALA-Arena. Multi-agent joint learning from argumentation is performed by three processes: individual association rule mining, multi-agent argumentation and know-ledge extraction. The results of experiments reveal MALA-Arena has an effective capability in learning from argumentation and the final sharing knowledge from MALA-Arena can perform well. Finally, our approach is focused on association rule mining, and future work aims at other data mining methods to integrate in the model for joint learning from argumentation.

References

1. Možina, M., Žabkar, J., Bench-Capon, T., Bratko, I.: Argument based machine learning applied to law. Artif. Intell. Law **13**(1), 53–73 (2005)
2. Ontanón, S., Plaza, E.: Arguments and counterexamples in case-based joint deliberation. In: Maudet, N., Parsons, S., Rahwan, I. (eds.) ArgMAS 2006. LNCS (LNAI), vol. 4766, pp. 36–53. Springer, Heidelberg (2007)
3. Governatori, G., Stranieri, A.: Towards the application of association rules for defeasible rules discovery. In: Jurix 2001, pp. 63–75 (2001)
4. Gómez, S.A., Chesnevar, C.I.: Integrating defeasible argumentation and machine learning techniques. arXiv preprint: cs/0402057 (2004)
5. Ontañón, S., Plaza, E.: Multiagent inductive learning: an argumentation-based approach. In: Proceedings of the ICML-2010, 27th International Conference on Machine Learning, pp. 839–846 (2010)
6. Wardeh, M., Bench-Capon, T., Coenen, F.: PADUA: a protocol for argumentation dialogue using association rules. Artif. Intell. Law **17**(3), 183–215 (2009)
7. Wardeh, M., Bench-Capon, T., Coenen, F.: Multi-party argument from experience. In: McBurney, P., Rahwan, I., Parsons, S., Maudet, N. (eds.) ArgMAS 2009. LNCS (LNAI), vol. 6057, pp. 216–235. Springer, Heidelberg (2010)
8. Wardeh, M., Bench-Capon, T., Coenen, F.: Arguing from experience using multiple groups of agents. Argum. Comput. **2**(1), 51–76 (2011)
9. Wardeh, M., Coenen, F., Bench-Capon, T., Wyner, A.: Multi-agent based classification using argumentation from experience. In: Huang, J.Z., Cao, L., Srivastava, J. (eds.) PAKDD 2011, Part II. LNCS (LNAI), vol. 6635, pp. 357–369. Springer, Heidelberg (2011)
10. Nonaka, I., Takeuchi, H.: The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation. Oxford University Press, Oxford (1995)

11. Yao, L., Xu, J., Li, J., Qi, X.: Evaluating the valuable rules from different experience using multiparty argument games. In: IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology: WI-IAT12, Macao, China (2012)
12. Cao, L., Gorodetsky, V., Mitkas, P.: Agent mining: the synergy of agents and data mining. IEEE Intell. Syst. **24**(3), 64–72 (2009)
13. Coenen, F., Leng, P., Ahmed, S.: Data structure for association rule mining: t-trees and p-trees. IEEE Trans. Knowl. Data Eng. **16**(6), 774–778 (2004)
14. Coenen, F.: The LUCS-KDD TFPC classification association rule mining algorithm. University of Liverpool, Department of Computer Science (2004)

Towards Mining Norms in Open Source Software Repositories

Bastin Tony Roy Savarimuthu[1(✉)] and Hoa Khanh Dam[2]

[1] University of Otago, P.O. Box 56, Dunedin, New Zealand
tonyr@infoscience.otago.ac.nz
[2] School of Computer Science and Software Engineering,
University of Wollongong, Wollongong, NSW 2522, Australia
hoa@uow.edu.au

Abstract. Extracting norms from computer-mediated human interactions is gaining popularity since huge volume of data is available from which norms can be extracted. Open source communities offer exciting new application opportunities for extracting norms since such communities involve developers from different geographical regions, background and cultures. Investigating the types of norms that exist in open source projects and their efficacy (i.e. *the usage of norms*) in enabling smoother functioning however has not received much attention from the normative multi-agent systems (NorMAS) community. This paper makes two contributions in this regard. First, it presents norm compliance results from a case study involving three open source Java projects. Second, it presents an architecture for *mining* norms from open source projects. It also discusses the opportunities presented by the domain of software repositories for the study of norms. In particular, it points towards how norms can be mined by leveraging and extending prior work in the areas of Normative Multi-Agent Systems (NorMAS) and mining software repositories.

1 Introduction

A good example of a large, real-world multi-agent organization involving a huge number of (human) agents is open source software development (OSSD). In fact, large-scale open source systems such as Linux and Android OS, used and tested by millions are developed by hundreds of contributors over extended period of time. Many large open source projects are highly successful although they did not *initially* have formal organizational structure with regulations (i.e. norms and policies) explicitly stated and enforced (as in traditional commercial software projects). Therefore, it is important and interesting to explore the role norms play in the success (or failure) of a particular open source project, and how these norms have emerged and are enforced in such large organizations of volunteering

An early (short) draft of this paper appears as a working paper (an informal publication) at Otago (http://otago.ourarchive.ac.nz/handle/10523/2101) and was orally presented at the 2012 Dagstuhl seminar on NorMAS.

L. Cao et al. (Eds.): ADMI 2013, LNAI 8316, pp. 26–39, 2014.
DOI: 10.1007/978-3-642-55192-5_3,

contributors who have different experience and are from different background, cultures and geographical regions.

Mining norms in open source projects is facilitated by the rich, extensive and readily available data from the software repositories of those projects. Software repositories can be in various forms such as historical repositories (e.g. SVN or CVS repositories, archived communication records, bug repositories), code repositories (e.g. Sourceforge.net and Google code), and run-time repositories (e.g. deployment and/or execution logs). Since these repositories contain information about human actions and their interactions with one another, these repositories contain explicit or implicit information on *norms* relevant to the communities involved in the process of software development. These repositories can be *mined* to uncover useful and important patterns and information about the normative processes associated with the development of software systems. For instance, we can directly observe developer discussions, identify their contents (e.g. patches, bugs, reviews) on mailing lists or forums. We can build social networks, and cross-check associated discussion and programming activity. In addition, we can leverage existing mining software repositories (MSR) technologies [8] such as data pre-processing, cross-linking data from different sources for mining norms.

Although OSSD offers a real, large-scale platform for norm mining, to the best of our knowledge, there has been no existing work in the study of norms in open source software projects. Thus, this line of work is a promising candidate for NorMAS researchers to investigate the processes associated with real norms such as norm formation, spreading and enforcement. Insights gained from this study can be used to inform both research and practice of norms. Moreover, this research can leverage existing techniques developed by NorMAS researchers, data mining researchers and other computer science disciplines such as information retrieval and natural language processing.

The work in this paper specifically contributes to call for identifying challenges and future research directions on the synergy between agents and data mining [4]. Towards that goal, this paper explores how norms govern the smoother functioning of large, distributed OSSD projects (which are real-life, large-scale multi-agent organizations). Towards the understanding of normative processes in OSSD, this paper makes two contributions. First, it discusses a case study involving three open source projects to investigate norm compliance. Second, it presents an architecture for mining norms in open source software repositories and identifies research challenges that can be addressed using the proposed architecture.

2 Background

Norms are expectations of behaviour in an agent society. Researchers in normative multi-agent systems (NorMAS) study how norms can play a role in the design and the operationalization of socio-technical systems. Research in normative multi-agent systems can be categorized into two branches. The first branch

focuses on normative system architectures, norm representations, norm adherence and the associated punitive or incentive measures. The second branch is concerned with the emergence of norms. For an overview of the study of norms in agent systems we refer the reader to [6,14]. In this section, we provide a brief overview of the relatively new domain of mining software repositories, and particularly how norms can be mined from huge volumes of data.

2.1 Mining Software Repositories

Mining Software Repositories (MSR) [8] is an emerging research area that focuses on mining the rich and large quantities of data available in software repositories to uncover useful and important patterns and information about software systems and projects. Such information assists developers, managers, testers, etc. working on those systems by offering insights into the nature of open source software development (OSSD) through the development techniques and tools.

Efforts in MSR[1] research have been mainly on providing techniques to extract and cross-link important information from different software repositories. Such information can be used in various activities during the development of software systems. For instance, a range of work (e.g. [19]) have proposed to make use of historical co-changes (e.g. entities or artefacts that were changed together) in a software project to propagate changes across the software system during maintenance and evolution. A large number of existing MSR work (e.g. [13]) also mine bug reports and historical changes to predict the occurrence of bugs, e.g. parts of the code that likely to contain errors. Such predictions are useful for managers in allocating and prioritizing testing resources. Information mined from reported bugs and execution logs can also be used to improve the user experience, e.g. warning the user when they are about to perform a buggy action, and suggesting when an existing piece of code can be re-used. Empirical software engineering also substantially benefits from MSR since many empirical studies can be done (and repeated) on a large number of subjects, i.e. OSS repositories enable the verification of generality of prior findings (e.g. the study in [9] confirms that cloning seems to be a reasonable or even beneficial design option in some situations). Recent MSR work (e.g. [1]) has also attempted to mine discussions from archived mailing lists, forums, and instant messaging to understand the dynamics of large software development teams, including how and when team members get invited, detecting team morale at a particular point in time, and understanding the process of bug triage.

2.2 Mining Norms from Large Repositories

Based on a large Twitter dataset of 1.7 billion tweets [11], researchers have investigated how two out of seven independently proposed re-tweeting conventions became widely adopted. Their main finding was that social conventions are

[1] A extensive review of the work in MSR can be found from the "Bibliography on Mining Software Engineering Data" available at http://ase.csc.ncsu.edu/dmse.

more likely to arise in the active and densely connected part of the community. However, the study was unable to ascertain why some conventions were widely adopted and why some were not.

Another research [3] has investigated the naming conventions used for Java classes to check whether the names correspond to the actual recommendation provided by the Java naming convention (i.e. they should be noun phrases) and have proposed whether the class names have to be changed to adhere to this recommendation or whether the class itself has to be refactored. The work of Boogerd and Moonan [2] notes that following coding conventions can increase the chance of faults occurring since any modification of the code to adhere to a convention has a non trivial probability of introducing errors. This finding is interesting, however, the result is based on investigating data from one project only. To our knowledge, there has been no prior work on mining different categories of conventions in open source projects. Also, none of the prior works have proposed an architecture for extracting norms from open source repositories. The rest of this paper contributes towards addressing these issues.

3 Classifications of Norms in Open Source Software Repositories

We note that several types of norms might exist in open source software development communities. We briefly discuss the distinction between norms and conventions using some examples in the context of OSSD. Also, we provide a brief overview of the norm life-cycle that can be observed in OSS projects.

Conventions. Conventions of a community are the behavioural regularities that can be observed. Coding standards of a project community is an example of a convention. The specifications of these conventions may be explicitly available from the project websites[2] or can be inferred implicitly (e.g. a wide spread convention that may not be explicitly specified in project websites).

Norms. Norms are conventions that are enforced. A community is said to have a particular norm, if a behaviour is expected of the individual members of the community and there are approvals and disapprovals for norm abidance and violation respectively. There have been several categorizations of norms proposed by researchers (cf. [14]). We believe that deontic norms - the norms describing prohibitions, obligations and permissions studied by the NorMAS community [18] is an appropriate categorization for investigating different types of norms that may be present in OSSD communities.

Prohibition norms prohibit members of a project group from performing certain actions. However, when those actions are performed, the members may be subjected to sanctions. For example, the members of an open source project may be prohibited to check-in code that does not compile, and they may be

[2] Refer to http://source.android.com/source/code-style.html for the coding guidelines for Android development.

prohibited to check-in a revised file without providing a comment describing the change that has been made. *Obligation norms* on the other hand describe activities that are expected to be performed by the members of a project community. When the members of a community fail to perform those, they may be subjected to sanctions. For example, the members may be expected to follow the coding convention that has been agreed upon. Failure to adhere to this convention may result in the code not being accepted by the repository (e.g. based on automatic checking) or a ticket may be issued by a quality assurance personnel. *Permission norms* describe the permissions provided to the members (e.g. actions they can perform). For example, an user playing the role of the project manager is permitted to create code branches.

In NorMAS, researchers have proposed a life-cycle for norms (e.g. [14]). The norm life-cycle in the context of open source development consists of four phase, convention creation, codification, monitoring and enforcement. In open source repositories, an initial phase is the convention formation (or creation) phase where the members of the community discuss what the conventions of the software project should be. Once the coventions have been agreed upon, they might be codified into written rules. There are several examples of codified conventions in several open source communities. For example, the open source Apache project[3] and the Android development community[4] provide guidelines on conventions including coding conventions. Once the convention has been codified, that forms the basis of norms through the creation of *normative expectations*. It is expected that the members of the project community adhere to these norms. Upon the codificiation of these norms, projects choose to monitor norms either through centralized or distributed mechanisms. Some projects have integrated convention checking tools such as CheckStyle[5] and StyleCop[6] in their project submission systems and any violations of norms are by default prohibited. Another option is for projects to facilitate a distributed monitioring mechanism which is primarily manual where individual contributors report any violations. The fourth stage is the enforcement stage. While using one of the convention checking tools, the enforcement is instantaneous. However, in a distributed approach to sanctioning, there could be several types of penalties. For example, a ticket could be issued for breaking a norm. There could be email exchanges between individuals discussing the importance of honouring conventions. Also, there might be invisible penalties for the violator such as the decrease of reputation and trust. Based on the discussions generated on a particular norm, there may be re-evaluations leading to the adjustment of norms (change of norms). Thus, the process enables a feedback loop to the norm formation phase. Norms can also be formed through an emergent approach. Once the project is well

[3] http://portals.apache.org/development/code-standards.html
[4] https://sites.google.com/a/android.com/opensource/submit-patches/code-style-guide
[5] http://checkstyle.sourceforge.net/
[6] http://archive.msdn.microsoft.com/sourceanalysis

underway, there could be a new convention[7] that advocates all version changes should be accompanied with a non-trivial explanation or comment on the change that was made. Thus, the emergent behaviour can be encoded as a convention (as a part of the norm formation stage).

4 A Case Study on Norm Mining

We conducted a case study to examine certain categories of conventions to study whether they are adhered in large software repositories (phase 3, the monitoring phase of the norm life-cycle). In order to conduct this study, we chose three representative open source projects based on Java which follow the Java coding conventions. The first two projects were Apache Ant[8] and Apache Struts[9] which explicitly advocate their participants to follow Java coding conventions[10]. The third project, Apache ODE[11] did not explicitly state it follows Java coding convention. We included this for comparison purposes to see if the honouring of conventions in this project are different from the ones that had explicit statements about honouring conventions. Basic details of these projects can be found in Fig. 4.

For these three projects, we identified the five categories of conventions given below (both explicit and non-explicit) and checked whether these conformed to the conventions. These five categories were chosen to broadly represent different aspects of software development (e.g. extensibility, redundancy, and smaller footprint). We used CheckStyle 5.5, a coding standard analyzer for Java to check whether conventions are honoured.

1. *Extensibility* refers to a set of criteria that tests whether the code that has been developed is amenable to easier modification in the future.
2. *Programming pitfall* refers to a set of criteria that tests whether the developed code has avoided some common programming pitfalls that are experienced by developers. These pitfalls affect the run-time behaviour of the system.
3. *Import* refers to a set of criteria on the proper use of file imports that are included in a Java file. For example, a Java file should not have import statements with * because importing all classes from a package may lead to tight coupling between packages which might lead to problems when a new version of a library introduces name clashes.
4. *Length* refers to a set of criteria where length related attributes are measured and compared with some default values. For example, when the length of a file or a method is over certain limit, it impacts readability and maintainability of code.

[7] The new convention could emerge based on discussions.
[8] http://ant.apache.org, version 1.8.4.
[9] http://struts.apache.org, version 2.3.4.
[10] Phases 1 and 2 of the norm life-cycle are complete at this stage.
[11] http://ode.apache.org, version 1.3.5.

Convention category	Conventions (As encoded in Checkstyle 5.5)	Description/Default Value (in italics)
Extensibility	AvoidInlineConditionalsCheck	In line conditionals are hard to read (e.g. the use of ternary operator in Java), which limits the ability to understand the code.
	DesignForExtensionCheck	Checks whether classes are designed for inheritance.
	SimplifyBooleanExpressionCheck*	Checks for overly complicated boolean expressions.
Programming pitfalls	HiddenFieldCheck	Checks that a local variable or a parameter does not shadow a field that is defined in the same class.
	EqualsHashCodeCheck	Checks that classes that override equals() also override hashCode().
Import	AvoidStarImportCheck	Avoid generic imports with *.
	IllegalImportCheck	Checks for imports from a set of illegal packages.
	RedundantImportCheck	Checks for imports those are redundant.
	UnusedImportsCheck	Checks for unused import statements.
Length	FileLengthCheck*	*2000 lines*
	LineLengthCheck*	*80 characters*
	MethodLengthCheck	*150 lines*
	MethodLimitCheck	*30 methods*
	ParameterNumberCheck	*7 parameters*
Redundancy	RedundantThrowsCheck	Checks for redundant exceptions declared in throws clause such as duplicates, unchecked exceptions or subclasses of another declared exception.

Fig. 1. Convention categories and description

5. *Redundancy* refers to redundant use of code. For example, there might be redundant exceptions that are caught or redundant import statements that need to be removed. Redundancy results in code bloat thus increasing both storage and transmission costs.

The table given in Fig. 1 shows convention categories in column one and convention names as given by CheckStyle 5.5 in column two and the description of the conventions considered for this case study (or default values corresponding to certain conventions in italics) in column three.

The table shown in Fig. 2 presents the results that were obtained by running the CheckStyle checker based on the standard checks template[12] on these three projects. We note again that some of these are conventions that were originally specified in the coding conventions (e.g. the length of a line in Java should be less than 80), while others have informally emerged over time as good practices, but have not been updated in the Java convention specification (e.g. avoid star imports). We have specified the codified conventions using * in column two of Fig. 1.

In terms of extensibility, all the three projects, seem to have substantial issues. In line conditionals are predominantly used which obscure the readability of the code and it was observed that many classes were not designed for extension. There were not many issues with regards to simplifying the complex boolean expression.

There were not a huge number of issues on imports (as a percentage of errors). However, what was interesting was that most of the import errors in the

[12] http://checkstyle.sourceforge.net/availablechecks.html

Row Labels	Sum of ODE errors	Sum of Ant errors	Sum of Struts errors
Extensibility			
AvoidInlineConditionalsCheck	272	755	545
DesignForExtensionCheck	4313	9124	10199
SimplifyBooleanExpressionCheck	2	9	9
Import			
AvoidStarImportCheck	71	14	157
IllegalImportCheck	0	1	0
RedundantImportCheck	17	11	8
UnusedImportsCheck	0	14	0
Length			
FileLengthCheck	6	7	9
LineLengthCheck	9853	3837	15378
MethodLengthCheck	13	18	19
MethodLimitCheck	0	1526	0
ParameterNumberCheck	10	14	28
Redundant			
RedundantThrowsCheck	959	915	1425
Programming pitfalls			
HiddenFieldCheck	359	1845	2977
EqualsHashCodeCheck	3	0	7
Grand Total	**15878**	**18090**	**30761**

Fig. 2. Comparing conventions across projects

ODE project were on the files in the testing package (implying testers were not following conventions). However, this was not observed in the other two projects.

There were substantial number of errors across all the projects on exceeding the line length (80 characters). The number of classes exceeding the method limits check (30 methods) were high in Ant project, but there were not any in the other two projects. There were more than 10 instances in all the three projects that had more than 7 methods in a class. The method limit check, method length check and the parameter number check are indicators of complexity of the system. Higher values for these indicate that there may be a need for refactoring (e.g. reducing the number of methods in the classes of the Ant project).

It is interesting to note that there are many instances in all the three projects where a global field was hidden by a local field (hiddenFieldCheck). This is an issue because this may result in erroneous run-time behaviour of the system. In both ODE and Struts projects there were a few instances where *equals* method was overridden while the *hashCode* method was not overridden. Ant did not have any *equalsHashCodeCheck* violation. All the three projects had substantial number of redundant throws classes which causes code bloating.

Overall, our observation is that all the three projects do not adhere to a number of conventions. However, there were not any substantial differences between the projects that explicitly following conventions and not-following conventions in our case study[13]. This of course needs to be evaluated in a larger context involving a large number of projects. Additionally, analyzing projects on these

[13] There could be reasons such as internalized (non-explicit) norms that could have operated in the ODE project which are similar to the explicit norms.

five types of conventions reveal some information that warrant further investigation. For example, it could be the case that the same (small) group of programmers may have introduced all the import errors or the programming pitfall errors. Such analysis can be undertaken in the future. Our objective in this work was to undertake initial investigation of convention adherence in some well-known projects.

5 Agent-Based Architecture for Norm Mining

The preliminary case study that we have conducted on convention monitoring is a first step towards the grand challenge of recognizing normative processes in the software repository setting. In this section, we present our second step, where we propose an agent-based architecture for identifying existing and emergent norms and the level of support (i.e. conformance) for these norms in OSS repositories.

Agent mining is an emerging interdisciplinary area that integrates several disciplines such as multi-agent systems, data mining and knowledge discovery and machine learning [5]. Agent-based mining architectures have been proposed by researchers for massively parallel data processing and integration [10] and also mining banking domain specific information [12]. Here, we propose an agent-based architecture for norm mining from OSS repositories. To our knowledge no other work has considered a similar approach.

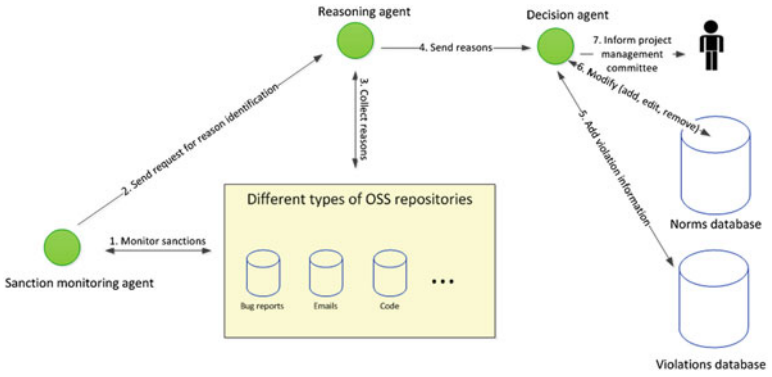


Fig. 3. Architecture for norm identification in open source software repositories

The architecture for the purpose of identifying norms in OSS repositories is given in Fig. 3. The framework makes use of software agents to store, manipulate and disseminate normative information. There are three types of agents, sanction monitoring agent, reasoning agent and decision-making agent. The rectangular box represents different types of information that may be stored in different repositories (e.g. code repository and bug repository) which are mined for appropriate information. The process associated with identifying two types of norms, the pre-specified norms and the emergent norms is described below.

- Step 1: A sanction monitoring agent monitors for sanction information (e.g. messages containing sanctions) that are added to any of the monitored repositories. For example, a message may contain information where agent A informs B that it has violated a norm. However, the veracity of the sanctions need to verified by the system (i.e. the sanction was for the right reason - an actual violation).
- Step 2: Upon sanction identification, the monitoring agent sends a request to the reasoning agent that identifies the reason for the sanction to occur.
- Step 3: The reasoning agents identifies the reason from the appropriate repositories. Note that the reason for the sanction to occur can be found in one or more of the repositories.
- Step 4: Upon identifying the reason for the sanction, the reasoning agent sends this information to the decision making agent.
- Step 5: The decision making agent checks if the reason for the violation is because one of the existing norms in the norms database has been violated. If that is the case, the norm violation is recorded in the violation database (which can be used to identify norm uptake).
- Step 6: On the other hand, if the norm is a new norm (potential norm), it is added to the norms database and the violations database. Norm change (modification and deletion) is also facilitated at this stage.
- Step 7: When a new norm is added, the project management committee is informed about the new norm (who have the ability to modify the norms if need be).

The processes associated with sanction monitoring, reasoning and decision making are elaborated further in the following sub-sections.

5.1 Sanction Monitoring

Enforcement of a pre-specified norm typically involves the delivery of appropriate sanctions. Such sanctions can be easily seen in a proprietary production process, e.g. if a developer working in a company is not following the rules, she may be at a risk of being warned or even fired. In contrast, it can be challenging to find evidence of sanctions in the OSSD setting since reliance on informal authority to accept/reject contributions tend to play a key role in the organization of OSSD. However, there are some good candidates that can be identified as sanctions.

The most visible forms of sanctioning in open source are “flaming” (public condemnation of developers that violate norms) and “shunning” (refusing to cooperate with the developers who have broken a norm) [17]. Evidences of such sanctions can be found in various archived sources (repositories) such as email lists, bug/ticket reports and forums. For example, a bug report on a module that does not deliver the specified functional requirements can be viewed as a sanction. Additionally, tickets issued for not resolving a bug completely can be considered as a sanction. Sanctions that follow violations act as triggers to infer norms. Frequency of norm violations over time may provide evidence for the uptake of a norm in a society.

We note that identifying and categorizing different types of sanctions from different types of artifacts is a challenge since the extraction of sanctions involves natural language processing. Verbose text may be used in the construction of sanction messages. For example, the messages may involve terms that are well beyond the deontic terms such as "should not", "must not", "ought not" in the case of prohibitions. One way to address this problem is to use existing ontological tools (e.g. WordNet [7]) to extract synonyms of terms used in the text to infer deontic terms and also use information retrieval tools that offer data manipulation functions such as cleaning and disambiguating the verbose text in order to extract sanctions. Suitability of tools such as OpenCalais[14] and AlchemyAPI[15] for this purpose can be investigated. We believe recognizing sanctions is indeed a huge challenge. At the same time, it presents opportunities such as the construction of normative ontologies that can be used across projects for recognizing sanctions. We envisage the Natural Language Processing (NLP) features discussed here will be built-in to the sanction monitoring agent.

5.2 Norm Identification

The process of recognizing norms (pre-specified and emergent) consists of two phases. In the first phase, the reasons for violations will be identified. In the second phase, the decision agent will decide whether a norm is pre-specified or emergent and also make other decisions accordingly.

Reason Identification. The machinery proposed by Savarimuthu et al. [15,16] can be used as a starting point to infer prohibition and obligation norms. In their work, prohibition norms are identified by extracting sequence of action (or actions) that could have caused the sanction by using a data mining approach [16]. Sanctions form the starting point for norm identification. In the case of obligation norms, missing event sequence (or sequences) that was responsible for the occurrence of a sanction, is identified [15]. The result of this process will provide a reason for the occurrence of violations (i.e. the reason for the sanction is the violation of a prohibition or an obligation norm).

While these work on norm identification can be used as a starting point for the extraction of norms in simple cases, the domain of OSSD poses non-trivial challenges. For example, correlating or linking different types of documents containing relevant information is required before a sequence of actions can be constructed. For example, an email message may contain the sanction message exchanged between developers A and B. Let us assume that A sanctions B for not adding a valid comment to the latest version of the file he or she uploaded. The problem in extracting the norm in this case is that, first, the verbose message sent from A to B should be understood as a normative statement which involves natural language processing. Second, a cross-check should be conducted to evaluate whether the normative statement is indeed true (i.e. checking whether the

[14] http://www.opencalais.com
[15] http://www.alchemyapi.com

comment entered by B is invalid by investigating the log)[16]. We envisage these tasks will be performed by the reasoning agent.

Decision Making. A decision making agent will identify whether the reason for the sanction is because of the violation of an existing norm or an emergent norm (not yet known, but could potentially be identified as a norm). If it is a violation of an existing norm, it records this information in the violation database (which can be used to study norm uptake, i.e. how common are violations of a particular norm). Otherwise, if it is a new norm and if it meets certain threshold (for reporting), it adds the new norm to the norm database and also sends this information to the project management committee who then may decide to approve the new norm.

5.3 Discussion

When the architecture is implemented and run, we envisage that there will be a number of instances of sanction monitoring agents and reasoning agents. For example, a monitoring agent can monitor certain type of norm violation and a reasoning agent can be responsible for mining the reason for the violation from different repositories.

We believe, the field of computer science now is in the cusp of transformation where the challenges posed by big data can only be solved by employing techniques from a variety of disciplines. The framework that is developed based on the above proposed architecture should be equipped with appropriate libraries for (a) information retrieval techniques (including natural language processing) in order to identify sanctions; (b) mining software repositories (e.g. cross-linking different sources); and (c) norm extraction (e.g. inferring norms from sequences of events).

We also believe that the architecture proposed in this paper can be the basis of studying interesting cross-disciplinary questions such as the ones given below.

Q1. How different are norms in large projects (e.g. measured based on total number of members or size of the project in kilo-lines of code) than the smaller projects? Are norm violation and enforcement patterns different in these projects?

Q2. What are the relationships between roles of individuals in software development and norms (e.g. contributor vs. reviewer vs. module administrator)?

Q3. Are there cultural differences within members of a project with regards to norms (inter- and intra- project comparisons) since individuals from different cultures may have different norms?

Q4. Is there a difference between norm adoption and compliance between open-source and closed-source projects?

[16] In this example only two artifacts, the email message and the log are involved. But in practice, several different types of documents may need to be traversed to find the relevant information. Techniques developed in the field of MSR (e.g. [1,13]) can be employed for cross-linking documents.

Projects	Description	Version	Convention violations
Apache ODE	Business process execution engine for processes written in WS-BPEL standard.	1.3.5	60608
Apache Ant	Ant is a Java library, mainly used to compile, assemble, test and run Java applications.	1.8.4	48082
Apache Struts	Struts is framework for creating Java web applications	2.3.4	97457

Fig. 4. Basic project information of the projects considered in the case study

These questions may interest both social researchers and computer scientists. Synergy between the two is required for addressing these questions. As computer scientists we can employ our expertise in several areas (i.e. information retrieval, MSR and normative multi-agent systems) to help answering these questions.

6 Conclusions

Open source projects are real-life, large-scale, multi-agent organizations whose data repositories are ripe for mining normative information. In this paper, we have discussed how this important application area can be leveraged by employing the techniques developed by researchers in Normative Multi-agent Systems (NorMAS), mining software repositories and also other research disciplines in computer science towards the goal of understanding normative processes that operate in human societies. Responding to the call for challenging domains to apply agent-based data mining techniques, the main motivation of the paper has been on bringing the issues forward to the agent community in general and also discussing the initial work we have undertaken. Towards that goal, first, we have presented the results on norm compliance on three open source software projects. Second, we have presented a high-level, agent-based architecture for mining norms in open source software repositories. We have also highlighted the research challenges that need to be addressed in the future.

Appendix

Basic information about the three projects are provided in Fig. 4. Columns 3 and 4 show the version numbers of the projects considered and the total number of violations observed (including the five categories of violations presented in the case study).

References

1. Bird, C., Gourley, A., Devanbu, P., Gertz, M., Swaminathan, A.: Mining email social networks. In: Proceedings of the 2006 International Workshop on Mining Software Repositories, MSR '06, pp. 137–143. ACM, New York (2006)

2. Boogerd, C., Moonen, L.: Assessing the value of coding standards: an empirical study. In: ICSM, pp. 277–286 (2008)
3. Butler, S., Wermelinger, M., Yu, Y., Sharp, H.: Mining java class naming conventions. In: ICSM, pp. 93–102 (2011)
4. Cao, L., Gorodetsky, V., Mitkas, P.A.: Agent mining: the synergy of agents and data mining. IEEE Intell. Syst. **24**(3), 64–72 (2009)
5. Cao, L., Weiss, G., Yu, P.S.: A brief introduction to agent mining. Auton. Agents Multi-Agent Syst. **25**, 419–424 (2012)
6. Criado, N., Argente, E., Botti, V.J.: Open issues for normative multi-agent systems. AI Commun. **24**(3), 233–264 (2011)
7. Fellbaum, C.: WordNet: An Electronic Lexical Database. Bradford Books, Cambridge (1998)
8. Hassan, A.E.: The road ahead for mining software repositories. In: The 24th IEEE International Conference on Software Maintenance, Frontiers of Software Maintenance, pp. 48–57, October 2008
9. Kapser, C., Godfrey, M.W.: Cloning considered harmful considered harmful. In: Proceedings of the 13th Working Conference on Reverse Engineering, pp. 19–28. IEEE Computer Society, Washington, DC (2006)
10. Kargupta, H., Hamzaoglu, I., Stafford, B.: Scalable, distributed data mining-an agent architecture. In: Proceedings Third International Conference on Knowledge Discovery and Data Mining, pp. 211–214 (1997)
11. Kooti, F., Yang, H., Cha, M., Krishna Gummadi, P., Mason, W.A.: The emergence of conventions in online social networks. In: Proceedings of the Sixth International Conference on Weblogs and Social Media, Dublin, Ireland, 4–7 June 2012
12. Li, H.X., Chosler, R.: Application of multilayered multi-agent data mining architecture to bank domain. In: International Conference on Wireless Communications, Networking and Mobile Computing, WiCom 2007, pp. 6721–6724, September 2007
13. Nagappan, N., Ball, T., Zeller, A.: Mining metrics to predict component failures. In: Proceedings of the 28th International Conference on Software Engineering, ICSE '06, pp. 452–461. ACM, New York (2006)
14. Savarimuthu, B.T.R., Cranefield, S.: Norm creation, spreading and emergence: a survey of simulation models of norms in multi-agent systems. Multiagent Grid Syst. **7**(1), 21–54 (2011)
15. Savarimuthu, B.T.R., Cranefield, S., Purvis, M.A., Purvis, M.K.: Obligation norm identification in agent societies. J. Artif. Soc. Soc. Simul. **13**(4), 3 (2010)
16. Savarimuthu, B.T.R., Cranefield, S., Purvis, M.A., Purvis, M.K.: Identifying prohibition norms in agent societies. Artif. Intell. Law **21**, 1–46 (2012)
17. Weber, S.: The Success of Open Source. Harvard University Press, Harvard (2004)
18. Wieringa, R.J., Meyer, J.-J.Ch.: Applications of deontic logic in computer science: a concise overview. In: Wieringa, R., Meyer, J.-J.Ch. (eds.) Deontic Logic in Computer Science: Normative System Specification, pp. 17–40. Wiley, New York (1994)
19. Zimmermann, T., Weisgerber, P., Diehl, S., Zeller, A.: Mining version histories to guide software changes. In: Proceedings of the 26th International Conference on Software Engineering, ICSE '04, pp. 563–572. IEEE Computer Society, Washington, DC (2004)

The Recognition of Multiple Virtual Identities Association Based on Multi-agent System

Le Li(✉), Weidong Xiao, Changhua Dai, Junyi Xu, and Bin Ge

Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha, China
lile10@126.com

Abstract. The recognition of multiple virtual identities association has aroused extensive attention, which can be widely used in author identification, forum spammer detection and other fields. We focus on the features of authors behavior on the dynamic data. This paper applies multi-agent system to the authors information mining fields and proposes a recognition model based on multi-agent system: MVIA-MAS. We cluster the author information in each time slice in parallel and then use association rule mining to find the target author groups, in which the multiple virtual identities are considered associated. Experiments show that the model has a better overall performance.

Keywords: Multiple virtual identities · Multi-agent system · Time slice

1 Introduction

With the development of network technology, forum, blog and microblog got wide-spread attention and became the main platform for the people to communicate with each other. There is a kind of authors, who often tend to publish the articles which are similar in topics in multiple sites over a period of time, in order to expand their influence (some famous people have a number of microblog accounts) or reach illegal purpose (spammer published the articles in a number of websites in a short time). This kind of behavior has obvious features in time. Traditional approaches do not consider the factor of time, deal with all the articles together, which will lead to overlook the behavioral features and affect the recognition accuracy. In this paper, we propose a recognition model based on multi-agent system: MVIA-MAS, in which multi-agent system was introduced to the authors information mining field. MVIA-MAS can be an effective simulation of the behavior features, which has a better overall performance in identifying the behavior.

Time factor is the focus of this paper, and its importance is mainly reflected in the following three aspects:

1. In order to achieve the purpose in a relatively short period of time, such posting behavior has obvious features in time. Shown in Fig. 1, authors post

L. Cao et al. (Eds.): ADMI 2013, LNAI 8316, pp. 40–50, 2014.
DOI: 10.1007/978-3-642-55192-5_4,

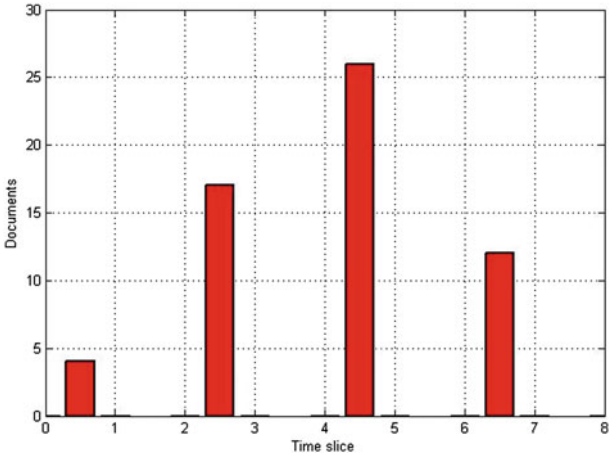


Fig. 1. The post number in different time slices

more articles within a certain time slices, but fewer of posts in the other time slices. It will be more consistent with the behavioral features of the authors by analyzing the dataset based on the time slice;

2. While a hot topic appears, it will attract a large number of users to participate in the discussion. A large number of authors will publish articles similar in topics in a short period. However, this phenomenon appears on a few time slices, but it will lead to lower identification accuracy if not take the time factor into account;

3. Some of the behavioral features will be mixed up and difficult to find if we do not consider the factor of time. It has been discussed in [1], but we take a more simple way to explain: it is assumed that there is a two-dimensional Euclidean space, the closer the distance to the author A and author B, the similarity it means. There is a similarity threshold value, if the distance is less than, it represents the two authors are similar. Figure 2(a)–(c) are the result at different time slices, (d) is the result that does not consider the factor of time. In (a) and (c), the distance between A and B is less than, the distance in (b) is greater than, (d) shows the distance between A and B is greater than. As can be seen, the relationship between A and B cannot be found if we do not consider the factor of time.

This paper proposes a recognition model based on multi-agent system: MVIA-MAS, to identify multiple virtual identities association in a dynamic data. The model takes full advantage of multi-agent system to solve the problems. Based on the analysis of time factor, we use the multi-agent system to cluster in each time slice in parallel, which cost less time; then the master agent processes the data reported by task agent as a whole, which will overcome the problem of lacking of experience of the individual agents and reach better recognition accuracy.

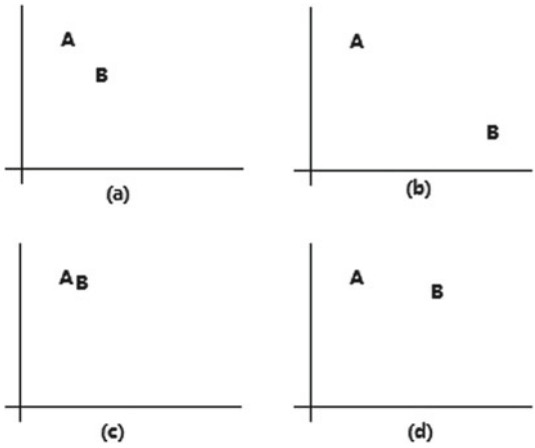


Fig. 2. Dynamic and static of similarity comparison

The rest of this paper is organized as follows. Section 2 describes some related works. Section 3 presents the research method, the model and algorithms, also discusses how to set time slice size. Section 4 includes the experiments for evaluating the proposed approach. Finally, Sect. 5 summarizes the main conclusions drawn from this study and indicates future work directions.

2 Related Works

The goal of this study is to identify whether multiple virtual identities are associated in the dynamic data, mainly focus on the similarity among the authors, which are related to the following works:

Authorship Attribution is a research process to assign a text of unknown authorship to one candidate author [2,3]. Author Verification is to study whether the current of the article belongs to a given author [5]. Author profiling aims to extract the authors the age, gender and other information from a given text [6]. Detection of stylistic inconsistencies can detect changes of the writing style [7]. Above works focus on the relationship between text and authors, however, they did not explore the relationship between authors in the dynamic data.

Author Disambiguation is to study whether a virtual identity belongs to a number of different authors [8] (such as Michael Jordan may be a basketball player or a professor major in machine learning), but our work concerns whether the multiple virtual identities belong to the same author. Therefore, it cannot meet our goal, but we can take use of its methods.

Author Topic model is a probabilistic topic model to study the relationship between author and text [9], which can be seen as a method of dimensionality reduction. Author can be seen as a distribution of topics by applying this model, therefore we can compare the similarity among different authors. Although the

paper has analyzed the trend of CiteSeer, but it only focused on the variation of the authors interests, did not concern the similarity among the authors. Daud proposed a new probabilistic model by adding time factor to Author Topic model - TAT in [10], but it did not focus on the association of multiple virtual identities. Our method is not so complicated but achieves satisfied recognition accuracy in a shorter time.

To find the association of multiple virtual identities, we need to use dimensionality reduction method to the author and text at the same time. However, the author and text belong to the heterogeneous data. Globerson in [11] proposed a novel technique for embedding heterogeneous entities such as authornames and paper keywords into two dimensional Euclidean space based on their co-occurrence counts. Sarkar extend the model to a dynamic setting in [1]. However, it mainly provides a visual analysis method, which did not explore the fields that analyzing the behavioral features. By introducing the multi-agent systems to the author information mining fields following the concept of agent mining [4], our method can achieve the goal in a different way.

3 Method, Model and Algorithms

This section briefly describes the research method including the method of dimensionality reduction and similarity comparison, and then introduces the recognition model (MVIA-MAS) and algorithm. Finally, we discuss how to set the size of time slice appropriately.

3.1 Research Method

For the dynamic network environment, we propose a method for multiple virtual identities association recognition, as shown in Fig. 3. In the first, the dataset is divided according to the size of time slices. Then we cluster the authors in each time slice based on the similarity of the author-topic distribution and extract a set of candidate groups that meet similar requirements in each time slice. Finally, we take all the candidate groups together into count and use frequent item sets mining methods to find the target groups. The virtual identities in a group are considered to have a strong relationship, which may belong to the same author.

This article uses Author Topic Model to reduce the dimensionality of the data set. Author Topic model is an important topic model based on the development of LDA in [12], it proposed a new unsupervised method to extract information from large-scale data. Author Topic model deals with the documents into two distributions, each author is represented as topic-based probability distribution and each topic is represented as word-based probability distribution.

In each time slice, a key step is to compare the similarity of the authors. This paper uses the Author Topic model as dimensionality reduction method, so similar comparison of authors is converted into a similar comparison of the author-topic distribution.

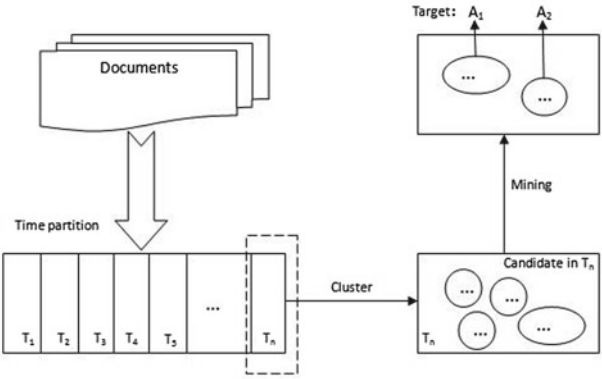


Fig. 3. Recognition process

There are several ways on the similarity measure of the two distributions, but most commonly used method is KL (Kullback Leibler) distance in [13],but KL distance is an asymmetric calculation formula, so we choose symmetrical KL distance (a modification of the KL formula) to measure the similarity between the authors:

$$KL(p,q) = \frac{1}{2}[\sum_{j=1}^{T} p_j \log_2 \frac{p_j}{q_j} + \sum_{j=1}^{T} q_j \log_2 \frac{q_j}{p_j}] \quad (1)$$

We set a similarity threshold value, if the distance between the authors is less than, which means the authors are similar and the author's ID will be put into the candidate groups.

When a hot topic appears, it will result in a number of people participating in the discussion. There will be some irrelevant authors have the similar distribution in a few time slices. Frequent itemsets mining is used in order to avoid this phenomenon.

In recent times, Adam Kirsch et al. in [14] sets the support threshold based on statistical methods, we put it into further work and still use the empirical formula proposed by Aggarwal et al. We take frequent itemsets mining as a method to overcome the problem of the lack of experience of the individual agents, which enhance the ability of the model to solve the problem.

3.2 Recognition Model (MVIA-MAS) and Algorithm

This paper proposes a recognition model based on multi-agent system: MVIA-MAS, to identify multiple virtual identities association in a dynamic data. The model is shown in Fig. 4.

In MVIA-MAS, M represents the master agent, takes charge of data pre-processing (including data cleaning, the size of the time slices setting, active task agents and so on) and choosing target group (gather results from task

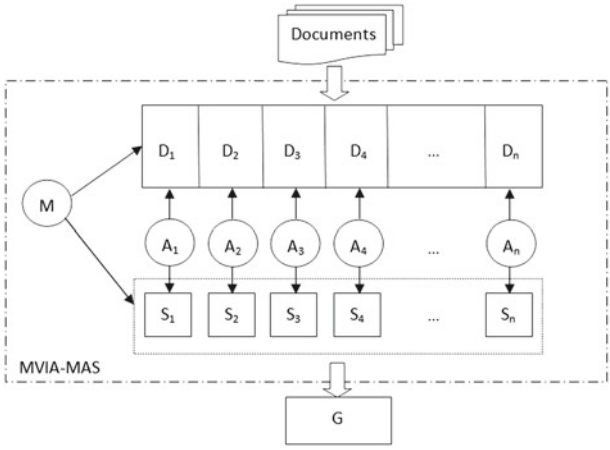


Fig. 4. MVIA-MAS model

agent and select target groups which meet the requirement). A_i represents of the task agent i, which is responsible for dimensionality reduction in each time slice (using Author Topic model) and the similarity comparison task (using KL distance). Dataset D was divided into small data set D_1, D_2, ..., D_n based on the setting of time slice. After task agent M preprocessing, task agent A_i found candidate group S = S_1, S_2, ..., S_n in each time slice. Then master agent M processed the result to find the target group G = G_1, G_2, ..., G_n, that each virtual identity in G_i has some kind of relationship.

The brief algorithm process is as follows:

```
Algorithm main control of MVIA-MAS
Input: document: D, time slice size: tss, Support_min: m, KL_distance: e,
       hyperparameters a , hyperparameters b.
function MVIA_MAS()
{
  Dn = Partition(D,tss);// partition the dataset
  Active(M);// active Master agent M
  Foreach(D1,...,Dn)
  {
    Active(Ai,Di);//active agent Ai in each time slice
    Ask(M,e,a,b); // ask for parameters
    Get(e,a,b); // get the parameters values
    Ci = AT(Di,a,b);//use Author Topic model
    Si = KL(Ci,e)//find candidate group in each time slice
    Report(Si);//report to master agent
  }
  G = M_F(S,m);// mining frequent item sets
  Return G;
}
Output: target group: G
```

3.3 Discussion on Time Slice Size

We need to pay attention to the selection of time slice size in the process of dividing the data set. If the time slice size is too small, the author's articles may span multiple time slices, so it is hard to find the similarity topics in one time slice, which will result in mixing up the behavior of the author. Nevertheless, if too coarse size of the dataset is divided, there will be a many different topics mixed in the same time slice, affecting the result of the similarity comparison, and will result in lower recognition accuracy. It can be seen, the selection of the size of the time slice makes a great impact on recognition accuracy. It is better if we select the appropriate time slice size according to the distribution of the dataset. Comparing the performance of different time slice size will be discussed in Sect. 4.

4 Experiments

In this sector, we test the accuracy and stability of the MVIA-MAS, and then compare with the static model: Author Topic model by changing the KL distance threshold. Finally, we change the size of the time slice to prove that selecting the appropriate size plays an important role for the model's recognition ability.

4.1 Data Set

We grab a data set from three famous web forums: tag as nba, qiuping and tianya, and select 14 well-known authors (see Table 1).

We choose the well-known basketball commentator's articles based on the follow-ing two points: First, these people have real-name authentication in the

Table 1. Authors in dataset

Author num	Web	Authors name
0	nba	Zuki
1	nba	Laomofenfei
2	nba	Zhangjiaweixinling
3	nba	Laoxu
4	qiuping	Zhangjiawei
5	qiuping	Yangyi
6	qiuping	Yujia
7	tianya	Zhangjiawei
8	tianya	Iforeverlvaj
9	tianya	Sixinzhiren
10	tianya	Rouruandeyanshi
11	tianya	Dalaoming
12	tianya	Lebulangxiaohuangdi
13	tianya	WCBAtuiguang

Table 2. Candidate group in each time slice

Time slice	Candidate group	Time slice	Candidate group
2010-04	{2,4}	2011-03	{2,4}
2010-05	{2,4}	2011-04	{2,4}
2010-06		2011-05	
2010-07	{2,4}	2011-06	
2010-08		2011-07	{2,7}
2010-09	{2,4} {2,7}	2011-09	{2,7}
2010-10	{2,4}	2011-10	{2,7}
2010-11	{2,4}	2011-11	
2010-12	{2,4}	2011-12	{2,7}
2011-01	{2,4}	2012-01	{2,7}
2011-02	{2,4}	2012-02	{2,7}

web. We can easily identify whether the virtual identities belong to the same author of the real world, which can contribute to evaluating the accuracy of the model; Furthermore, above authors always discuss basketball as the topics in their articles. A popular game will attract many authors to discuss, their topic distribution will be similar, which can test MVIA-MAS model the ability to identify.

We grab 651 articles based on basketball topics from April 2010 to February 2012. In this experiment, author number 2, 4 and 7 belong to the same well-known basket-ball commentator: Zhang Jiawei, others belong to different authors.

4.2 Accuracy Test

As shown in Table 2, we can get the candidate groups, which task agents report in each time slice based on the default parameters. Two groups {2, 4} and {2, 7} meet the minimum support requirements, so we conclude these virtual identities are associated. This result is consistent with the real situation, which shows the model achieves satisfied accuracy.

4.3 Stability Test

For testing the stability of the MVIA-MAS, we replace a variety of experimental parameters to conduct experiments: {topic number = 20, KL threshold value = 0.1, number of iterations = 200}, {topic number = 25, KL threshold value = 0.5, number of iterations = 500}, {topic number = 20, KL threshold value = 0.2, number of iterations = 2000}. The candidate groups reported by task agents are as shown in Fig. 5.

As the parameters changed, there are some noise data in the candidate groups (other collections does not satisfy the minimum support), but the MVIA-MAS can still ensure the target groups {2, 4} and {2, 7} by using frequent itemsets mining method. The experiment indicates the MVIA-MAS has better stability.

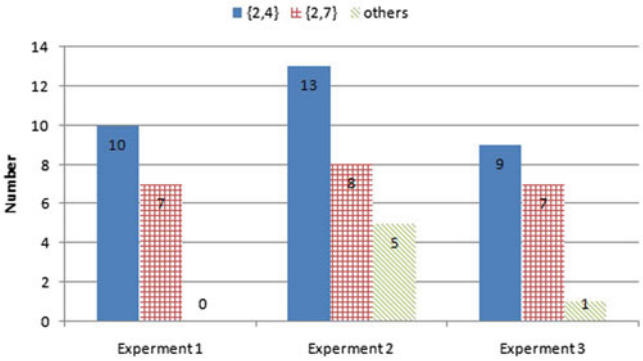


Fig. 5. Stability test

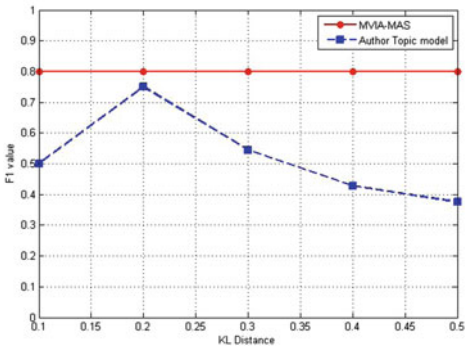


Fig. 6. MVIA-MAS and Author Topic model comparison experiment

4.4 Contrast with Author Topic Model

Under the default parameters, we change the KL distance threshold and compare the results between MVIA-MAS and Author Topic model. We use the F_1 value to evaluate the experimental results.

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{2}$$

F_1 value is a measure of a test's accuracy. It considers both the precision p and the recall r of the test to compute the score. The results are shown in Fig. 6.

As can be seen, in the case of replacement of the KL threshold value, MVIA-MAS uses time factor to analysis the dataset in the preprocessing and use association rule mining to filter the accidental events, which performs better than the AT model method, the ability to identify is better and relatively stable. At the same time, by applying the multi-agent system to each time slice in parallel, we get a faster processing speed. MVIA-MAS is especially suitable for the real-time requirements of the occasion.

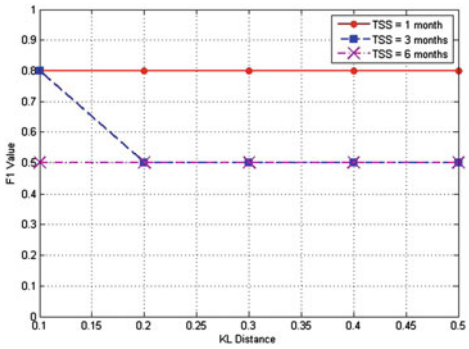


Fig. 7. MVIA-MAS model replacement of the size of the time slice experiments

4.5 Contrast Experiment by Setting Different Time Slice Size

In order to understand the importance of the time slice size, we replace the TSS (time slice size) and set for a month, three months and six months, and test the model on different the KL threshold value. The results are shown in Fig. 7.

As can be seen, TSS and KL threshold affect the overall performance. Therefore, how to adjust the model automatically and set a suitable time slice size will be focused on future research. It is interesting to see that above experiments can identify the target group {2, 4} and {2, 7} accurately, but does not identify the {4, 7}, which is mainly due to the reason of the data set. According to the distribution of the data set, the distribution of 4 and of 7 in the same time slice is less, so the system cannot confirm the author 4 and 7 belong to the same group. Incomplete data sets are often encountered in a real network environment, so for such cases it relies on people to use common sense and judgment to achieve a better level of recall rate.

5 Conclusion

For research on multiple virtual identities association in a dynamic network environment, the paper presents a recognition model based on multi-agent system: MVIA-MAS. By extracting relevant behavioral features in data sets, the data is divided into different time slices. The task agent compares the similarity of authors in each time slice, and then master agent uses the association rule mining method to filter out the impact of accidental events. By choosing the appropriate parameters, MVIA-MAS achieves better accuracy, stability and time-efficiency compared to the Author Topic model. The direction of future research is to use a larger and more complete data set to test the model, add writing style analysis and sentiment analysis to the model parameters to enhance the overall performance of the model.

Acknowledgements. We warmly thank Wentang Tan for his guidance. This work was funded under National Science and Technology Support Program (NO.2012BAH08B01).

References

1. Sarkar, P., Siddiqi, S.M., Gordon, G.J.: Approximate Kalman filters for embedding author-word co-occurrence data over time. In: Airoldi, E.M., Blei, D.M., Fienberg, S.E., Goldenberg, A., Xing, E.P., Zheng, A.X. (eds.) ICML 2006. LNCS, vol. 4503, pp. 126–139. Springer, Heidelberg (2007)
2. Stamatatos, E.: A survey of modern authorship attribution methods. J. Am. Soc. Inf. Sci. Technol. **60**(3), 538–556 (2009)
3. Stamatatos, E.: Author identification using imbalanced and limited training texts. In: 18th International Workshop on Database and Expert Systems Applications, 2007, DEXA'07, pp. 237–241 (2007)
4. Cao, L., Gorodetsky, V., Mitkas, P.: Agent mining: the synergy of agents and data mining. IEEE Intell. Syst. **24**(3), 64–72 (2009)
5. Koppel, M., Schler, J.: Authorship verification as a one-class classification problem. In: Proceedings of the 21st International Conference on Machine Learning, p. 62. ACM Press, Banff (2004)
6. Koppel, M., Argamon, S., Shimoni, A.R.: Automatically categorizing written texts by author gender. Lit. Linguist. Comput. **17**(4), 401–412 (2002)
7. Graham, N., Hirst, G., Marthi, B.: Segmenting documents by stylistic character. Nat. Lang. Eng. **11**(4), 397–416 (2005)
8. Mann, G.S., Yarowsky, D.: Unsupervised personal name disambiguation. In: Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL 2003, vol. 4, pp. 33–40 (2003)
9. Steyvers, M., Smyth, P., Rosen-Zvi, M., Griffiths, T.: Probabilistic author-topic models for information discovery. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 306–315 (2005)
10. Daud, A., Li, J., Zhou, L., Muhammad, F.: Exploiting temporal authors interests via temporal-author-topic modeling. In: Huang, R., Yang, Q., Pei, J., Gama, J., Meng, X., Li, X. (eds.) ADMA 2009. LNCS, vol. 5678, pp. 435–443. Springer, Heidelberg (2009)
11. Globerson, A., Chechik, G., Pereira, F., Tishby, N.: Euclidean embedding of co-occurrence data. Adv. Neural Inf. Process. Syst. **17**, 497–504 (2004)
12. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
13. Lin, J.: Divergence measures based on the Shannon entropy. IEEE Trans. Inf. Theory **37**(1), 145–151 (1991)
14. Kirsch, A., Mitzenmacher, M., Pietracaprina, A., Pucci, G., Upfal, E., Vandin, F.: An efficient rigorous approach for identifying statistically significant frequent itemsets. J. ACM (JACM) **59**(3), 12 (2012)

Data Mining

Redundant Feature Selection for Telemetry Data

Phillip Taylor[1](✉), Nathan Griffths[1], Abhir Bhalerao[1], Thomas Popham[2], Xu Zhou[2], and Alain Dunoyer[2]

[1] Department of Computer Science, University of Warwick, Coventry CV4 7AL, UK
phil@dcs.warwick.ac.uk
[2] Jaguar Land Rover Research, Coventry, UK

Abstract. Feature sets in many domains often contain many irrelevant and redundant features, both of which have a negative effect on the performance and complexity of agents that use the data [9]. Supervised feature selection aims to overcome this problem by selecting features that are highly related to the class labels, yet unrelated to each other. One proposed technique to select good features with few inter-dependencies is minimal Redundancy Maximal Relevance (mRMR) [12], but this can be impractical with large feature sets. In many situations, features are extracted from signal data such as vehicle telemetry, medical sensors, or financial time-series, and it is possible for feature redundancies to exist both between features extracted from the same signal (intra-signal), and between features extracted from different signals (inter-signal). We propose a two stage selection process to take advantage of these different types of redundancy, considering intra-signal and inter-signal redundancies separately. We illustrate the process on vehicle telemetry signal data collected in a driver distraction monitoring project. We evaluate it using several machine learning algorithms: Random Forest; Naïve Bayes; and C4.5 Decision Tree. Our results show that this two stage process significantly reduces the computation required because of inter-dependency calculations, while having minimal detrimental effect on the performance of the feature sets produced.

1 Introduction

Feature sets in a range of domains often contain numerous irrelevant and redundant features, both of which have a negative effect on the performance and complexity of agents that use the data [2,9]. Supervised feature selection aims to overcome this problem by selecting features that are highly correlated to the class labels, yet uncorrelated to each other. However, finding redundancy between features is computationally expensive for large feature sets.

In many cases the features themselves are extracted from multiple signal data such as vehicle telemetry [17], medical sensors [15], weather forecasting [11], and financial time-series analysis [14]. When features are extracted from signal data, it is possible for feature redundancy to be either between features extracted from the same signal (intra-signal), or between features extracted from different signals (inter-signal). In this paper we propose to consider these types of

L. Cao et al. (Eds.): ADMI 2013, LNAI 8316, pp. 53–65, 2014.
DOI: 10.1007/978-3-642-55192-5_5,

redundancies separately, with the aim of both speeding up the feature selection process and minimizing redundancy in the feature set chosen. We illustrate this two stage process on vehicle telemetry data collected in a driver distraction monitoring project, although the concept generalizes to other domains where signal data is used as the basis for intelligent agents to build prediction models.

The remainder of this paper is structured as follows. In Sect. 2, we examine current approaches to feature selection and introduce driver monitoring and the issues associated with vehicle telemetry data. In Sect. 3 we propose a two stage feature selection process aimed at minimizing feature and signal level redundancies, which reduces the computational cost compared to existing methods. Finally, in Sect. 4, we describe our evaluation strategy and present results for the proposed method alongside results for existing techniques.

2 Related Work

2.1 Redundant Feature Selection

In general, there are two approaches to performing feature selection: wrapper and filter [9]. The wrapper approach generates feature subsets and evaluates their performance using a classifier to measure their discriminating power. Filter approaches treat feature selection as a preprocessing step on the data without classification, thus being agnostic to the classification algorithm that may be used.

The wrapper approach requires a search through the feature space in order to find a feature subset with good performance [9]. The merit of a feature subset is determined by the performance of the learning algorithm using features from that set. Methods for searching through the space of feature combinations include complete, forward, and backward searches. A complete search generates all possible feature subsets in order to find the optimal one, but can be infeasible with more than a few features. Forward selection starts with no selected features and the feature which improves performance the most is then added to the selected features. This is repeated until some stopping criterion is met, such as when performance cannot be improved further, or the required number of features have been selected. Backwards selection begins by selecting all features and then removes those which degrade performance the least. This is repeated until a stopping criterion is met.

With large datasets however, the wrapper approach still requires significant computation in building several classification models, as features are added or removed from the set. Therefore, filter methods, which require considerably less computation, are often preferred. The filter methods can be split into ranking algorithms [8], which consider features independently, and those which consider inter-dependencies and redundancy within the feature sets [1,12,13].

This means that the process can be slow on datasets with large numbers of features and samples, as is the case in many domains with signal data. Kira and Rendell proposed the Relief algorithm [8], which was later extended by Kononenko to deal with noisy, incomplete and multi-class datasets [10].

The Relief algorithm repeatedly compares random samples from the dataset with samples that are most similar (one of the same label and one of a different label), to obtain a ranking for feature weights. Although less computationally expensive than wrapper approaches, this still requires searching through the dataset for Near-hit and Near-miss examples. This means that the process can be slow on datasets with large numbers of features and samples, as is the case in many domains with signal data.

Other ranking methods are based on information measures, such as Mutual Information (MI),

$$MI(f_1, f_2) = \sum_{v_1 \in vals(f_1)} \sum_{v_2 \in vals(f_2)} p(v_1, v_2) \log_2 \frac{p(v_1, v_2)}{p(v_1)p(v_2)}, \tag{1}$$

where f_1 and f_2 are discrete features, $p(v_i)$ is the probability of seeing value v_i in feature f_i, and $p(v_i, v_j)$ is the probability of seeing values v_i and v_j in the same sample.

Because MI is summed over all the values of a feature, it tends to score features with many values highly. Therefore, normalized variants of MI are often used to remove this bias. A common method of normalization is to divide by the entropy of the features, $H(f)$,

$$H(f) = -\sum_{v \in vals(f)} p(v) \log_2 p(v). \tag{2}$$

Two variants of MI which normalize by entropy are the Gain Ratio (GR) and Symmetric Uncertainty (SU),

$$GR(f_1, f_2) = \frac{MI(f_1, f_2)}{H(f_1)}, \tag{3}$$

$$SU(f_1, f_2) = 2\frac{MI(f_1, f_2)}{H(f_1) + H(f_2)}. \tag{4}$$

However, none of these ranking methods consider the issue of feature redundancy. Redundancy is known to cause problems for efficiency, complexity and performance of models or agents that use the data [1,9]. Therefore, it is important to consider interdependencies between features and remove those which are redundant. Herman *et al.* [6] provide an in-depth review of redundant feature selection using MI. In their paper, a general framework is proposed for the several techniques in defining redundancy and combining it with relevancy. One such method is minimal redundancy maximum relevance (mRMR) [1,12], referred to as MI difference in [6]. In mRMR, the relevancy, $Rel(F, C)$, of a feature set, F, is given by the mean MI of the member features and the class labels, C, namely,

$$Rel(F, C) = \frac{1}{|F|} \sum_{f_i \in F} MI(f_i, C). \tag{5}$$

The redundancy, $Red(F)$, of F is given by the mean of all inter-dependencies as measured by MI:

$$Red(F) = \frac{1}{|F|^2} \sum_{f_i, f_j \in F} MI(f_i, f_j). \tag{6}$$

The difference between relevance and redundancy can be used for selecting features,

$$\max_{S \subset F} \; Rel(S, C) - Red(S). \tag{7}$$

However, this is computationally expensive as the redundancy for all possible feature subsets must be computed. In practice, therefore, Ding and Peng suggest performing forward selection for mRMR, iteratively selecting the features which satisfy:

$$\max_{f \in F \setminus S} \; Rel(\{f\}, C) - Red(S \cup \{f\}). \tag{8}$$

Here, MI is used as a measure of both relevance and redundancy, and this may again bias towards features with many values. It is therefore possible to use normalized variants of MI, such as GR or SU instead. We will refer to these versions as MImRMR, GRmRMR and SUmRMR depending on whether MI, GR or SU are used as relevance measures respectively.

Others methods in the general framework combine relevancy and redundancy in a ratio rather than a difference [6], as also mentioned in [1]. Others still redefine redundancy as the maximum MI value between two features in the selected set, or define it with respect to the class label values as follows:

$$Red(F, C) = \frac{1}{|F|^2} \sum_{f_i, f_j \in F} MI(f_i, f_j) - MI(f_i, f_j | C). \tag{9}$$

One further related method, not mentioned in [6], is the correlation-based feature selector (CFS) [5]. In the CFS, a ratio between redundancy and relevancy is used, and is normalized by a factor as follows:

$$\max_{S \subset F} \frac{|S| Rel(S, C)}{\sqrt{|S| + |S|(|S| - 1) Red(S)}}. \tag{10}$$

All of these methods are applicable to this work, but in this paper we use mRMR for our experiments because of its relative simplicity and widespread use in the literature.

2.2 Driver Monitoring

Driving a vehicle is a safety critical task and demands a high level of attention from the driver. Despite this, modern vehicles have many devices with functions that are not directly related to driving. These devices, such as radio, mobile phones and even internet devices, divert cognitive and physical attention from the primary task of driving safely. In addition to these distractions, the driver may also be distracted for other reasons, such as dealing with an incident on

the road or holding a conversation in the car. One possible solution to this distraction problem is for an intelligent agent to limit the functionality of in-car devices if the driver appears to be overloaded. This can take the form, for example, of withholding an incoming phone call or holding back a non-urgent piece of information about traffic or the vehicle.

It is possible for an autonomous agent to infer the level of driver distraction from observations of the vehicle and the driver. Based on these inferences, the agent can determine whether or not to present the driver with new information that might unnecessarily add to their workload. Traditionally, such agents have monitored physiological signals such as heart rate or skin conductance [3,17]. However, such approaches are not practical for everyday use, as drivers cannot be expected to attach electrodes to themselves before driving. Other systems have used image processing for computing the driver's head position or eye parameters, but these are expensive, and unreliable in poor light conditions.

Therefore, our aim is to use non-intrusive, inexpensive and robust signals, which are already present in vehicles and accessible by the Controller Area Network (CAN) [4]. The CAN is a central bus to which all devices in the vehicle connect and communicate by a broadcast protocol. This allows sensors and actuators to be easily added to the vehicle, enabling agents to receive and process telemetric data from all modules of the car. This bus and protocol also enables the recording of these signals, allowing us to perform offline data mining.

Agents processing data from the CAN-bus on modern vehicles have access to over 1000 signals, such as vehicle and engine speeds, steering wheel angle, and gear position. From this large set of signals, many potential features can be extracted using sliding temporal windows. These include statistical measures such as the mean, minimum or maximum, as well as derivatives, integrals and spectral measures. In [17], Wollmer *et al.* extract a total of 55 statistical features over temporal windows of 3 seconds from 18 signals including steering wheel angle, throttle position and speed, and driver head position. This provides a total of 990 features for assessing online driver distraction. They used the correlation based feature selector as proposed in [5] with SU as a measure of correlation.

3 Proposed Approach

As previously noted, redundancy in signal data can be considered as either intra-signal, between features extracted from within one signal, or inter-signal, between features extracted from different signals. For instance, in CAN-bus Data there is unsurprisingly a large inter-signal redundancy between the features of *Engine Speed* and *Vehicle Speed* signals. This is confirmed by the correlation between the raw values of the signals, of 0.94 in our data. There may also be a high intra-signal feature redundancy, as with the minimum, mean and maximum features. This is particularly the case when the temporal window is small and the signal is slowly varying.

Therefore, we propose a two step procedure to remove these intra-signal and inter-signal redundancies, by considering them separately. In the first stage, feature selection is performed solely with extracted features from individual signals,

aiming to remove intra-signal redundancies. In the second stage, selection is performed on these selected features as a whole, removing inter-signal redundancies. This then produces a final feature set with an expected minimal redundancy for an agent to use in learning a prediction model.

This two stage process has benefits with regards to computation. For instance, the forward selection method of mRMR requires a great deal of computation with large feature sets. Moreover, large feature sets, such as those extracted from CAN-bus Data, often do not fit into memory in their entirety, meaning that features have to be loaded from disk in sections to be processed. This not only lengthens the feature selection process, but also impacts on the complexity of the implementation. With our two stage process however, smaller numbers of features are considered at a time, meaning that at each stage, these problems do not occur.

In using this process, we expect there to be fewer redundancies in the final feature sets because redundancies are removed at both stages. However, returning fewer features in this first stage may reduce the relevance of the selected features to be used in learning. This will particularly be the case when many of the best performing features are from the same signal, but this is assumed to be an extreme and uncommon case.

The two stage selection process can be used in conjunction with any feature ranking method, regardless of whether redundancy is considered. For instance, if MI is used without mRMR and one feature is selected per signal, this will be equivalent to MImRMR also with one feature per signal in the first stage. The ranking method used does not have to consider redundancy for intra-signal redundancies to be removed. To remove inter-signal redundancies however, the second stage must remove redundancies itself. Also, wrapper methods can be used in place of the mRMR framework.

4 Evaluation

4.1 CAN-bus Data

CAN-bus Data was collected during a study where participants drove under both normal and distracted conditions. To impose distraction on the driver, a series of tasks, as listed in Table 1, were performed at different intervals. For the duration of a task, the data is labelled as *Distracted*, otherwise it is labelled as *Normal*. In this study there are 8 participants, each driving for approximately 1.5 h during which each of the 6 tasks are performed twice. Data was recorded from the 10 signals shown in Table 2 with a sample rate of 10Hz.

In addition to the tasks listed in Table 1, participants performed two driving manoeuvres, namely abrupt acceleration and a bay park. The data from these are, however, considered to be unrelated to distraction and therefore can be viewed as noise and were removed from the dataset. This removal was done after feature extraction to avoid temporal continuity issues.

The features listed in Table 2 are extracted temporally from each signal over sliding windows of sizes 5, 10, 25 and 50 samples (0.5, 1, 2.5, 5 s respectively),

Table 1. Secondary tasks the driver was asked to perform. If there is a secondary task being performed, the data is labelled as *Distracted* for the duration, otherwise it is labelled as *Normal*. Tasks were performed in the same order for all experiments, with intervals of between 30 and 300 s between tasks.

Secondary task	Description
Select radio station	Selection of a specified radio station from presets
Mute radio volume	The radio is muted or turned off
Number recall	Recite a 9 digit number provided before the drive
Satellite navigation programming	A specified destination is programmed into the incar Sat-Nav
Counting backwards	Driver counts backwards from 200 in steps of 7 (i.e., 200, 193, 186...)
Adjust cabin temperature	Cabin temperature increased by 2 °C

Table 2. The 10 signals and 40 features used in the data mining process. Signals are recorded from the CAN-bus at 10 Hz. Features are extracted temporally from each signal over sliding windows of 5, 10, 25 and 50 samples (i.e. 0.5, 1, 2.5, 5 s). This gives 120 features per signal and a feature set of size 1200.

Signal	Features
Steering wheel angle	Convexity
Steering wheel angle speed	First, Second and Third derivatives
Pedal position	First 5 and Max 5 DFT coefficient magnitudes
Throttle position	Max 5 DFT coefficient frequencies
Absolute throttle position	Entropy
Brake pressure	Fluctuation
Vehicle speed	Gradient: positive, negative or zero
Engine speed	Integral and absolute integral
Yaw rate	Min, Max, Mean and Standard deviation
Gear selected	Number of zero values and Zero crossings

providing 120 features per signal. This gives a feature set of size $10 \times 30 \times 4 = 1200$ in total. After feature extraction, the data is sub-sampled temporally by a factor of 10, providing a total of 6732 samples with the label *Distracted*, and 26284 samples with the label *Normal* over 8 datasets. This sub-sampling is done in order to speed up experiments and allow more features to be selected per signal to go forward to the second stage, which would otherwise be limited due to computational limits.

4.2 Experimental Set-up

We select features with the MImRMR, GRmRMR and SUmRMR feature selectors. During the first selection stage, 1, 2, 3, 4 and between 5 and 50 features (in intervals of 5) are selected from each signal. The second stage outputs a ranking of features from which between 1 and 30 are selected for evaluation. In the cases where 1 and 2 features are selected from each signal, there will be a

maximum of 10 and 20 features output from the two stage process respectively. In the evaluations where more than these numbers of features are required, we use all of the available features. The selection algorithm used in the first stage is always the same as the one used in the second stage.

The feature sets are then evaluated using the Naïve Bayes, C4.5 Decision Tree, and Random Forest learning algorithms available in WEKA [16]. In each experiment, after feature selection the data is sub-sampled in a stratified random way by a factor of 10. This gives 673 samples with the label *Distracted*, and 2628 samples with the label *Normal* for each experiment. This random sub-sampling reduces autocorrelation in the data, which would mean evaluations give overly optimistic performance and cause models to overfit. It also introduces some randomness into the evaluations, meaning that they can be run multiple times to gain a more accurate result. All evaluations are run 10 times, each giving an area under the receiver operator characteristic (ROC) curve (AUC).

In each evaluation, a cross folds approach is used, where training is performed on seven datasets and testing on the final one for each fold. These are averaged to give a performance metric for each feature selection procedure. A higher AUC value with smaller number of features indicates a good feature selector.

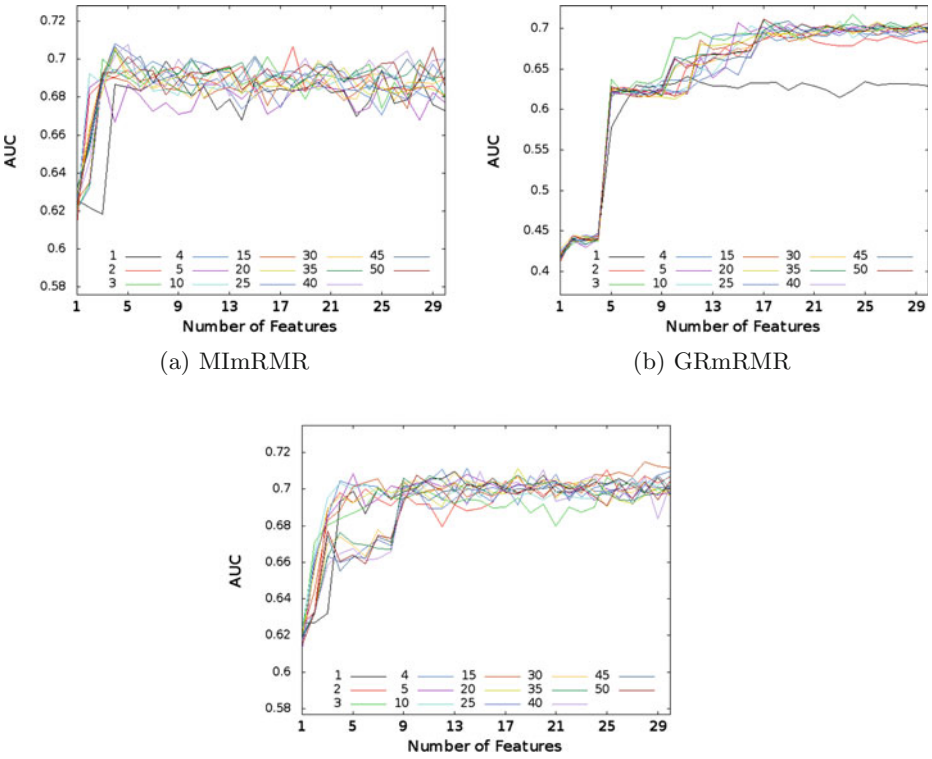
4.3 Results

First we present performance of our two stage process with the selection of varying numbers of features from each signal in the first stage. Second, we show computation times for our two stage selection process and compare these with selecting all of the features at once. Finally, we compare the performance of our two stage selection process against selecting features without our process.

Figure 1 shows AUC values for different feature set sizes, selected using the two stage selection process. Each line represents a different number of features per signal selected in the first stage. From these results we can make three observations about the two stage selection process. First, learning with more features provides better performance. Second, with more than five features, there is little further performance gain. Finally, in most cases, performance is unaffected by the number of features selected per signal. However, in some cases where small numbers of features are used for learning, worse performance is observed when small numbers of features are selected per signal. Also, when using GRmRMR we can see that selecting 1 feature per signal is not sufficient for maximal performance. This is possibly because GRmRMR is not selecting the best feature from each signal first, and therefore produces a sub-optimal feature set.

The same results are also seen with other learning algorithms. In Fig. 2, the AUC values are shown for the C4.5 Decision Tree and Random Forest classification algorithms built with features selected by MImRMR using our two stage process. One small difference here is that the Decision Tree performs much worse than Naïve Bayes and Random Forest with very few features, but still achieves comparable performance when 10 features are used for learning.

Second, we show that, while selecting features that have the same performance, computation times are substantially reduced. In Table 3, computation



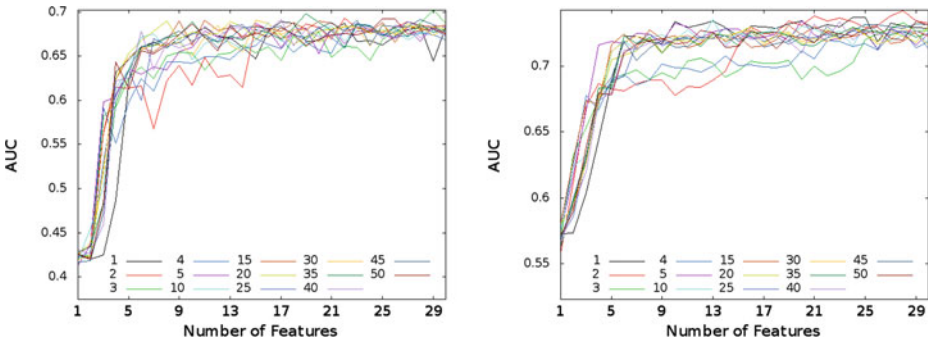
(a) MImRMR (b) GRmRMR

(c) SUmRMR

Fig. 1. AUC values for (a) MImRMR, (b) GRmRMR and (c) SUmRMR using a Naïve Bayes classifier for different numbers of features being selected after the two stage process. Each line represents a different number of features selected per signal in the first stage. In most cases, performance is unaffected by the number of features selected per signal. However, in some cases where small numbers of features are selected, performance can be worse.

times for selecting 30 features using our two stage process are presented. Denoted in parentheses are the speed-ups when compared to selecting from the total 1200 features without using the two stage process. Selecting more than 15 features per signal does not provide any significant speed-up. Selecting 1 or 2 features per signal in the first stage provides speed-ups of over 30x, however, this is not advised due to the performance results presented above. Selecting between 3 and 5 features provides equivalent performance to other methods investigated, and gives a speed-up, of over 10x for GRmRMR and SUmRMR. The smaller speed-ups for MImRMR are likely due to a simpler selection process, as multiple entropies do not have to be computed.

It is worth noting that these timings do not include reading the data from disk. Obviously, with larger feature sets, reading the data requires more time. However, in the two stage process IO times would continue to increase linearly



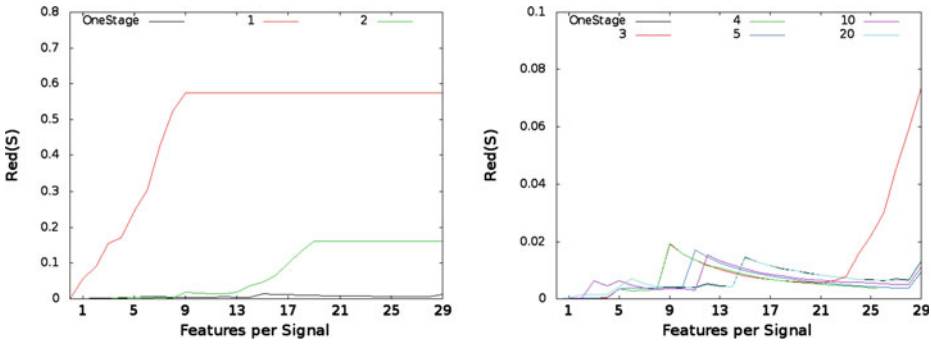
(a) Decision Tree (b) Random Forest

Fig. 2. AUC values for MImRMR using the (a) Decision Tree and (b) Random Forest classifiers for different numbers of features being selected after the two stage process. Each line represents a different number of features selected per signal in the first stage. We see the same results as with the Naïve Bayes learner, that there is little difference in performance in most cases.

Table 3. Computation times in seconds for selecting 30 features from 1200 using the two stage process with varying features per signal. Denoted in parentheses are the speed-ups, which are computed with respect to selecting 30 features from 1200 as a whole. We see that selecting more than 15 features per signal does not provide significant speed-up. However, as the performance results show, only a small number of features is required for equivalent performance. In these cases we see a much larger speed-up, of around 10x.

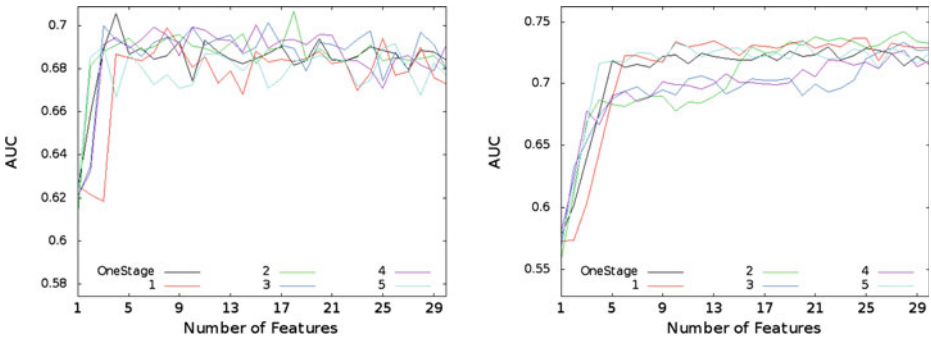
Features/Signal	MImRMR	GRmRMR	SUmRMR
1	2.6 (16.4x)	5.0 (73.2x)	5.1 (71.7x)
2	4.2 (9.9x)	11.3 (32.6x)	11.6 (31.6x)
3	6.1 (6.9x)	19.1 (19.3x)	19.4 (18.9x)
4	8.4 (5.0x)	25.6 (14.3x)	25.8 (14.2x)
5	9.9 (4.2x)	32.3 (11.4x)	33.0 (11.1x)
10	18.7 (2.2x)	67.6 (5.4x)	66.7 (5.5x)
15	26.5 (1.6x)	102.4 (3.6x)	102.3 (3.6x)
20	35.4 (1.2x)	144.2 (2.5x)	143.8 (2.5x)
25	43.3 (1.0x)	186.3 (2.0x)	186.0 (2.0x)
30	52.7 (0.8x)	236.1 (1.6x)	232.1 (1.6x)

with the number of features. This is because we consider each signal separately and only select a small number of features from each, meaning the features brought forward to the second stage are likely to fit in memory. Without this process, the increase in IO times may be more than linear because redundancies between all features are required. It is unlikely that all features will fit in memory, meaning that the data would have to be loaded in chunks, with each chunk being loaded multiple times. Therefore, the speed-ups we present here are likely to be conservative for very large datasets.



(a) features per signal (b) features per signal

Fig. 3. Redundancies of feature sets produced when selected using MImRMR, with and without two stage selection. With 1 or 2 features per signal, the redundancy is high in the returned feature set. As more features per signal are used, the redundancies become indistinguishable from when selecting features in one stage (OneStage).



(a) NaïveBayes (b) Random Forest

Fig. 4. AUC values for MImRMR using the (a) Naïve Bayes and (b) Random Forest classifiers for different numbers of features being selected. These results compare our two stage process (selecting 1, 2, 3 and 5 features per signal), with selecting features in one stage (OneStage). We again see very similar performance for the two methods.

Finally, we compare the performance of mRMR with and without our two stage feature selection process. The redundancies calculated using Eq. 6 for selecting between 1 and 30 features with MImRMR are shown in Fig. 3. When selecting only 1 or 2 features per signal, the redundancy increases to 0.57 and 0.16 respectively. This is likely because, as previously mentioned, several of the signals have a high correlation, meaning that features extracted from them will also be similar. This means that in the second stage of selection, there are very few non-redundant features to select from, and a feature set with high redundancy is produced. As more features are selected per signal, the redundancy begins to resemble that of when two stage selection is not used, until 20 features are selected per signal, when the redundancies are indistinguishable.

In Fig. 4, the AUC values are again very similar, even when selecting a small number of features from each signal. Even with only having 1 feature per signal selected, the AUC performance is very similar. This similar performance suggests that the two stage selection process is itself removing intra-signal redundancies in the feature set when 1 feature is selected in the first stage. This is because with 1 feature selected, MImRMR, GRmRMR and SUmRMR are equivalent to MI, GR and SU respectively; which do not themselves consider redundancy. Finally, when the two stage process is used with 5 features per signal, there is no discernible difference in AUC than selecting features without it. It follows then that, the equivalent performance and the speed-ups with this number of features, make it beneficial to use the two stage selection process.

5 Conclusions

In this paper we have investigated a two stage selection process to speed up feature selection with signal data. The process is demonstrated with mRMR but is also applicable to other feature selection techniques, including wrapper methods. We evaluated this process on vehicle telemetry data for driver monitoring, and have shown that the process provides a computational speed-up of over 10x while producing the same results. It follows then, that it is worthwhile to consider features extracted from each signal before considering them as a whole. Furthermore, the first stage of our process can easily be parallelized, selecting features from each signal in a different thread, which would provide further speed-ups.

In future work, we intend to inspect the feature rankings after selection. This will provide insight into the features that are selected, rather than merely their performance and redundancy. Also, it will highlight any instabilities in the feature sets produced by the two stage process, which could harm performance [7].

In this paper, we have assumed that each signal has an equal probability of producing a good feature. This may not be the case, as some signals may have many high performing features whereas others may have none. Second, this assumption means that selecting 5 features from 1000 signals produces a total of 5000 features for selection in the second stage. Therefore, selection of *signals* may be necessary before any feature selection in order to gain a further speed-up.

Finally, although it is unlikely given the size of the data we have used, it is possible that the feature selection methods are overfitting to the data [9]. In future we will evaluate the feature selector algorithms on a hold-out set, not included during the feature selection process, to avoid this.

References

1. Ding, C., Peng, H.: Minimum redundancy feature selection from microarray gene expression data. J. Bioinform. Comput. Biol. **3**(2), 185–205 (2005)
2. Cao, L., Gorodetsky, V., Mitkas, P.: Agent mining: the synergy of agents and data mining. IEEE Intell. Syst. **24**(3), 64–72 (2009)

3. Dong, Y., Hu, Z., Uchimura, K., Murayama, N.: Driver inattention monitoring system for intelligent vehicles: a review. IEEE Trans. Intell. Transp. Syst. **12**(2), 596–614 (2011)
4. Farsi, M., Ratcliff, K., Barbosa, M.: An overview of controller area network. Comput. Control Eng. J. **10**(3), 113–120 (1999)
5. Hall, M.: Correlation-based feature selection for machine learning. Ph.D. thesis, The University of Waikato (1999)
6. Herman, G., Zhang, B., Wang, Y., Ye, G., Chen, F.: Mutual information-based method for selecting informative feature sets. Pattern Recogn. **46**, 3315–3327 (2013)
7. Iswandy, K., Koenig, A.: Towards effective unbiased automated feature selection. In: Sixth International Conference on Hybrid Intelligent Systems, pp. 29–29. IEEE (2006)
8. Kira, K., Rendell, L.: The feature selection problem: traditional methods and a new algorithm. In: Proceedings of the Tenth National Conference on Artificial Intelligence, pp. 129–134. AAAI Press (1992)
9. Kohavi, R., John, G.: Wrappers for feature subset selection. Artif. Intell. **97**(1), 273–324 (1997)
10. Kononenko, I.: Estimating attributes: analysis and extensions of relief. In: Bergadano, F., De Raedt, L. (eds.) Machine Learning: ECML-94. LNCS, vol. 784, pp. 171–182. Springer, Heidelberg (1994)
11. Paras, S., Kumar, A., Chandra, M.: A feature based neural network model for weather forecasting. Eng. Technol. World Acad. Sci. **34**, 66–73 (2007)
12. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans. Pattern Anal. Mach. Intell. **27**(8), 1226–1238 (2005)
13. Saleh, S. El Sonbaty, Y.: A feature selection algorithm with redundancy reduction for text classification. In: Proceedings of the 22nd International Symposium on Computer and Information Sciences, pp. 1–6. IEEE (2007)
14. Tsay, R.: Analysis of Financial Time Series, vol. 543. Wiley-Interscience, New York (2005)
15. Wegener, A.: Compression of medical sensor data [exploratory DSP]. IEEE Sig. Process. Mag. **27**(4), 125–130 (2010)
16. Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann Series in Data Management Systems, 3rd edn. Morgan Kaufmann, San Francisco (2005)
17. Wollmer, M., Blaschke, C., Schindl, T., Schuller, B., Farber, B., Mayer, S., Trefflich, B.: Online driver distraction detection using long short-term memory. IEEE Trans. Intell. Transp. Syst. **12**(2), 574–582 (2011)

Mining Emerging Patterns of PIU from Computer-Mediated Interaction Events

Yaxin Yu[1(✉)], Ke Yan[2], Xinhua Zhu[3], Guoren Wang[1], Dan Luo[3], and Suresh Sood[3]

[1] College of Information Science and Engineering, Northeastern University, Shenyang, China
{Yuyx,Wanggr}@mail.neu.edu.cn
[2] College of Software, Northeastern University, Shenyang, China
Yanke1992.yk@gmail.com
[3] QCIS, University of Technology, Sydney, Australia
{Xinhua.Zhu,Dan.Luo,Suresh.Sood}@uts.edu.au

Abstract. It has been almost 20 years since Internet services became an integral part of our lives. Especially recent popularization of SNS (Social Network Services) such as Facebook, more and more people are attracted to Internet. Internet provides many benefits to people, but yields a consequent disturbing phenomenon of obsession with Internet, which is called PIU (Pathological Internet Use) or IAD (Internet Addiction Disorder) in academia. PIU or IAD has negative effects on people's health of mind and body, therefore, it is necessary to detect PIU. Among tools of surfing Internet, since computer is the most widely interactive media, it is significant to mine PIU emerging patterns from human-computer interaction events. As a result, an emerging pattern mining method based on interactive event generators, called PIU-Miner, is proposed in this paper. Experimental results show that PIU-Miner is an efficient and effective approach to discovering PIU.

Keywords: Emerging pattern · PIU · Computer-mediated interaction · Complex event · Generator

1 Introduction

Since Internet has widely spread over the world, using computer to surfing Internet has been a basic part of our daily life. Especially, with the popularity of SNS in recent years, Internet users exploded in exponential scale. Unfortunately, some heavy users suffer from extreme dependency on Internet, which affects their work, study and living severely. This phenomenon is named as Internet Addiction Disorder (IAD) by Goldberg in 1996 [1] or Pathologica Internet Use (PIU) by Young [2]. A common approach to diagnosing PIU or IAD is based on diagnostic questionnaire made by medical or psychology specialists such as Young's 20 items questionnaire [3] and Beard's 5 criteria [4]. However, to our best knowledge, no

L. Cao et al. (Eds.): ADMI 2013, LNAI 8316, pp. 66–78, 2014.
DOI: 10.1007/978-3-642-55192-5_6,

existing work discusses how to diagnose PIU according to computer-mediated interaction information.

Aiming at the issues mentioned above, in this paper, we proposed an emerging pattern detecting model, named PIU-Miner, to detect PIU. Referring to the emerging patterns (EPs) mining idea introduced by Li *et al.* [6], PIU-Miner only mined minimum EPs of interaction events, i.e., Generators, to detect PIU. Instead of mining all emerging patterns of interactive events, the efficiency of PIU-Miner model is improved dramatically. The main contributions of this paper are summarized as follows.

1. An emerging pattern discovering model, i.e., PIU-Miner, is proposed to detect PIU by mining computer-mediated interaction events, which successfully solve PIU discovering issues from computer's data mining viewpoint other than from traditional medical and psychology perspective.
2. Instead of mining all EPs of events, only minimum EPs, i.e., generators are mined in PIU-Miner model. At the same time, a score criterion of ranking generators to reduce the number of generators is also exploited. All these methods improve the mining effectiveness of PIU-Miner greatly.
3. For dealing with excessive fragmentation of complex event's occurrence time and lasting time, Occurrence Time Mapping Strategy (OTMS) and Duration Rounding Strategy (DRS) are proposed in this paper. Based on these two strategies, multiple complex events with similar semantic meaning can be merged to a certain degree so as to reduce excessive numbers of frequent event itemsets.
4. Extend frequent itemset mining to frequent event itemset mining, which gives the practical experience worth referring to generalize definitions, properties and corollaries in transaction database.

The rest of this paper is organized as follows. Section 2 introduces some preliminaries such as simple event, complex event, equivalence class and generator. The details of PIU-Miner model and generator mining algorithm of complex events are discussed in Sect. 3. In Sect. 4, experimental results and evaluation are described. Section 5 gives some related work, while Sect. 6 concludes the paper and suggests the future work.

2 Preliminaries

Before introducing the PIU-Miner model, some basic concepts such as simple event, complex event, event equivalence class and event generator will be given in this section.

2.1 Simple Event

Events are real-world occurrences that unfold over space and time. In other words, an event is something notable that happens, owning a lasting time, occurs in a specific place, and typically will involve certain change of state. The formal

Table 1. Symbol and meaning of $Type_s$

Symbol	Meaning of $Type_s$
A	CPU utilization
B	Capacity of memory occupied
C_1	Number of clicking mouse's left buttons
C_2	Number of clicking mouse's right button
C_3	Amount of moving pixel of mouse
D	Number of pressing keyboards
EF	Network flow
F_p $(1 \leq p \leq 7)$	p^{th} front running process of monitored applications

definition about simple event and association rules to infer complex events are described in the following.

Definition 1. *Let E($type_s$,t_s,p_s,S_s) represent a simple event, where parameter $type_s$, t_s, and p_s represent type, time and place that a simple event occurs respectively. And S_s is an attribute set of different simple events, i.e.,S_s =(S_1,S_2,..., S_n), where S_i $(1 \leq i \leq n)$ is the i^{th} attribute in S_s.*

Since all simple computer-mediated interactive events occur in a common place, parameter p_s can be omitted in this paper. In computer-mediated interactions, a simple event's S_s has an unique attribute depending on $type_s$. Therefore, both $type_s$ and different values of S_s's unique attribute can act as a monitoring measure together. Because of this, simple event $E(type_s,t_s,p_s,S_s)$ can be simplified into the form $E(Type_s, t_s)$, where parameter $Type_s$ not only reflects an event type but also gives its measurable value. Further, in order to facilitate discussing, we will use abbreviation E to replace $E(Type_s, t_s)$ in the following unless otherwise specified.

We focus on 8 aspects of simple events relating to computer-mediated interactions, which are (1) CPU utilization, (2) capacity of memory occupied, (3) the number of clicking mouse's left buttons, (4) the number of clicking mouse's right buttons, (5) the amount of moving pixel of mouse, (6) the number of pressing keyboards, (7) network flow and (8) a front running process of monitored applications. In fact, since there are lots of applications in real computer world, it is unrealistic to monitor all applications. Therefore, for simplifying, only 7 typical processes are selected, denoted by F_p $(1 \leq p \leq 7)$, which are IE explorer, Google explorer, War3 (a real time strategy game), Trading Card Game Online (a board game), Windows Media Player, QQ (an instant message software) and MSN. The different values of $Type_s$ are listed in Table 1. Based on the 8 aspects mentioned above, there are totally 8 types of simple events need to monitor, which are listed in the following.

1. Once A is over 50 %, an instance of E, $e(A,t)$, is captured and created.
2. Once B is over 60 %, an instance of E, $e(B,t)$, is captured and created.
3. Once C_1 is over 30, an instance of E, $e(C_1,t)$, is captured and created.
4. Once C_2 is over 10, an instance of E, $e(C_2,t)$, is captured and created.

5. Once C_3 is over 1600, an instance of E, $e(C_3,t)$, is captured and created.
6. Once D is over 100, an instance of E, $e(D,t)$, is captured and created.
7. Once EF is over 40 MB, an instance of E, $e(EF,t)$, is captured and created.
8. Once F_p is running, one of instances of E, $e(F_p,t)$, is captured and created.

2.2 Complex Event

Let $cE(type_c, t_c, p_c, S_c)$ represent a complex event, where each parameter's subscript c distinguishes complex events from simple events. Above all, parameter p_c can be omitted due to the same reason as that of simple events. Second, similar with simple event's characteristic of unique attribute, it is enough for complex events to let S_c only record their lasting time. Thus, based on two points just mentioned, $cE(type_c, t_c, p_c, S_c)$ can be simplified into the form $cE(type_c, t_c, dur)$, where the first two parameters represent the type and time that a complex event occurs and the last parameter dur represents event's duration time. In addition, substitute $Type_c$ for $type_c$ in order to keep coincident with the type expression of simple events. As a result, $cE(type_c, t_c, dur)$ is written into $cE(Type_c, t_c, dur)$. The different values and meanings of parameter $Type_c$ are listed in Table 2.

Table 2. $Type_c$ and Weight

Time Interval	Meaning	Weight
$WVOn$	Watching Video Online	0.198
$WVOff$	Watching Video Offline	0.131
$PRTSG$	Playing Real Time Game	0.205
PBG	Playing Board Game	0.167
BWS	Browsing Web Site	0.155
COn	Chatting Online	0.143
DL	DownLoading	0.001

Table 3. T and Weight

Time interval	T	Weight
$6am-11am$	$morning$	0.14
$11am-14pm$	$noon$	0.16
$14pm-18pm$	$afternoon$	0.14
$18pm-23pm$	$evening$	0.23
$23pm-6am$	$before\ dawn$	0.33

The association rules for identifying complex events are represented in a disjunctive normal form. Let R denote association rules set, then $R = (r_1 \vee r_2 \vee ... \vee r_w)$, $(1 \leq w \leq 7)$, where $v_i's$ are the disjuncts. Each rule can be expressed in Formula (1).

$$r_v : (Condition_v) \rightarrow cE(Type_c^v, t, dur) \tag{1}$$

The left-hand side of the rule is called the rule antecedent or precondition. It contains a disjunctive normal of the conjunction of simple event tests, which is shown in Formula (2).

$$\begin{aligned} Condition_v = & [e(Type_s^1, t) \wedge e(Type_s^2, t) \wedge ... \wedge e(Type_s^m, t)] \\ & \vee [e(Type_s^1, t) \wedge e(Type_s^2, t) \wedge ... \wedge e(Type_s^h, t)] \\ & \vee ... \\ & \vee [e(Type_s^1, t) \wedge e(Type_s^2, t) \wedge ... \wedge e(Type_s^g, t)] \end{aligned} \tag{2}$$

Table 4. Association Rules of Generating Complex Events

r_1: $e(A, t) \wedge e(B, t) \wedge e(EF, t) \rightarrow ce(WVOn, t, dur)$
r_2: $e(A, t) \wedge e(B, t) \wedge e(EF_1, t) \rightarrow ce(WVOff, t, dur)$
r_3: $e(A, t) \wedge e(B, t) \wedge e(C_1, t) \wedge e(C_2, t) \wedge e(C_3, t) \wedge e(D, t) \wedge e(F_4, t)$
$\rightarrow ce(PRTSG, t, dur)$
r_4: $[e(A, t) \wedge e(B, t) \wedge e(C_1, t) \wedge e(C_3, t) \wedge e(F_3, t)]$
$\vee [e(A, t) \wedge e(C_1, t) \wedge e(C_3, t) \wedge e(F_3, t)] \rightarrow ce(PBG, t, dur)$
r_5: $[e(C_3, t) \wedge e(F_1, t)] \vee [e(C_3, t) \wedge e(F_2, t2] \vee [e(C_3, t) \wedge e(F_1, t) \wedge e(EF, t)]$
$\vee [e(C_3, t) \wedge e(F_2, t) \wedge e(EF, t)] \vee [e(C_1, t) \wedge e(C_3, t) \wedge e(F_1, t)]$
$\vee [e(C_1, t) \wedge e(C_3, t) \wedge e(F_2, t)] \vee [e(C_1, t) \wedge e(C_3, t) \wedge e(F_1, t) \wedge e(EF, t)]$
$\vee [e(C_1, t) \wedge e(C_3, t) \wedge e(F_2, t) \wedge e(EF, t)]$
$\rightarrow ce(BWS, t, dur)$
r_6: $[e(D, t) \wedge e(F_6, t)] \vee [e(D, t) \wedge e(F_7, t)] \vee [e(D, t) \wedge e(C_3, t) \wedge e(F_6, t)]$
$\vee [e(D, t) \wedge e(C_3, t) \wedge e(F_7, t)] \rightarrow ce(PBG, t, dur)$
r_7: $e(EF, t) \rightarrow ce(DL, t, dur)$

where $1 \leq (m,h,g) \leq 8$. Parameter m, g, h represent the number of simple events in a different conjunction normal. The right-hand side of the rule is called the rule consequent. If the precondition of r_v is satisfied, then r_v is said to be triggered, which results in the generation of $ce(Type_c^v, t, dur)$, i.e., an instance of a complex event. Association rules to deduce complex events are shown in Table 4. It is obvious that there are 7 association rules, which results in $1 \leq v \leq 7$. For example, if $e(A, t)$, $e(B, t)$ and $e(EF, t)$ are monitored simultaneously at time t, rule r_1 will be triggered. As a result, $ce(WVOn, t, dur)$ will be generated. In other words, we can induce that the computer user is watching video online at time t with dur=null because t is a time point. In order to obtain the duration time of $ce(Type_c^v, t, dur)$, an approach to obtaining complex event's lasting time will be exploited, which is introduced in next paragraph in detail.

During computing the lasting time of complex event, an interesting phenomenon is observed. Many complex events with a same event type are treated as different events just because their occurring time or lasting time is different. In fact, if time is limited to a reasonable range, these events have no obvious distinction. For example, given two complex events, "Surfing the Internet starting at 8:00 a.m. for 47 min" and "Surfing the Internet at 9:00 a.m. for 52 min", it is obvious that both of them have little semantic difference in real life, as they all happen in morning and the duration difference is not too much. However, two independent complex events are generated in event processing. This phenomenon results in the number of frequent complex events is too much, here, which is called Excessive Fragmentation of Time (EFT). For avoiding this issue, complex events can be merged together based on some coarse time granularity. Premise is this reduction has no negative effect on the precision of emerging pattern mining.

Considering the time semantic nature of real life, it is reasonable to partition a day with 24 h into 5 time intervals, i.e., (6:00 a.m. - 11:00 a.m.), (11:00 a.m. - 14:00 p.m.), (14:00 p.m. - 18:00 p.m.), (18:00 p.m. - 23:00 p.m.) and (23:00 p.m. -

6:00 a.m.) in this paper, and each of them represents "morning", "noon", "afternoon", "evening" and "before dawn" respectively. For avoiding EFT effects on both t_c and dur, two solution strategies, i.e., Occurrence Time Mapping Strategy (OTMS) and Duration Rounding Strategy (DRS), are proposed in this paper. The basic idea of OTMS is to merge some complex events into a group, where one to one mapping relationship between groups and 5 time intervals need to satisfy. For emphasizing the semantic meaning of occurrence time, parameter t_c is replaced with T_c. The basic idea of DRS is to let duration is the multiple times of integer 10, here, time unit is minute, and if not, round it. Both OTMS and DRS all decrease the number of complex events dramatically. Finally, complex event's abstract form $cE(Type_c, t_c, dur)$ is represented as $CE(Type_c, T, Dur)$. Table 3 lists 5 values of T in detail. In addition, in order to facilitate discussing, we will use abbreviation CE and Ce, to replace $CE(Type_c, T, Dur)$ and its instance respectively, in the following unless otherwise specified.

2.3 Equivalence Class and Generator of Complex Event

A CE dataset is a set of CE transactions. A CE transaction is a non-empty set of CEs. The key idea of PIU-Miner is to mine a concise representation of equivalence classes of frequent CE set from a transactional CE database D. Formally, an equivalence class of CEs is defined as follows.

Definition 2. *Let EC_{CE} represent an equivalence class of CEs. EC_{CE} is a set of CEs that always occur together in some CE transactions of D. That is, $\forall X, Y \in EC_{CE}, \exists f_D(X) = f_D(Y)$, where $f_D(Z) = \{R \in D \mid Z \subseteq R\}$.*

Definition 3. *The support of an CE itemset P_{CE} in a dataset D, denoted by sup(P_{CE}, D), is the percentage of CE transactions in D that contain P_{CE}.*

Definition 4. *Let X be a CE itemset of a CE dataset D. The equivalence class of X in D is denoted $[X]_D$. The maximal CE itemset and the minimal itemsets of $[X]_D$ are called the closed pattern and the generators of this CE equivalence class respectively. Generator of $[X]_D$ is represented as G. The closed patterns and generators of D are all the closed patterns and generators of their equivalence classes.*

Property 1. Let C_{CE} be the closed pattern of an equivalence class EC_{CE} and G_{CE} be a generator of EC_{CE}. Then all CE itemset X satisfying $G_{CE} \subseteq X \subseteq C_{CE}$ are also in this equivalence class.

Corollary 1. *An equivalence class EC_{CE} can be uniquely and concisely represented by a closed pattern C_{CE} and a set G_{CE} of generators, in the form of $EC_{CE} = [G_{CE}, C_{CE}]$, where $[G_{CE}, C_{CE}] = \{X \mid \exists g \in G_{CE}, g \subseteq X \subseteq C_{CE}\}$.*

Corollary 2. *The entire equivalence class can be concisely bounded as EC_{CE} = [G_{CE}, C_{CE}], where G_{CE} is the set of generators of EC_{CE}, C_{CE} is the closed pattern, and [G_{CE},C_{CE}] = $\{X \mid \exists g \in G_{CE}, g \subseteq X \subseteq C_{CE}\}$.*

In order to facilitating discuss, generator G_{CE} is abbreviated to G in the following unless otherwise specified.

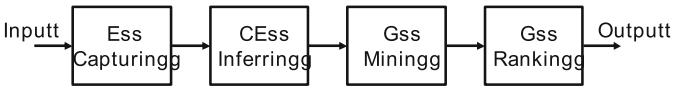


Fig. 1. PIU-Miner Model

3 PIU-Miner Model

PIU-Miner model, which is shown in Fig. 1, consists of four processing modules, 1) *Es* Monitoring Module for capturing $E(Type_s,t_s)$, 2) *CEs* Inferring Module for inducing $CE(Type_c,T,Dur)$ based on OTMS and DRS strategies, 3) *Gs* Mining Module for discovering PIU generators, and 4) *Gs* Ranking Module for scoring PIU generators to select optimal emerging patterns. In PIU-Miner model, user's interactions with computer are inputs and optimal PIU generators are outputs. Algorithm 1 further describes the total processing steps of PIU-Miner. Since *Gs* Mining Module is the core of PIU-Miner, Sect. 3.1 illustrated the detailed mining procedure. Section 3.2 gives the score formula to rank generators.

3.1 Discovering Generators of PIU

Two algorithms are introduced in this section, one is PIU Detecting Algorithm (PDA) and the other is Generator Mining Algorithm (GMA). The former illustrates the global processing steps and the latter focuses on how to mine generator, i.e., the lower bound of equivalence class of *CE itemset* in detail.

Algorithm 1. PIU Detecting Algorithm

Input: *SESet*, *CE*, *CESet*, *ESet*;
Output: *FG*

```
1: while (current time is out of sliding window) do
2:    while (simple event is not coming any longer) do
3:       ESet ← CapturingSimpleEvent();
4:       cE ← CreatingComplexEvent(ESet);
5:       insert cE to cESet;
6:    end while
7:    CESet ← MergingEvent(cESet);
8: end while
9: FG ← MiningGenerator(CESet);
10: OFG ← RankingGenrator(FG);
```

Given that *ESet* is a set storing simple event *E*, *cE* is a complex event inferred by *Es*, *cESet* is a set storing *cEs*, *CESet* is a set storing *CEs* processed by OTMS and DRS strategies, *FG* is the generator set, and *OFG* is the optimal generator set. PDA algorithm is given in Algorithm 1 and its processing

procedure includes 4 steps mainly. First, capture simple event set in last x time unit. Second, create a cE set based on simple event set captured. Third, cEs are mapped into CEs and their time durations are obtained. Finally, mine Gs based on the CEs resulting from merge processing.

GM algorithm is illustrated in Algorithm 2. Given that l is a frequent generator, D_l is the conditional database of l stored in an FP-tree [6] and FG is the set of Generators. Here, we use the modified FP-tree that the frequent events with a full support, i.e., those events appear every day, must be removed from the head table of FP-trees. After the modified FP-tree is constructed, the subsequent mining is operated on top of such tree. As an input of Mining Generator Algorithm, the details for frequent generator and conditional database are in [7].

Algorithm 2. Generator Mining Algorithm

Input: l, D_l
Output: FG
1: **for** l **do**
2: scan D_l to count frequent events and sort them into descending frequency order, denoted as $F_l = \{ a_1, a_2, ..., a_m \}$;
3: **for** $a_i \in F_l$ **do**
4: if sup($l \cup \{a_i\}$, D) = sup (l, D) then $F_l = F_l$ - $\{a_i\}$
5: **end for**
6: **for** $a_l \in F_l$ **do**
7: if $\exists\ l' \subset l \cup \{a_i\}$ and sup(l',D)= sup(l $\cup \{a_i\}$, D) then $F_l = F_l$ - $\{a_i\}$
8: else insert $l \cup \{a_i\}$ into FG
9: **end for**
10: **end for**

3.2 Ranking Generators of PIU

Since we have mined a set of generators FG, then our work is ranking these generators, identifying which one enables to represent PIU markedly. In this process, experts in PIU domain are involved in detecting work, scoring each one in generator set and rank the useful ones. According to knowledge and experiences, a formula for scoring every generator is given in Formula (3). Just like mentioned in Sect. 2.2, in Formula (3), $Type_c$ represents the type of frequent CE , T represents a CE's time interval projected by OTMS and Dur is CE's lasting time. The weights of $Type_c$ and T are listed in Table 2 and 3 respectively. Parameter k points out the number of CEs included in a generator G. After scoring all the generators, the generator set whose score less than threshold r_s will be pruned. The remaining ones are the final emerging patterns to represent PIU optimally.

$$Score(G_i\langle\bigcup_{k=1}^{n} CE_k\rangle) = \sum_{k=1}^{n} weight(Type_c^k) \times weight(T_k) \times Dur_k \qquad (3)$$

Table 5. The first week's Gs

G_1 ⟨ $Ce(COn, evening, 30)$, $Ce(BWS, before\ dawn, 20)$ ⟩
G_2 ⟨ $Ce(WVOn, evening, 80)$, $Ce(BWS, noon, 20)$ ⟩
G_3 ⟨ $Ce(WVOn, before\ dawn, 80)$, $Ce(COn, evening, 30)$ ⟩
G_4 ⟨ $Ce(WVOn, before\ dawn, 80)$ ⟩

4 Experiment Results and Evaluation

In this paper, PIU-Miner model is proposed for detecting PIU from computer-mediated interaction events. For testing the performance of PIU-Miner model, extensive experiments are exploited in this section. All the experiments are done with Intel Core i3 processor (2.53 GHz CPU with 4GB RAM) and operating system is Windows 7 professional edition.

Among 20 students working in our college, we select 5 persons who have high PIU probability such as watching video online, playing board game, and Browsing Web Site by questionnaires as testing objects. We collected their daily computer interactions about 10 weeks as sample data set. For easy to understand generators, an example of generators mined by PIU-Miner model, i.e., the first week's Gs, is shown in Table 5. PIU-Miner model discovers 4 generators for detecting PIU in the first week, which tell us some clues about a person who has some PIU possibilities of watching video online, chatting on line and browsing web site frequently.

Since threshold r_s is a key factor that effects the precision of generators, precision values as threshold varies are tested and is shown in Fig. 2. Every precision measure is the average values of 10 weeks' accuracy data. It is obvious that r_s=3.5 is the best selection for our testing sample data. Threshold r_s=3.5 means that the generators will be pruned if their scores are less than 3.5. Based on threshold 3.5, we statistic the precision measures for detecting PIU based on discovered generators every week and then give the corresponding results in Fig. 3. This histogram graph indicates the best precision is up to 0.84 and the lowest precision is 0.6. Figure 3 proves that the generators mined by PIU-Miner are effective.

5 Related Work

Discovering frequent patterns from large database is meaningful and practical in association rule mining [8,9], correlation analysis [10], classification [11], emerging pattern [12] and other domains. Some frequent pattern approaches, like Apriori [13] and FP-growth [7], base on a single minimum support (minsup), thus confront a dilemma when mining frequent patterns consisting of both frequent and rare items. That is, at high minsup, frequent patterns involving rare items will be missed, and at low minsup, the number of frequent patterns explodes. To solve this problem, the approach using multiple minsup constraints is raised in [14–18]. There are mainly three types of mining frequent patterns approach.

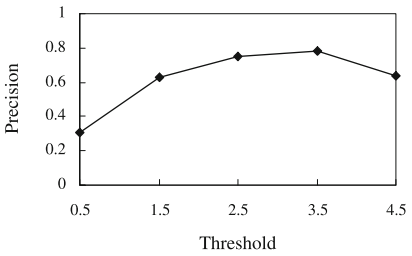


Fig. 2. Precision of Gs under r_s

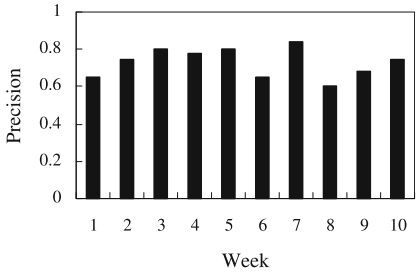


Fig. 3. Precision of 10 weeks' Gs

The first is traversing iteratively the set of all patterns. The most prominent algorithm based on this approach is Apriori and many improved algorithms [19–22] arose later. The second is extracting maximal frequent patterns of which all supersets are infrequent and all subsets are frequent. The most prominent algorithm based on this approach is Max-Miner [23]. The third is based on the theoretical framework [24] that uses the closure of the Galois connection [25]. The most representative is Close algorithm [26].

Sequential pattern mining was first introduced by Agrawal and Strikant [27] used in data mining research field. In sequential pattern mining process, the algorithm only mines the closed patterns instead of all frequent ones, for the former leads to a complete yet compact result thus better efficiency. Agrawal *et al.* [28] developed a generalized and refined algorithm, GSP, based on the Apriori property [8]. Since then, many improved algorithm have been proposed. SPADE [29], PrefixSpan [30] and SPAM [31] are quite popular ones. All these search strategies can be divided into two types [32] - breadth-first search, such as GSP and SPADE, generating many candidate patterns, and depth-first search, such as PrefixSpan and SPAM, iteratively partitioning the original data set. While most of previously developed closed pattern mining algorithms are inherently costly in both runtime and space usage when the support threshold is low or the patterns become long, Wang *et al.* [33] presented an efficient algorithm for mining frequent closed sequences, BIDE, without candidate maintenance. In recent years, the studying domain of sequential pattern mining has been extended. Since existing work of studying the problem of frequent sequence generator mining is rare, Gao *et al.* [34] present a novel algorithm, FEAT (abbr. frequent sequence generator miner), to perform this work. Yan *et al.* [35] studied the closed pattern mining problem and proposed the CloSpan algorithm. Zhang *et al.* [36] studied the sequential pattern mining from a single long sequence with gap constraints. Sequential pattern mining is widely used in many domains. For instance, in text mining, sequential patterns are mined in a single document and a document collection. It is also successfully applied to question answering [37] and authorship attribution extraction [38].

Internet addiction was first introduced by Young [39], but the illustrate of term addiction is various between scholars and has greatly developed. In paper [1], Young defined Internet addiction as impulse-control disorder by using Pathological Gambling as a model. The behavior was first named as Pathological Compulsive Internet Usage (PCIU) and later changed to Pathological Internet Use (PIU), divided into 5 types - information overload, net compulsions, cyber-relationship addiction, cyber-sexual addiction and game addiction [40]. There are two approaches for PIU diagnosis. One is based on The Diagnostic and Statistical Manual of Mental Disorders (DSM). A representative is Coldberg's idea. He came up with 6 criteria according to DSM and anyone who reaches these criteria can be diagnosed as PIU [1]. Another is based on cognitive-behavioral criteria. Davis believes that cognitive-behavior should be considered in PIU diagnosis and he divides PIU into two types - Specific Pathological Internet Use (SPIU) and Generalized Pathological Internet Use (GPIU), thus can treat characteristic and generality in a different way [41].

6 Conclusion and Future Work

Be absorbed in Internet so addictively as to unable to control, this phenomenon is called PIU or IAD. PIU is a negative production of Internet popularizing and need to avoid. Aiming at this issue, a novel PIU-Miner model is proposed in this paper. The basic idea of PIU-Miner is to mine Generators only (i.e., a minimum event pattern set), other than all possible event emerging patterns, for detecting PIU group from Internet users. Extensive experimental results show that Generators-based event pattern discovering method for diagnosing PIU is efficient and effective.

Future work will focus on how to build PIU classifiers based on mined generators. For achieving this motivation, we will take domain knowledge such as medical science, psychology and sociology into account, which would be helpful to improve the accuracy of classifiers further.

References

1. Goldberg: Internet addiction disorder. http://www.psycom.net
2. Young, K.: Internet addiction: symptoms, evaluation and treatment. J. Innov. Clin. Pract. **17**, 19–31 (1999)
3. Young, K.: Caught in the Net. Wiley, New York (1998)
4. Beard, W., Wolf, M.: Modification in the proposed diagnostic criteria for internet addiction. J. Cyberpsychol. Behav. **3**, 377–383 (2001)
5. Cao, L., Gorodetsky, V., Mitkas, P.: Agent mining: the synergy of agents and data mining. IEEE Intell. Syst. **24**(3), 64–72 (2009)
6. Li, J., Liu, G., Wong, L.: Mining statistically important equivalence classes and delta discriminative emerging patterns. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 430–439 (2007)

7. Han, J., Pei, J., Yin, Y., Mao, R.: Mining frequent patterns without candidate generation: a frequent-pattern tree approach. J. Data Min. Knowl. Disc. **8**(1), 53–87 (2004)
8. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Proceedings of 20th International Conference on Very Large Data Bases, pp. 487–499 (1994)
9. Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H., Verkamo, A.I.: Finding interesting rules from large sets of discovered association rules. In: Proceedings of the Third International Conference on Information and Knowledge Management, pp. 401–408 (1994)
10. Brin, S., Motwani, R., Silverstein, C.: Beyond market basket: generalizing association rules to correlations. In: Proceedings ACM SIGMOD International Conference on Management of Data, pp. 265–276 (1997)
11. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, pp. 80–86 (1998)
12. Dong, G., Li, J.: Efficient mining of emerging patterns: discovering trends and differences. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 43–52 (1999)
13. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pp. 207–216 (1993)
14. Liu, B., Hsu, W., Ma, Y.: Mining association rules with multiple minimum supports. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 337–341 (1999)
15. Lee, Y., Hong, T., Lin, W.: Mining association rules with multiple minimum supports using maximum constraints. Int. J. Approx. Reason **40**(1–2), 44–54 (2005)
16. Hu, Y., Chen, Y.: Mining association rules with multiple minimum supports: a new algorithm and a support tuning mechanism. J. Decis. Support Syst. **42**(1), 1–24 (2006)
17. Uday Kiran, R., Krishna Reddy, P.: An improved multiple minimum support based approach to mine rare association rules. In: Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, pp. 340–347 (2009)
18. Uday Kiran, R., Krishna Reddy, P.: An improved frequent pattern-growth approach to discover rare association rules. In: Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, pp. 43–52 (2009)
19. Brin S., Motwani R., Ullman J., Tsur S.: Dynamic itemset counting and implication rules for market basket data. In: Proceedings ACM SIGMOD International Conference on Management of Data, pp. 255–264 (1997)
20. Park, J., Chen, M., Yu, P.: An efficient hash based algorithm for mining association rules. In: Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, pp. 175–186 (1995)
21. Savasere, A., Omiecinski, E., Navathe, S.: An efficient algorithm for mining association rules in large databases. In: Proceedings of 21th International Conference on Very Large Data Bases, pp. 432–444 (1995)
22. Voivonen, H.: Sampling large databases for association rules. In: Proceedings of 22th International Conference on Very Large Data Bases, pp. 134–145 (1996)
23. Bayardo, R.: Efficiently mining long patterns from databases. In: Proceedings ACM SIGMOD International Conference on Management of Data, pp. 85–93 (1998)
24. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Pruning closed itemset lattices for association rules. In: BDA, pp. 177–196 (1998)

25. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Heidelberg (1999)
26. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Efficient mining of association rules using closed itemset lattices. J. Inf. Syst. **24**(1), 25–46 (1999)
27. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Proceedings of the Eleventh International Conference on Data Engineering, pp. 3–14 (1995)
28. Srikant, R., Afrawal, R.: Mining sequential patterns: generalizations and performance improvements. In: Proceedings of the 5th International Conference on Extending Database Technology, pp. 3–17 (1996)
29. Zaki, M.: Spade: an efficient algorithm for mining frequent sequences. J. Mach. Learn. **42**, 31–60 (2001)
30. Pei, J., Han, J., Mortazavi-Asl, B., Chen, Q., Dayal, U., Hsu, M.: PrefixSpan: mining sequential patterns by prefix-projected pattern growth. In: Proceedings of the 17th International Conference on Data Engineering, pp. 215–224 (2001)
31. Ayresm, J., Gehrke, J., Yiu, T., Flannick, J.: Sequential pattern mining using a bitmap representation. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 429–435 (2002)
32. Feng, J., Xie, F., Hu, X., Li, P., Cao, J., Wu, X.: Keyword extraction based on sequential pattern mining. In: The Third International Conference on Internet Multimedia Computing and Service, pp. 34–38 (2011)
33. Wang, J., Han, J.: BIDE: efficient mining of frequent closed sequences. In: Proceedings of the 20th International Conference on Data Engineering, pp. 79–90 (2004)
34. Gao, C., Wang, J., He, Y., Zhou, L.: Efficient mining of frequent sequence generators. In: Proceedings of the 17th International Conference on World Wide Web, pp. 1051–1052 (2008)
35. Yan, X., Han, J., Afshar, R.: CloSpan: mining closed sequential patterns in large datasets. In: Proceedings of the Third SIAM International Conference on Data Mining, pp. 166–177 (2003)
36. Zhang, M., Kao, B., Cheung, D., Yip, K.: Mining periodic patterns with gap requirement from sequences. J. ACM Trans. Knowl. Discov. Data **1**(2) (2007)
37. Denicia-Carral, C., Montes-y-Gómez, M., Villaseñor-Pineda, L., Hernández, R.G.: A text mining approach for definition question answering. In: Salakoski, T., Ginter, F., Pyysalo, S., Pahikkala, T. (eds.) FinTAL 2006. LNCS (LNAI), vol. 4139, pp. 76–86. Springer, Heidelberg (2006)
38. Coyotl-Morales, R.M., Villaseñor-Pineda, L., Montes-y-Gómez, M., Rosso, P.: Authorship attribution using word sequences. In: Martínez-Trinidad, J.F., Carrasco Ochoa, J.A., Kittler, J. (eds.) CIARP 2006. LNCS, vol. 4225, pp. 844–853. Springer, Heidelberg (2006)
39. Young, K.: Internet addiction: the emergence of a new clinical disorder. J. Cyberpsychol. Behav. **1**(3), 237–244 (1996)
40. Young, K.: Internet addiction: a new clinical phenomenon and its consequences. J. Am. Behav. Sci. **48**, 402–415 (2004)
41. Davis, R.: A cognitive-behavioral model of pathological internet use. J. Comput. Hum. Behav. **17**(2), 187–195 (2001)

Learning Heterogeneous Coupling Relationships Between Non-IID Terms

Mu Li[1(✉)], Jinjiu Li[1], Yuming Ou[1], Ya Zhang[2], Dan Luo[1], Maninder Bahtia[3], and Longbing Cao[1]

[1] University of Technology, Sydney, Australia
mu.li@student.uts.edu.au
[2] Shanghai Jiaotong University, Xuhui, China
[3] Australian Taxation Office, Adelaide, Australia

Abstract. With the rapid proliferation of social media and online community, a vast amount of text data has been generated. Discovering the insightful value of the text data has increased its importance, a variety of text mining and process algorithms have been created in the recent years such as classification, clustering, similarity comparison. Most previous research uses a vector-space model for text representation and analysis. However, the vector-space model does not utilise the information about the relationships between the term to term. Moreover, the classic classification methods also ignore the relationships between each text document to another. In other word, the traditional text mining techniques assume the relation between terms and between documents are independent and identically distributed (iid). In this paper, we will introduce a novel term representation by involving the coupled relations from term to term. This coupled representation provides much richer information that enables us to create a coupled similarity metric for measuring document similarity, and a coupled document similarity based K-Nearest centroid classifier will be applied to the classification task. Experiments verify the proposed approach outperforming the classic vector-space based classifier, and show potential advantages and richness in exploring the other text mining tasks.

Keywords: Non-iid · Coupled similarity · Vector representation

1 Introduction

Text processing and agent mining methods have seen increasing interest in recent years, because of a variety of applications such as social media, blogs, and online communities [5]. The most general form of text data are in strings, and the most common representation for text is the vector-space representation. The vector-space model represents the text for each document as a "bag-of-words". Though the vector-space representation is very efficient, it loses information about the structural information of the words in the document, especially when used it purely in the form of individual word presentations.

L. Cao et al. (Eds.): ADMI 2013, LNAI 8316, pp. 79–91, 2014.
DOI: 10.1007/978-3-642-55192-5_7,

In many applications, the "unordered bag of words" representation is insufficient for finding the analytical insights, especially in the case for fine-grained applications, which the structure of the documents affect the underlying semantics. Intuitively, the advantage of the vector-space representation is the simplicity lends itself to straightforward processing, however, the vector-space representation is very inaccurate because it does not include any information about the ordering of the term in the document. Additionally, as it shows in the Fig. 1 the vector-space model implement the term frequency-inverse document frequency (TFIDF) of each term as the feature of one document, the discriminative power of this approach not strong on the low frequency term because many low frequency terms share a relative same TFIDF value. In the fact that, most of the text documents are made by these low frequency terms. Therefore, it is very hard to distinguish the similarity among those documents, regardless using Euclidean distance or Cosine distance. There are some examples from the Reuters-21578 R8 data set, we take three for instance:

noranda to spin off forest interests into separate company

burlington northern inc st qtr shr profit cts vs loss dlrs

api says distillate stocks off mln bbls gasoline off crude up

Clearly, the three documents have a different topic. However, the vector-space model representation lack of ability to distinguish them from each other, because many terms's TFIDF value in these documents is relative low and similar.

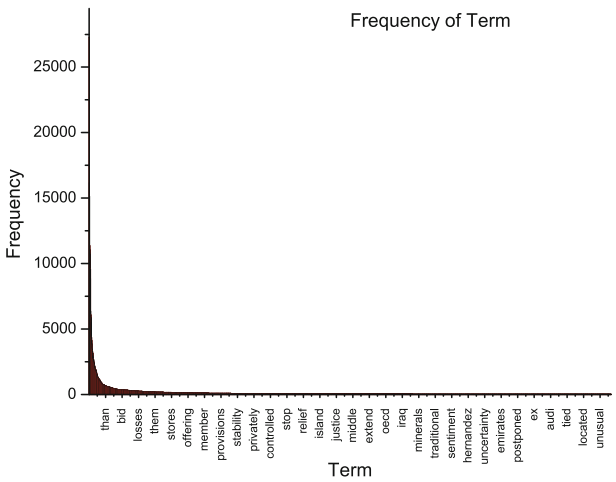


Fig. 1. Term frequency in Reuters-21578 R8

To prevent the aforementioned weakness of the previous research, the key contributions of this paper are as follows:

– First, we proposed a novel method that capture the order information of the terms by aggregate information from the term's most co-occurred neighborhood.
– Second, we build a vector presentation for each term instead of the scalar value which used in the traditional vector space model. By doing this, we involve much richer contest information for document similarity comparison. We define it as the coupled similarity between document.
– Third, we proposed a novel classification method by involving the coupled similarity among each document.

This paper is organized as follows: In the next section, we explore the recent research on text representation and mining method. In Sect. 3, we illustrate the detail of neighborhood co-occurred based text term representation approach. In Sect. 4, we discuss the coupled similarity metric and the novel classification method. In Sect. 5, the experiment result will be explained. The conclusions and summary are presented in Sect. 6.

2 Related Work

The research to explore term's representations have a long history, early proposals can be found in [1–4]. Recently, many models have been proposed for involving the information over the "next" word to given words. For instance, it has been explored in approaches that are based on learning a clustering of the words: each word is associated deterministically or probabilistically with a discrete class, and terms in the same categories are in the same respect.

The concept of using vector-space representation for terms in the information retrieval also has been researched, where feature vectors for words are learned on the basis of their probability of co-occurring in the same documents [6]. An important difference is that this work looks for a representation for terms that is helpful in representing Non-iid relations [18] between each other from its context. This is similar to what has been done with documents for information retrieval with LSI. The idea of using a continuous representation for words has however been exploited successfully by in the context of an n-gram based statistical language model [7], using LSI to dynamically identify the topic of discourse.

Another main research area is using the neural networks to create representation of the terms, these models were first studied in the context of feed-forward networks [8], and later in the context of recurrent models [9,10]. Also the use of distributed topic representations has been studied in [11–13] proposed a semantically driven method for obtaining word representations, [23] underlined implicitly semantic relation into the traditional measure based on the co-occurrence frequency.

Meanwhile, there are many other works shown that the word vectors can be used to improve many NLP applications [14–16]. Estimation of the word vectors itself using different models on various corpora [17,19]. The word vectors are very useful to future research and comparison, but the most previous research

with independent and identically distributed assumption [20], this work propose a novel method, aims to capture the coupled relationships from not only term to term level but also document to document level. The coupled relation has been explored in structured numerical data [21,22], but only a few research talk about the coupled relation in the text mining task [23]. This work is to make a comprehensive framework to involve the coupled relations in the document classification task.

3 Vector Presentation of Terms

We develop a novel text representation method which by describing the term's feature in a vector. The vector contains information about the term not only IDF attribute as other methods does, but also considered the impact from the other terms. We defined a coupled relationships between term to term in two perspectives: one is the intra-coupled relationships which directly describes the term's feature by its most co-occurred neighbor's IDF value; while another is inter-coupled relationships, which involve the indirect information, when the two terms share some neighborhood. Moreover, the coupled relation can be accumulated in many ways, in this paper, we explored one possible way. Next two subsection illustrates the definition of intra-coupled and inter-coupled relationships respectively.

3.1 Intra-Coupled Relation

Firstly we discussed the direct relations between term to term which means the relations can be captured from the physical appearance of the terms in the documents. It is reasonable that there are some kind of relations between two terms when they appeared as neighbors. The proposed method is straightforward, by the Fig. 2, we make a sliding window with width w, the middle block is m, it also contains left blocks L and right blocks R. The term in the middle block is the key what we focus on, aiming to capture the impact from its neighbors. After initialization, the sliding window traversed all the text content, as a consequence, every term's neighbor can be discovered and the frequency can be calculated as well.

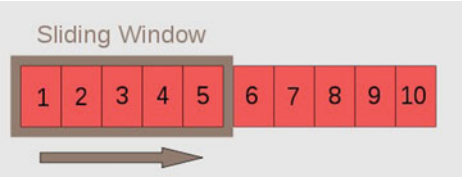


Fig. 2. Sliding window

Definition 1. *A set of terms* $T_\phi = \{t_1, t_2, \ldots, t_n\}$ *is a related terms to certain term* t_c *if the terms in* T *co-occurred within the width* w *of the sliding window. There is a mapping function* $F_\phi(t_c) = T$ *to find the correlated terms of* t_c.

The ϕ is defined by the different relation criterion, so the mapping function F_ϕ could be any format. The simplest way is to rank the frequency of the neighbors for a certain term t_c. In this paper, we use the neighborhood based relation criterion. In detail, we select top n ranked high frequency neighbor of t_c, and build their relation by choosing the neighbors in left and right side separately, and join them all together. The mapping function F_ϕ becomes F_l,F_r,$F_{l\&r}$, and the related terms set T_ϕ becomes T_l,T_r,$T_{l\&r}$ respectively.

Assume t_c as the center term, F_l override F_ϕ which finding the most frequent terms appear as the prefix of the t_c within w, while the $F_r(t_c)$ stands for terms appears in as the suffix of t_c within w. The traverse procedure with the purpose of finding the most high frequency neighbor terms T_ϕ of the center term t_c.

The term share the same meaning would have relative same neighbors based on this selection because the same meaning term should have relative same neighbors. At this stage, each term can be described by itself and its top ranked high related neighbors, more precisely, we use the IDF as the base numerical measure of each term, each high related neighbor's IDF values combined as a vector to describe a term t_c. For instance, set $n = 2$ to choose the 2 most relevant term and IDF value is defined as $IDF(t_{\phi_n})$ where ϕ is the relating method. After that, t_c can be represented as a vector $V_{t_c} = \{IDF(t_{l_1}), IDF(t_{l_2}), IDF(t_x), IDF(t_{r_1}), IDF_{t(r_2)}\}$ instead of just a single scalar representation $IDF(t_c)$.

Table 1. Most related terms of the key-term

2nd most in left	1st most in left	key-term	1st most in right	2nd most in right
credit	bank	guarantee	lead	intern
bid	offer	comparison	Illinois	year
affecting	of	liquidity	the	in
with	the	proxy	materials	statement
in	raw	material	sciences	on
subject	to	regulatory	approval	approvals

After determining the most revelent terms to every term t_c, the vector representation V_{t_c} of the term t_c available simultaneously. For simplicity, we defined the vector presentation $V_{t_c} = \{v_1, v_2, \ldots, v_n\}$,$n$ is the number that stands for top n high frequency neighbor, and $v_x = IDF_x$. As it shows in the Table 1, the key-terms in the table are all in a same range of IDF value, which means it cannot be distinguished from each other by the traditional method. However, with the proposed method, each term has it's unique neighborhood, by transforming the IDF into the vector presentation V_{t_x}, it is easy to distinguish every term from one to another. Also, if two terms that share the similar neighbor, in other words they have similar context, they can be judged as the synonyms. With this

advantage, many applications can capture more semantic meaning of each term instead of using a scalar IDF value as the only feature for each team.

The V_t is a vector which includes the IDF values of t_c most related terms, and v_c is the IDF value of t_c. we use $\gamma(t_c)$ to involve the information from related terms of t_c and $\gamma(t_c)$ also is a vector.

$$\gamma(t_c) = \frac{1}{nw} V_c / v_c \tag{1}$$

where n is the number of top ranked terms, and w is the sliding window size, divided by them aims to normalization.

Finally, with the normalization function $\gamma(t_c)$, the direct coupled relation we call intra-coupled similarity between two terms can be defined as follows:

$$\delta^{Ia}(t_i, t_j) = \frac{\gamma(t_i) \cdot \gamma(t_j)}{\sqrt{\gamma(t_i) \cdot \gamma(t_i)} \cdot \sqrt{\gamma(t_j) \cdot \gamma(t_j)}} \tag{2}$$

3.2 Inter-Coupled Relation

The intra-coupled relation is getting the information from the neighborhood most co-occurred directly. However, there are many terms, which have the relations, but not occurred in the sliding window. In this section, we proposed a novel method to detect the relations that not directly related.

The inter-coupled relation analysis is inspired by the fact that terms with the similar sense must appear in a similar context. Therefore, we explore the relation between a pair of terms by their most related terms, for simplicity we use the most co-occurred neighborhoods as an example.

As it shows in Fig. 3, the $Term1$ and $Term2$ does not have the direct connection which means they never co-occurred within the sliding window, however,

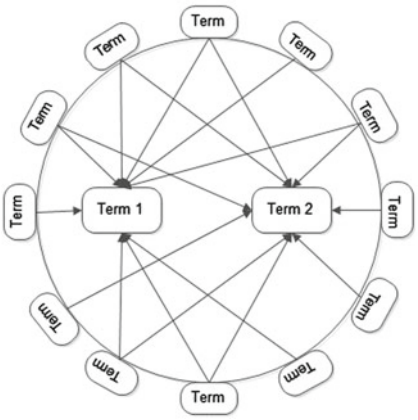


Fig. 3. Inter relation from term to term

we may find some indirect relations by its direct related terms. $Term1$ has some direct related terms while $Term2$ also has some, actually they share some direct related terms T_s, therefore, we can capture the indirect relation by accumulating all theirs shared direct related items. We set a threshold ϵ to restrict the minimal number s of share terms which to ignore the too small impact from its direct related terms. We defined the inter-coupled similarity between two terms as follows:

$$\delta^{Ie}(t_i, t_j) = \begin{cases} 0 & s < \epsilon \\ \sum_{t_s \in T_s} \delta^{Ia}(t_i, t_s) \cdot \delta^{Ia}(t_s, t_j) & s > \epsilon \end{cases} \tag{3}$$

The inter coupled relations involved richer context information for a term to term similarity than the traditional method. This is a fundamental criterion and can be applied on many applications as classification and clustering. The next section will discuss the document classification task based on the intra-coupled and inter-coupled relations.

4 Coupled Similarity Based Document Classification

With aforementioned definition or the coupled relations between term to term, we make a comprehensive coupled similarity term to term paradigm Cst as follows:

$$Cst(t_i, t_j) = \alpha\delta^{Ie}(t_i, t_j) + \beta\delta^{Ia}(t_i, t_j) \tag{4}$$

α and β is the parameter to adjust the weight for the importance of intra and inter coupled similarities. After defined the coupled similarity between term to term, we should define a new criterion for the coupled similarity measurement for the document to document. We try to capture the relation between terms by their coupled term to term similarity across the two documents $D = \{d_i, d_j\}$ which need to be compared. As a consequence we simply utilize the document-term matrix W which contain the coupled similarity Cst for each pair of terms shown in the documents. The coupled similarity between two documents by using corresponding kernel is expressed as:

$$Csd(d_i, d_j) = \overrightarrow{d_i} W \overrightarrow{d_j}^T \tag{5}$$

where $\overrightarrow{d_i} = tfidf(t_1, d), tfidf(t_2, d), \ldots, tfidf(t_n, d)$ is the vector space model presentation of the document.

Once we have the coupled similarity of the documents, the classification task is straightforward. If we use the coupled similarity of the document as the distance between two documents, we can simply apply the KNN algorithm as the classification method. Moreover, we make an adjust to the traditional KNN to make it adapted to the data set we plan to use. Firstly, the classification task is multi-label classification. Secondly, the computational cost of comparing two documents is high, therefore we should make the comparatione time as less as possible. To do this, we find the most λ representative document from each class

in the training set. A centroid document m_t of one class c_t is a document within the c_t which has maximal similarity to all other documents within the class, for any document d' in c_t, the centroid document m_t satisfy:

$$\sum_{d_i \in c_t} Csd(m_t, d_i) \leq \sum_{d_i \in c_t} Csd(d', d_i) \tag{6}$$

where $\{c_t\} = \{d_{t1}, d_{t2}, \ldots, d_{tn}\}$ is the class which contains some documents. After choosing the most λ representative documents into T_λ for each class, we run the classic KNN algorithm on the set T_λ to do the classification task.

5 Experiment and Evaluation

The purpose of this section is to illustrate the advantage after using vector representation of terms. This will be archived with extracting the coupled relation among terms. The both intra-coupled and inter-coupled relations will enhance the performance of the classification task, and the future optimization of many algorithms is possible, we leave this issue in the next section. The application of classification does provide qualitatively effective results. Furthermore, we will use different kinds of application to show that our results are not restricted to a specific data source, but can achieve effective results over a wide variety of applications.

All our experiments were performed on a quad core Intel I5 CPU, 4G memory, and running windows 7 enterprise. All algorithms were implemented in $C\#$ and running on the .Net Framework.

5.1 Data Sets

We choose three popular data sets used in traditional text mining and information retrieval applications in our experimental studies: (1) 20 newsgroups (2) Reuters-21578 and (3) WebKB. Furthermore, the Reuters-21578 data set is of two kinds: Reuters-21578 R8 and Reuters-21578 R52. So we have four different data sets in total. The 20 newsgroups data set includes 20,000 messages from 20 Usenet newsgroups, each of which has 1,000 Usenet articles. Each newsgroup is stored in a directory, which can be regarded as a class label, and each news article is stored as a separate file. The Reuters-21578 corpus5 is a widely used test collection for text mining research. The data was originally collected and labeled by Carnegie Group, Inc. and Reuters, Ltd. in the course of developing the CONSTRUE text categorization system. Due to the fact that the class distribution for the corpus is very skewed, two sub-collections: Reuters-21578 R52 and Reuters-21578 R8, are usually considered for text mining tasks. In our experimental studies, we make use of both of these two data sets to evaluate a series of different data mining algorithms. Every document in the aforementioned data sets is in raw formate, not to be preprocessed by eliminating non-alphanumeric symbols, specialized headers or tags and stop-words, and also not be stemmed. The proposed methods are defined with respect to this never processed representation.

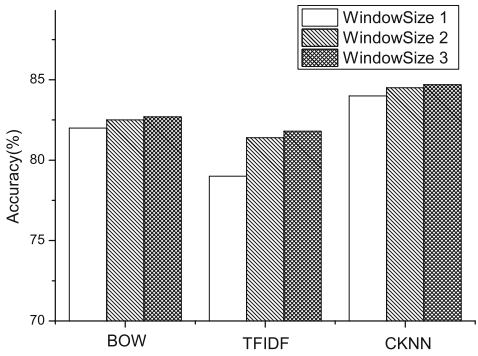


Fig. 4. Classification (20 newsgroup)

5.2 Experiment

In this section, we will first test the effectiveness of the coupled similarity on a variety of classification algorithms. Our algorithm reads and indexes text documents and builds the statistical model. Then, different text classification algorithms are performed upon the statistical model. We used three different algorithms for text classification. These algorithms are the Naive Bayes based on bag of words model, SVM classifier based on TFIDF vector representation of document and the proposed coupled similarity based K-Nearest centroid classifier respectively. For each classification method of interest, we employ the vector-space models including unigram, bigram and trigram models, and the sliding window size ranging from 1 to 3, respectively, as the underlying representational models for text classification. In order to simulate the behaviors of the bigram model and the trigram model, we extract the most frequent 100 doublets and triplets from the corpora and augment each document with such doublets and triplets, respectively. The vector-space models are therefore further categorized as unigram with no extra words augmentation, bigram with doublets augmentation and trigram with triplets augmentation. We conduct 5-fold cross validation for each algorithm in order to compare the classification accuracies derived from different representation strategies. All the reported classification accuracies are statistically significant with 95 % significance level.

In Fig. 4, we have illustrated the classification accuracy results in the 20 newsgroups data set for the three different classifiers. In addition to the vector space representations for unigram, bigram and trigram models (for simple display, those model also named as window size 1 to 3 respectively), we have also illustrated the classification results for the coupled distance representations with different sliding window size ranging from 1 to 3. It is clear that the addition of neighborhood information in the coupled similarity models improves the quality of the underlying result in most cases. Specifically, the best classification results are obtained for sliding window size 3 in K-Nearest centroid classifier.

Meanwhile, in all of the cases, the coupled similarity models consistently obtain better classification results than all the traditional vector-space models, including the unigram, the bigram and the trigram models. Even though the optimal classification accuracy is achieved for coupled similarity model with sliding windows size of 1 and 2 in some experimental scenarios, it is noteworthy that the vector-space representations did not even perform better than the bigger sliding window size of coupled similarity model in all cases.

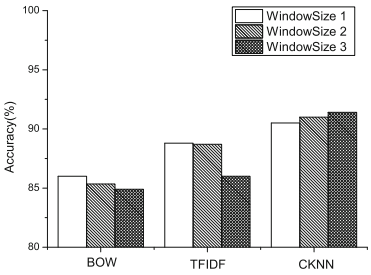


Fig. 5. Classification (Reuters-21578 R52)

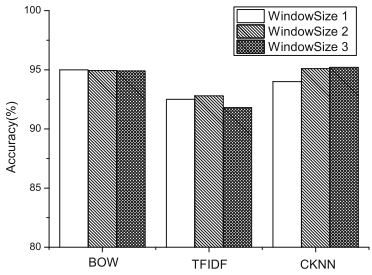


Fig. 6. Classification (Reuters-21578 R8)

We also tested the classification results for the Reuters-21578 (R8 and R52) data sets. The classification accuracy results are illustrated in Figs. 6 and 5, respectively. It is evident that the coupled similarity based Nearest K-centroid classifier is able to provide a higher classification accuracy over the different kinds of classifiers as compared to the vector-space representations. The reason for this is that the both intra and inter coupled similarity can capture structural information about the documents which is used in neighborhoods to help improve the classification accuracy. As a result, the classification results obtained with the use of the coupled similarity models are superior to those obtained using the vector-space representations.

The previous part of this section discussed the effective of accuracy between the traditional method and proposed method. Moreover, there are many parameters can be adjusted to fine tune the performance. As the Fig. 7 shows, we tried different sliding window types on different window size. The optimize window type is left, and the right neighborhood of the center term, just as we discussed in the Sect. 3. For window size, the peak point for different window type is uncertain, but most of the best performance is around 3 to 4, that is why we only tried 3 as the maximal window size in the previous experiment. The number of nearest neighborhood in KNN algorithm also has an impact the to the final performance, but it is most related to the KNN algorithm itself, we do not discuss it in this work.

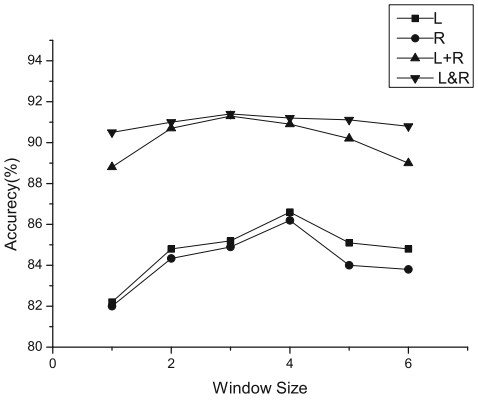


Fig. 7. Different window type

6 Conclusion

In this paper, we introduced the concept of coupled similarity from term to term based on its co-occurred as neighborhood in a sliding window, a new paradigm for text representation and processing. The coupled relation maintains information about the relative placement of words with respect to each other, and this provides a richer representation for document classification purposes. We can use this similarity in order to exploit the recent advancements in text mining algorithms. Furthermore, the coupled similarity criterion can be used with minimal changes to existing data mining algorithms if desired. Thus, the new coupled similarity framework does not require additional development of new data mining algorithms. This is a huge advantage since existing text processing and mining infrastructure can be used directly with the coupled similarity based model. In this paper, we tested our approach with a large number of different classification applications. Our results suggest that the use of the coupled similarity provides significant advantages from an effectiveness perspective.

The future works are the follows: Firstly, this work only consider term frequency-inverse document frequency as the feature to build the vector presentation, it can be improve we involve more information such as semantic annotation for each term or position of the term in the document. Secondly, when we calculate the intra-coupled similarity, we equally treat every feature in the vector presentation; however, they are not homogeneous, we should find a better way to measure the relations of each feature, and it also useful to consider the same thing when deal with the relation from term to term and follow by document to document. Finally, in the longer plan, the optimization of the parameter should be taken into account, and some approximation also should be added because of the computational cost is high. Moreover, we will explore specific applications, which are built on top of the coupled term to term similarity in greater detail. Specifically, we will study the problems of similarity search, plagiarism detection, and its applications. We have already performed some initial work on

performing similarity search, when the target is a set of documents, rather than a single document. We will also study how text can be efficiently indexed and retrieved with the use of the coupled similarity.

References

1. Hinton, G.E., Sejnowski, T.J.: Learning and Relearning in Boltzmann Machines, vol. 1, pp. 282–317. MIT Press, Cambridge (1986)
2. Pollack, J.B.: Recursive distributed representations. Artif. Intell. **46**(1), 77–105 (1990)
3. Elman, J.L.: Distributed representations, simple recurrent networks, and grammatical structure. Mach. Learn. **7**(2–3), 195–225 (1991)
4. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. JASIS **41**(6), 391–407 (1990)
5. Cao, L., Gorodetsky, V., Mitkas, P.: Agent mining: the synergy of agents and data mining. IEEE Intell. Syst. **24**(3), 64–72 (2009)
6. Hofmann, T.: Unsupervised learning by probabilistic latent semantic analysis. Mach. Learn. **42**(1–2), 177–196 (2001)
7. Bellegarda, J.R.: Statistical language model adaptation: review and perspectives. Speech Commun. **42**(1), 93–108 (2004)
8. Bengio, Y., Schwenk, H., Senécal, J.-S., Morin, F., Gauvain, J.-L.: Neural probabilistic language models. In: Holmes, D.E., Jain, L.C. (eds.) Innovations in Machine Learning. STUDFUZZ, vol. 194, pp. 137–186. Springer, Heidelberg (2006)
9. Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., Khudanpur, S.: Recurrent neural network based language model. In: Proceedings of INTERSPEECH, pp. 1045–1048 (2010)
10. Mikolov, T., Kombrink, S., Burget, L., Cernocky, J.H., Khudanpur, S.: Extensions of recurrent neural network language model. In: Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5528–5531 (2011)
11. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
12. Hinton, G., Salakhutdinov, R.: Discovering binary codes for documents by learning deep generative models. Top. Cogn. Sci. **3**(1), 74–91 (2011)
13. Bordes, A., Glorot, X., Weston, J., Bengio, Y.: Joint learning of words and meaning representations for open-text semantic parsing. In: Proceedings of International Conference on Artificial Intelligence and Statistics, pp. 127–135 (2012)
14. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. J. Mach. Learn. Res. **12**, 2493–2537 (2011)
15. Turian, J., Ratinov, L., Bengio, Y.: Word representations: a simple and general method for semi-supervised learning. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 384–394 (2010)
16. Collobert, R., Weston, J.: A unified architecture for natural language processing: deep neural networks with multitask learning. In: Proceedings of the 25th International Conference on Machine Learning, pp. 160–167 (2008)
17. Mnih, A., Hinton, G.E.: A scalable hierarchical distributed language model. In: Proceedings of Advances in Neural Information Processing Systems, pp. 1081–1088 (2008)

18. Cao, L.: Non-iidness learning in behavioral and social data. Comput. J. (2013). doi:10.1093/comjnl/bxt084
19. Huang, E.H., Socher, R., Manning, C.D., Ng, A.Y.: Improving word representations via global context and multiple word prototypes. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, pp. 873–882 (2012)
20. Cao, L., Philip, S.Y. (eds.): Behavior Computing: Modeling, Analysis, Mining and Decision. Springer, Heidelberg (2012)
21. Wang, C., She, Z., Cao, L.: Coupled attribute analysis on numerical data. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence, pp. accepted (2013)
22. Wang, C., She, Z., Cao, L.: Coupled clustering ensemble: incorporating coupling relationships both between base clusterings and objects. In: Proceedings of the 29th IEEE International Conference on Data Engineering, pp. accepted (2013)
23. Cheng, X., Miao, D., Wang, C., Cao, L.: Coupled term-term relation analysis for document clustering. In: Proceedings of IJCNN 2013 (2013)

Learning the Hotness of Information Diffusions with Multi-dimensional Hawkes Processes

Yi Wei[1(✉)], Ke Zhou[2(✉)], Ya Zhang[1], and Hongyuan Zha[2]

[1] Shanghai Jiao Tong University, Dongchuan RD. 800, Shanghai 200240, China
{ywei1990,ya_zhang}@sjtu.edu.cn
[2] Georgia Institute of Technology, Atlanta, GA 30332, USA
{kzhou,zha}@cc.gatech.edu

Abstract. Modeling the information cascading process over networks has attracted a lot of research attention due to its wide applications in viral marketing, epidemiology and recommendation systems. In particular, information cascades can be useful for not only inferring the underlying structure of the network, but also providing insights on the properties of information itself. In this paper, we address the problem of jointly modeling the influence structure and the hotness of the information itself based on the temporal events describing the process of the information cascading. Specifically, we extend the multi-dimensional Hawkes process, which captures the mutual-excitation nature of information cascading, to further incorporate the hotness of the information being propagated. In the proposed method, the hotness of information and the network structure are modeled in a unified and principled manner, which enables them to reinforce each other and thus enhances the estimation of both. Experiments on both real and synthetic data show that our algorithm typically outperforms several existing methods and accurately estimates the hotness of information from the observed data.

Keywords: Information diffusions · Hawkes processes · Hotness

1 Introduction

Characterizing the influence network has been considered critical for a wide variety of applications such as viral marketing and epidemiology. Taking viral marketing as an example, in order to maximize the ROI (Return of Investment) of a marketing campaign, it is important to identify the most influential users to target. In recent years, fueled by the rapid growth of online services, particularly the microblog services, the influence network formed by online users becomes increasingly attractive as a unique platform to study patterns of information diffusion and decision making of online users [2–5]. Microblog services allow users to follow each other and share information through tweeting and retweeting. With microblog services becoming increasingly popular in recent years, a large volume of messages are generated and propagate via microblog services everyday. The content of messages and the influence of authors are generally

L. Cao et al. (Eds.): ADMI 2013, LNAI 8316, pp. 92–110, 2014.
DOI: 10.1007/978-3-642-55192-5_8,

considered as two major factors that impact the dissemination of information in microblogs. Previous studies have revealed that the content [7–9] and the sentiment [10,11] of messages play important roles in the propagation. Several other studies focus on scoring the user's influence by analyzing message cascades or user relationships [12,13].

Unfortunately, the influence network is hidden and difficult to observe in general. In fact, in many of the applications mentioned above, only the cascading propagation processes for a diverse set of information are observable. For example, in e-commerce web sites such as Amazon and Ebay, one can only observe the timestamps of each user's purchasing behaviors over a wide range of products, i.e., one knows when a user purchases a particular product. However, it is difficult or even impossible to know whether the user take his/her own initiatives to purchase the product or whether the user is influenced by a certain user for the purchasing decision. As a result, it is important and desirable to estimate the underlying hidden influence network based on the observed information cascading processes, e.g., the collection of time-stamped user purchasing histories [4].

One critical issue that has been largely ignored in influence network analysis is the important roles played by the intrinsic characteristics of information in the diffusion of information [14]. In the above example, the buyers of different products have quite distinct purchasing behaviors, which can lead to quite different observed patterns. Specifically, the purchasing behaviors of a limited-time sale product are very different from the behaviors of purchasing luxury furniture. Ignoring the intrinsic characteristics of the information could lead to poor estimations of the influence network. As a result, it is important to simultaneously model the underlying influence network based on the observed temporal patterns of information cascading and take into account the intrinsic characteristics of the information.

Another important question is whether we should model information diffusion in a *multiple-infection* manner or a *single-infection* manner. Previous studies have proposed different models to recover the hidden network structure under the assumption that nodes can be only infected once in each cascade [2–4]. However, this assumption does not seem proper in many real-world settings. In the example of e-commerce websites, each user can purchase one particular product or products under different brands multiple times. In mircoblog websites, each user can post/repost a particular message or messages discussing the same topic multiple times. There are many other examples of *multiple-infection* in real-world diffusion networks. Truncating the *multiple-infection* data into *single-infection* data may lead to inaccurate results. Therefore, in this paper, we use a point process, which can deal with the *multiple-infection* data nicely, to model information diffusion.

In this paper, we propose a continuous-time model that can capture both the mutual-excitation property of the information diffusion and the intrinsic characteristics of information in a unified and principled manner. The proposed model extends the multi-dimensional Hawkes processes by introducing a set of

variables representing the popularity of the information (called '*hotness*' thereafter), capturing to some extend the intrinsic characteristics of the information being propagated. Both the influence network and the hotness of information can be estimated by the maximal likelihood estimator. We propose an efficient algorithm to optimize the likelihood function by maximizing a tight lower bound of the likelihood function at each iteration. We experiment with both synthetic and real networks to validate the effectiveness of the proposed methods in comparison with several other existing methods.

The rest of the paper is organized as follows. Related work is reviewed in Sect. 2. In Sect. 3, we present the Multi-Dimensional Hawkes Process Model, a continuous-time model for influence, and the corresponding parameter-estimation problem. In Sect. 4, we extend the multi-dimensional Hawkes process to further incorporate the hotness of the information and describe the EM algorithm for solving this problem. We present experiments with synthetic and real networks in Sect. 5 and conclude in Sect. 6.

2 Related Work

Relevant to the area of agent mining [6], many research works in social-network analysis have been devoted to study different factors that affect information diffusions.

Early research works have shown that the process of diffusion is largely affected by social influence and homophily among people. For example, Leskovec et al. [15] analyzed how users are influenced by their neighbors' recommendations in a recommendation network. Crandall et al. [16] focused on the interplay between similarity and social ties and found clear feedback effects between the two factors. Bakshy et al. [17] proposed a method to quantify the social influence and find the cost-effective marketing strategy.

However, the above studies have not taken the content of information into account. A number of other related works have shown that the content of information plays an important role in information diffusions. Gruhl et al. [18] characterize and model the information diffusions among the blog network at both topic level and individual level. Leskovec et al. [19] tracked topics or distinctive phrases that travel in the news media. Romero et al. [20] found that different topics exhibit different mechanics of information diffusions by studying the spread of Twitter hashtags. Agrawal et al. [14] proposed a model that caputre the nature of information items by two parameters, the tendency to spread and the tendency to be received. There have been a number of studies, which are close to ours, on the temporal patterns of information diffusion.

Then we will review some works of inferring the underlying network structure which use generative probabilistic models. NetInf [2] infers the network connectivity using submodular optimization by considering only the most probable directed tree. NetInf assumes that given a tree pattern $T(V_T, E_T)$, the cascade transmits over edge $(u, v) \in V_T$ with probability β and stops with probability $(1 - \beta)$, where E_T is edge set and V_T is the vertex set of tree T. The likelihood

of a cascades on a given tree T can be written as:

$$P(c|T) = \prod_{(u,v)\in E_T} \beta P_c(u,v) \prod_{u\in V_T,(u,x)\in E\setminus E_T} (1-\beta),$$

where $P_c(u,v)$ is the probability of cascade c transmitting trough edge (u,v). They proved that maximizing the above function is NP-hard and proposed a greedy algorithm to obtain at least a constant fraction of $(1-1/e)$ of the optimal value achievable using k edges. In the following work MultiTree [21], they consider all possible directed trees in which a diffusion process spreading over the network can create the cascade and surprisingly they show that the running time of MultiTree and NetInf are similar. ConNIe [3] and NetRate [4] infer not only the network connectivities but also prior probabilities of infection or transmission rates of infection using convex optimization. NetRate is based on a continuous-time transmission model. The likelihood of a cascade can be written as:

$$f(\mathbf{t};\mathbf{A}) = \prod_{t_i\le T}\prod_{t_m>T} S(T|t_i;\alpha_{i,m}) \times \prod_{k:t_k<t_i} S(t_i|t_k;\alpha_{k,i}) \sum_{j:t_j<t_i} H(t_i|t_j;\alpha_{j,i}),$$

where S is the survival function representing the probability of survival, H is the harzard function representing the conditional probability of infection, α is the transmission rate of the edge, t_i, t_j, t_k are the coresponding timestamps of node i, j, k, and T is the observation time window. They proved that the network inference problem based on the above function is convex and solved the problem using a convex opitmization toolbox. Based on the framework of NetRate, Du et al. [22] propose a kernel-based method which can capture a diverse range of different types of influence without any prior assumption. In InfoPath [23], the authors extend the model of NetRate to infer dynamic networks which change over time and develop an algorithm based on projected stochastic gradient method to solve the problem. Du et al. [24] propose a continuous-time model to infer topic dependent transmission rates in topic-sensitive information diffusion networks following the work of NetRate. Gomez-Rodriguez et al. [25] first introduce a general additive risk model under which the hazard rate of each node is an additive function of the infection times of other previously infected nodes and show that previous works ([4,22,23]) are particular cases of their model. Then they develop a multiplicative risk model under which the hazard rate of each node is multiplicative on the infection times of other previously infected nodes which allows previously infected nodes to either increase or decrease the risk of another node getting infected.

Another line of the research focused on exploiting the application of point processes to solve the network inference problem. Stomakhin et al. [26] developed a model based on the Hawkes process to predict the unknown participants in a portion of gang rivalry events. Iwata et al. [27] employ multiple inhomogeneous Poisson processes, which share parameters, such as influence for each user and relations between users to model adoption of multiple items in online

communities. Zhou et al. [28] focus on the nonparametric learning of the triggering kernels of multi-dimensional Hawkes processes in social network analysis and propose an algorithm MMEL to solve it. Zhou et al. [29] use nuclear and l_1 norm regularization in parameter estimation procedure of multi-dimensional Hawkes processes to take into account the prior knowledge of the networks.

3 Multi-dimensional Hawkes Process Model

In this section, we briefly introduce the multi-dimensional Hawkes process model that were first proposed in [30]. For clarity, the frequently used notations that are used in rest of the paper are listed in Table 1. Suppose that the cascades are observed over a U-node hidden directed network G. The observed data consist of a set of cascades $\mathbf{D} := \{c_1, \ldots, c_{|\mathbf{D}|})\}$, where $c_d = \{(t_i^c, u_i^c)\}_{i=1}^{n_c}$ is a sequence of events observed during a time segment of $[0, T]$. Each pair (t_i^c, u_i^c) represents an event occurring at the node u_i^c at time t_i^c. For example, the user u_i purchases the product c at the time t_i^c. Figure 1 provides an example of the Hawkes process model. Based on individual nodes's transaction records, we may construct the transmission cascades, based on which we try to infer the latent directed influence network. The self-pointed circle stands for the self-exciting coefficient and the arrows between nodes represent the mutually-exciting coefficient.

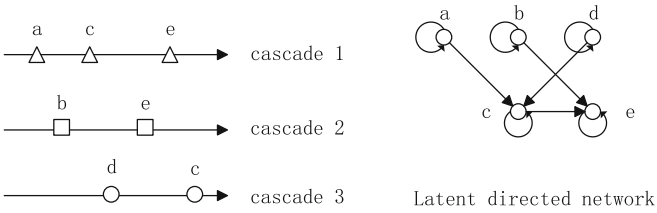


Fig. 1. Different cascades may be extracted from the timeline (left). In the latent directed network we attempt to infer, the self-pointed circle stands for the self-exciting coefficient μ and the arrows between nodes represent the mutually-exciting coefficient $a_{uu'}$(right).

Table 1. Frequently used notations.

Symbol	Meaning
c	Cascade, a series of infection timestamps
u	Node u
n_c	Length of cascade c
$\lambda_u(t)$	Conditional intensity function of node u
μ_u	Base intensity of node u
$a_{uu'}$	Degree of influence from node u' to node u
$g_{uu'}(\cdot)$	Kernel function of edge $u' \rightarrow u$
$w_{uu'}$	Time decay factor of edge $u' \rightarrow u$
b_c	Hotness of cascade c

Formally, the multi-dimensional Hawkes process is defined by a U-dimensional point process $N_t^u, u = 1, \ldots, U$, with the conditional intensity for the u-th dimension expressed as follows:

$$\lambda_u(t) = \mu_u + \sum_{(t',u'):t'<t} a_{uu'} g(t - t') \tag{1}$$

where $\mu_u \geq 0$ is the base intensity for the u-th Hawkes process. The coefficient μ_u captures the background intensity of each user's activity. For example, in the a microblog site, with larger value of μ_u, user u is more likely to spread a message without any outside triggers. The coefficient $a_{uu'} \geq 0$ captures the degree of influence of events occurred in the u'-th dimension to the u-th dimension. In the above example, larger value of $a_{uu'}$ indicates that messages from user u' are more likely to spread to user the u in the future. Function g is the triggering kernel. In our model, we define $g(t - t') = w \exp(-w(t - t'))$, where w is a tunable time decay factor. Thus function (1) can be written as:

$$\lambda_u(t) = \mu_u + \sum_{(t',u'):t'<t} a_{uu'} w \exp(-w(t - t')) \tag{2}$$

We collect the parameters into matrix-vector forms, $\boldsymbol{\mu} = (\mu_u)$ for the base intensities and $\mathbf{A} = (a_{uu'})$ for the self-exciting and mutually-exciting coefficients. We use $\mathbf{A} \geq 0$ and $\boldsymbol{\mu} \geq 0$ to indicate that we require the matrices to be entry-wise nonnegative.

Given the set of cascades that are sampled from the multi-dimensional Hawkes process $\mathbf{D} := \{c_1, \ldots, c_{|\mathbf{D}|})\}$. The log-likelihood function of the parameters $\Theta = \{\boldsymbol{\mu}, \mathbf{A}\}$ can be expressed as follows.

$$\mathcal{L}(\Theta) = \sum_c \left(\sum_{i=1}^{n} \log \lambda^c_{u_i^c}(t_i^c) - \sum_{u=1}^{U} \int_0^T \lambda_u^c(t) dt \right), \tag{3}$$

Thus, the inference problem is:

$$\begin{aligned} \underset{\Theta}{\text{maximize}} \quad & \mathcal{L}(\Theta) \\ \text{subject to} \quad & a_{ij} \geq 0,\ i, j = 1, \ldots, U, \\ & \mu_u \geq 0,\ u = 1, \ldots, U. \end{aligned} \tag{4}$$

By maximizing the likelihood function, we can obtain the estimates of the parameters $\boldsymbol{\mu}$ and $\mathbf{A}$.

4 Incorporate the Hotness of Information

In this paper, we argue that the intrinsic characteristics should be taken into account when modeling the information cascading process in social networks. In particular, we focus on modeling the hotness, which is meant to capture the *popularity* of the information itself. In real-world applications, the hotness of information plays an important role in the dynamics of information cascading, it can influence the speed at which the information propagates [14,24].

4.1 The MultiHawkes Model

Formally, we associate a positive real number b_c to each cascade c to represent its hotness. The goal is to model the hotness $\{b_c\}$ and the social influence jointly in a unified framework. Specifically, the intensity function for the cascade c with the hotness b_c is formulated as:

$$\lambda_u^c(t) = \mu_u + b_c \sum_{(t',u'):t'<t} a_{uu'} w \exp(-w(t-t')). \tag{5}$$

Intuitively, larger value of b_c leads to generally larger intensity function for every node u, indicating that the information cascade c is popular and will be participated by more users more frequently.

With the above definition of the intensity function, the log-likelihood of observed cascades can be expressed as follows:

$$\begin{aligned}
\mathcal{L}_{\text{hotness}}(\Theta) &= \sum_c \left(\sum_{i=1}^{n} \log \lambda_{u_i^c}^c(t_i^c) - \sum_{u=1}^{U} \int_0^T \lambda_u^c(t) dt \right) \\
&= \sum_c \left(\sum_{i=1}^{n_c} \log \lambda_{u_i^c}^c(t_i^c) - b_c \left(T \sum_u \mu_u + \sum_{u=1}^{U} \sum_{j=1}^{n_c} a_{uu_j^c} G(T - t_j^c) \right) \right) \\
&= \sum_c \left(\sum_{i=1}^{n_c} \log \lambda_{u_i^c}^c(t_i^c) - b_c \left(T \sum_u \mu_u + \right. \right. \\
&\left. \left. \sum_{u=1}^{U} \sum_{j=1}^{n_c} a_{uu_j^c} (1 - \exp(-w(T - t_j^c))) \right) \right),
\end{aligned} \tag{6}$$

where $G(t) = \int_0^t g(s) ds$.

We can obtain the estimations of the hotness b_c together with $\mathbf{A}$ and $\boldsymbol{\mu}$ by maximizing the likelihood function:

$$\begin{aligned}
\underset{b_c, \boldsymbol{\mu}, \mathbf{A}, \mathbf{W}}{\text{maximize}} \quad & \mathcal{L}_{\text{hotness}}(\Theta) \\
\text{subject to} \quad & a_{ij} \geq 0, \quad i, j = 1, \ldots, U, \\
& \mu_u \geq 0, \quad u = 1, \ldots, U, \\
& b_c > 0, \quad c = 1, \ldots, |\mathbf{D}|.
\end{aligned} \tag{7}$$

It turns out that the above objective function can be optimized efficiently by the MM algorithm [31] in a iterative manner. In particular, we construct a lower-bound $Q(\Theta|\Theta^{(t)})$ at current estimation $\Theta^{(t)}$ as follows:

$$\begin{aligned}
Q(\Theta|\Theta^{(t)}) = \sum_c \left(\sum_{i=1}^{n_c} \left(p_{ii}^c \log \frac{\mu_{u_i^c}}{p_{ii}^c} + \sum_{j=1}^{i-1} p_{ij}^c \log \frac{b_c a_{u_i^c u_j^c} g(t_i^c - t_j^c)}{p_{ij}^c} \right) \right. \\
\left. - \left(\sum_u \mu_u T + \sum_{u=1}^{U} \sum_{j=1}^{n_c} (b_c^2 \frac{a_{uu_j^c}^{(t)}}{2b_c^{(t)}} + a_{uu_j^c}^2 \frac{b_c^{(t)}}{2a_{uu_j^c}^{(t)}}) (G(T - t_j^c) - G(0)) \right) \right),
\end{aligned} \tag{8}$$

where p_{ij}^c and p_{ii}^c is defined as follows:

$$p_{ij}^c = \frac{b_c a_{u_i^c u_j^c}^{(t)} g(t_i^c - t_j^c)}{\mu_{u_i^c}^{(t)} + b_c \sum_{j=1}^{i-1} a_{u_i^c u_j^c}^{(t)} g(t_i^c - t_j^c)}, j = 1, \cdots, i-1 \tag{9}$$

$$p_{ii}^c = \frac{\mu_{u_i^c}^{(t)}}{\mu_{u_i^c}^{(t)} + b_c \sum_{j=1}^{i-1} a_{u_i^c u_j^c}^{(t)} g(t_i^c - t_j^c)} \tag{10}$$

The p_{ij}^c and p_{ii}^c have the nice interpretations that reveal the influence structures for the information cascade c. Specifically, p_{ij}^c represents the probability that the i-th event in cascades c is triggered by the j-th event, while p_{ii}^c represents the probability that the i-th event is sampled from the background intensity.

The following two properties hold for $Q(\Theta|\Theta^{(t)})$:

$$\mathcal{L}_{\text{hotness}}(\Theta) \geq Q(\Theta|\Theta^{(t)}), \quad \forall \Theta \tag{11}$$

$$\mathcal{L}_{\text{hotness}}(\Theta^{(t)}) = Q(\Theta^{(t)}|\Theta^{(t)}) \tag{12}$$

Moreover, let $\Theta^{(t+1)} = \max_{\Theta} Q(\Theta|\Theta^{(t)})$, we have

$$\mathcal{L}_{\text{hotness}}(\Theta^{(t+1)}) \geq Q(\Theta^{(t+1)}|\Theta^{(t)}) \tag{13}$$

$$\geq Q(\Theta^{(t)}|\Theta^{(t)}) = \mathcal{L}_{\text{hotness}}(\Theta^{(t)}), \tag{14}$$

which shows that $\mathcal{L}_{\text{hotness}}$ increases monotonically during the iterations and it can be shown that the iterates converges to the local optimal $\mathcal{L}_{\text{hotness}}$ [31].

Another advantage of optimizing $Q(\Theta|\Theta^{(t)})$ is that all variables $\{b_c\}$, $\boldsymbol{\mu}$ and $\mathbf{A}$ can be optimized independently with closed-form solution and the non-negativity constraints are naturally taken care of.

Optimizing with respect to μ_u. Let $\frac{\partial Q}{\partial \mu_u} = 0$, we have

$$\sum_c \sum_{u_i^c = u} \frac{p_{ii}^c}{\mu_i^c} - CT = 0 \tag{15}$$

the following update equation for μ_u:

$$\mu_u = \frac{\sum_c \sum_{u_i^c = u} p_{ii}^c}{CT} \tag{16}$$

Optimizing with respect to $\mathbf{a_{uu'}}$. Let $\frac{\partial Q}{\partial a_{uu'}} = 0$, we have

$$\frac{\sum_c \sum_{i:u_i^c = u} \sum_{j:j<i, u_j^c = u'} p_{ij}^c}{a_{uu'}} \tag{17}$$

$$- \sum_c \sum_{j:u_j^c = u', j < n_c} a_{uu'} \frac{b_c^{(t)}}{a_{uu'}^{(t)}} G(T - t_j) = 0 \tag{18}$$

Therefore, we can update $a_{uu'}$ as follow:

$$a_{uu'} = \sqrt{\frac{\sum_c \sum_{i:u_i^c=u} \sum_{j:j<i,u_j^c=u'} p_{ij}^c}{\sum_c \sum_{j:u_j^c=u',j<n_c} b_c^{(t)}(G(T-t_{u_j^c}^c)-G(0))} a_{uu'}^{(t)}} \tag{19}$$

Optimizing with respect to $\mathbf{b_c}$. Let $\frac{\partial Q}{\partial b_c} = 0$, we have

$$\frac{\sum_{i=1}^{n_c}\sum_{j=1}^{i-1} p_{ij}^c}{b_c} - \sum_{u=1}^{U} \sum_{j:t_{u_j^c}^c < t_u^c} b_c a_{uu_j^c}^{(t)}(G(T-t_{u_j^c}^c)-G(0)) = 0 \tag{20}$$

Therefore, we can update b_c as follow:

$$b_c = \sqrt{\frac{\sum_{i=1}^{n_c}\sum_{j=1}^{i-1} p_{ij}^c}{\sum_{u=1}^{U}\sum_{j:t_{u_j^c}^c<t_u^c} a_{uu_j^c}^{(t)}(G(T-t_{u_j^c}^c)-G(0))} b_c^{(t)}} \tag{21}$$

We hereafter denote the above proposed method as MultiHawkes.

5 Experiments

We experiment with both synthetic and real-world data sets to evaluate the proposed method in comparison with a few other existing methods.

5.1 Experiments on Synthetic Data

To validate the proposed algorithm, we first experiment with the simulation data. A two-step process is employed to generated the event cascades: 1. Generate an underlying network based on the Kronecker Graph model; 2. Generate the event cascades with the Hawkes process based on the network. We experiment with various parameter settings in order to understand the performance of the proposed algorithms under different settings.

Experiment Setup. The underlying network is generated with the Kronecker Graph model [32], which has been proved to accurately mimic the properties of real influence networks. We choose three types of Kronecker Graphs which have very different structures: random [33], hierarchical [34], and core-periphery [35]. In the cascade-generation step, based on the generated networks, multi-dimensional Hawkes processes incorporating the hotness of information are used to generate a set of event cascades under different parameter settings.

We compare the performance of the proposed method with several other methods on the generated synthetic data. Because NetRate is not originally designed to handle the multi-infection cascades, for the fairness of the comparison, we also compare MultiHawkes(SI), a truncated model of MultiHawkes, with NetRate on the synthetic data generated with NetRate model.

Evaluation Metrics. Three metrics are employed to measure the performance of influence network reconstruction: *precision*, *recall* and *accuracy*. Suppose a^* and $\hat{a}$ are the true and inferred triggering factor, respectively. The precision, recall, and accuracy are defined as follows:

$$\text{precision} = \frac{\sum_{i,j}(I(a^*_{ij}) = 1 \text{ AND } I(\hat{a}_{ij}) = 1)}{\sum_{i,j} I(\hat{a}_{ij})}$$

$$\text{recall} = \frac{\sum_{i,j}(I(a^*_{ij}) = 1 \text{ AND } I(\hat{a}_{ij}) = 1)}{\sum_{i,j} I(a^*_{ij})}$$

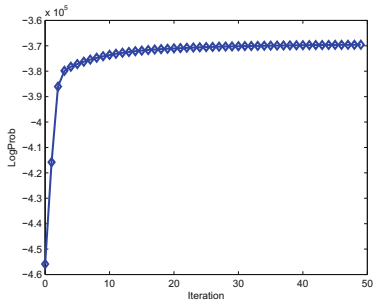
$$\text{accuracy} = 1 - \frac{\sum_{i,j} |I(a^*_{ij}) - I(\hat{a}_{ij})|}{\sum_{i,j}(I(a^*_{ij}) + I(\hat{a}_{ij}))},$$

where $I(a) = 1$ if $a > \tau$ and $I(a) = 0$ otherwise, τ is a threshold to judge the existence of an edge. By scanning different values of τ, we generate the precision-recall graph to compare the performance of different methods.

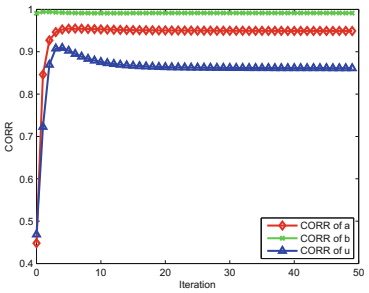
One needs to note that the absolute values of the variables inferred by different methods may not be directly comparable on the same numerical scale. Therefore, we also evaluate accuracy of the prediction by computing Pearson's correlation (CORR) between the inferred values and the true values.

Analysis of Convergence. We first verify the convergence of the proposed algorithm using the synthetic data. The parameters to generate the synthetic data is set as follows: the background rate $\mu \sim \text{uniform}(0.001, 0.005)$, the time window $T = 10$, the number of cascades $|\mathbf{D}| = 2000$, the number of nodes $U = 1024$, the triggering factors and hotness factors are sampled uniformly $a \sim \text{uniform}(0.01, 1), b \sim \text{uniform}(0, 1)$, and the kernel function factor $w = 10$. We fit the MultiHawkes model using the above generated data and plot the log-likelihood of the cascades against iterations in Fig. 2. We also plot the Pearson's correlation (CORR) for a, b and μ against iterations in Fig. 2. The plots clearly show that the log-likelihood of cascades does monotonically increase at each iteration, and the CORR of a, b and μ also converge after a certain number of iterations.

The Effect of Kernel Function Factor w. To investigate the effect of the kernel function factor w, we generate three data sets on the same network with $w = 1$, $w = 10$, and $w = 100$ respectively. We then fit the proposed MultiHawkes model with different values of the kernel function factor w ($w = 1$, $w = 10$, and $w = 100$) on all data sets and summarize the corresponding results in Fig. 3. As can be seen from the figure, the kernel function factor w play an important role in the model fitting and does impact the precision, recall, and accuracy of the prediction. For the synthetic data generated with $w = 10$ and $w = 100$, applying the proposed algorithm with $w = 10$ and $w = 100$ achieves the best results respectively. For the synthetic data generated with $w = 1$, the proposed algorithm with $w = 1$ performs best in estimating a, but the best estimation of b and μ is obtained with $w = 10$.



(a) Log-likelihood of cascades against iterations



(b) CORR of a, b and μ against iterations

Fig. 2. The convergence of Log-likelihood, a, b, and μ.

Results and Analysis. We compare the proposed method with two other state-of-the-art methods: NetInf and NetRate, which have been briefly introduced in Sect. 2. Please note that both NetInf and NetRate assume that once a node is infected, the node maintains infected thereafter. One major difference between the proposed model and the two models is that the proposed model allows each node to be infected repeatedly. For the fairness of the comparison, in the experiments of NetInf and NetRate, we only record the first infected time of a node in each cascade if it is infected more than once.

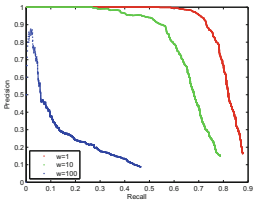
Table 2 compares the Pearson's correlation of parameters a, b and μ of MultiHawkes and NetRate for 3 kinds of synthetic networks (Random, Hierarchical and Core-periphery) with $w = 1, 10, 100$. Note that NetInf does not provide a numerical estimation of parameters and NetRate only estimate parameter a numerically, so only the corresponding number are listed. Table 2 shows that, MultiHawkes outperforms in the estimation of parameter a than NetRate and also provide a accurate estimation of parameter b on all 3 kinds of networks.

Figure 4 compares the precision, recall and accuracy of MultiHawkes, NetRate and NetInf on three different types of Kronecker networks. Figure 4 shows that in terms of precision, recall and accuracy, MultiHawkes outperforms NetRate and NetInf for all the synthetic networks, especially in the Hierarchical and Core-periphery networks.

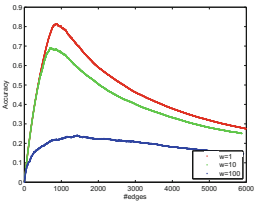
5.2 Experiments on Real Data

Dataset Description. We crawled the Sina Weibo[1] data of about 250,000 users until May. 2012, including the follower/followee relationship and all the tweets each user posts, from which we extract the information propagation history. We start with the most active user (according to the number of his posts and reposts), and we traverse the follow relationship graph forward and backward

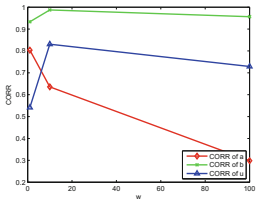
[1] http://weibo.com/, which is the largest microblog service in China.



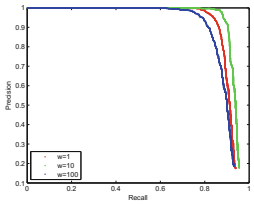
(a) Precision-recall of $w = 1$ dataset



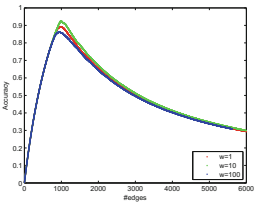
(b) Accuracy of $w = 1$ dataset



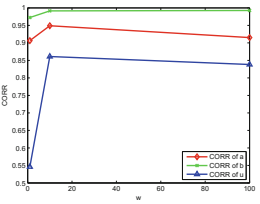
(c) CORR of a, b and μ of $w = 1$ dataset



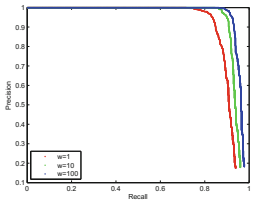
(d) Precision-recall of $w = 10$ dataset



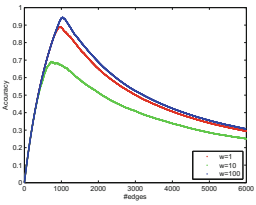
(e) Accuracy of $w = 10$ dataset



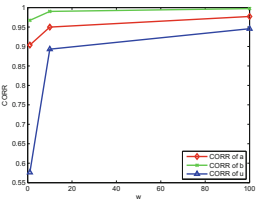
(f) CORR of a, b and μ of $w = 10$ dataset



(g) Precision-recall of $w = 100$ dataset



(h) Accuracy of $w = 100$ dataset



(i) CORR of a, b and μ of $w = 100$ dataset

Fig. 3. Comparion of applying our algorithm with $w = 1$, $w = 10$ and $w = 100$ on both $w = 1$ and $w = 100$ datasets.

to retrieve the network. Then we finally get a 987-node network with 13808 edges and 58749 cascades within a observation interval $T = 480$ h. We plot the distribution graph of in-degree, out degree and length of cascades on the entire Sina Weibo dataset and the dataset we sampled. From Fig. 5, we can see that the dataset we sampled are generally consistent with the entire Sina Weibo dataset in terms of the distribution of node degree and length of cascades. Due to the restriction of sina weibo that the a non VIP user can follow 2000 users at most, the in-degree distribution shown in Fig. 5 has a spike at in-degree = 2000. We tune the kernel function factor $w = 1000$ in the above setting.

Processing Real Data. In many real-world information networks, like news media networks or microblog networks, each topic or message may generate lots

Table 2. Comparison of MultiHawkes against NetRate in terms of CORR of a, b and μ. Three types of underlying networks (random, hierarchical, and core-periphery) are employed, with time window $T = 10$ and 1024 nodes. In total 5000 cascades are generated.

Network	w	Algorithm	CORR of a	CORR of b	CORR of μ
Random	1	MultiHawkes	0.803678	0.933289	0.541471
		NetRate	0.721950	-	-
Random	10	MultiHawkes	0.949057	0.991408	0.861143
		NetRate	0.859655	-	-
Random	100	MultiHawkes	0.977149	0.997344	0.945641
		NetRate	0.853932	-	-
Hierarchical	1	MultiHawkes	0.737367	0.751655	0.475071
		NetRate	0.642023	-	-
Hierarchical	1	MultiHawkes	0.883611	0.965733	0.861498
		NetRate	0.732625	-	-
Hierarchical	100	MultiHawkes	0.969883	0.986070	0.949475
		NetRate	0.799974	-	-
Core-periphery	1	MultiHawkes	0.775754	0.925108	0.555683
		NetRate	0.719666	-	-
Core-periphery	10	MultiHawkes	0.922608	0.968046	0.88543
		NetRate	0.797916	-	-
Core-periphery	100	MultiHawkes	0.969366	0.995371	0.944173
		NetRate	0.811800	-	-

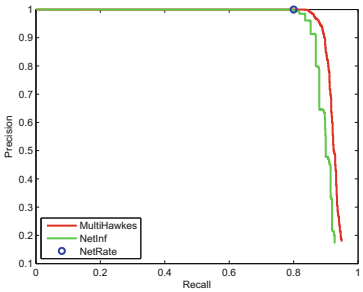
of variants of cascades, which have strong connections to each other. This type of connections is helpful for estimating the hotness of cascades.

In this paper, we take advantage of the hashtags in the microblog data to establish connections between cascades as shown in Fig. 7. We simply assume that cascades with the same hashtag have the same value of hotness. If the cascades $\{c_1, \cdots, c_k\}$ have the same hashtag, we update $b_{c_1}, \cdots, b_{c_k}$ as follow:

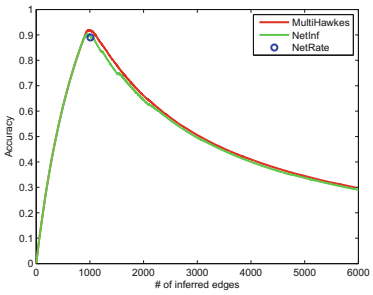
$$b_{c_1} = \cdots = b_{c_k} = b_c = \sqrt{\frac{\sum_{l=1}^{k} \sum_{i=1}^{n_{c^l}} \sum_{j=1}^{i-1} p_{ij}^{c^l}}{\sum_{l=1}^{k} \sum_{u=1}^{U} \sum_{j:t_{u_j^{c^l}}^{c^l} < t_u^{c^l}} a_{uu_j^{c^l}}^{(t)} \left(G(T - t_{u_j^{c^l}}^{c^l}) - G(0)\right)} b_c^{(t)}}$$

We denote the model using the above extension as MultiHawkes(MC).

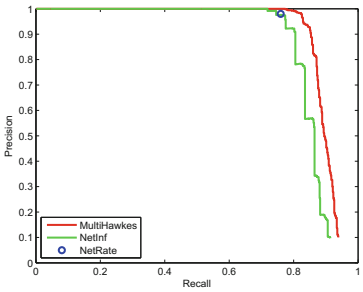
Results and Analysis. We empirically set the kernel function factor w to 1000 for the experiment on the Sina Weibo data set. Figure 8 shows that the proposed method typically outperforms NetRate in terms of precision, recall, and accuracy. NetInf has a better precision when recall is low. But when recall increases, the proposed method has a higher precision. In terms of accuracy, the



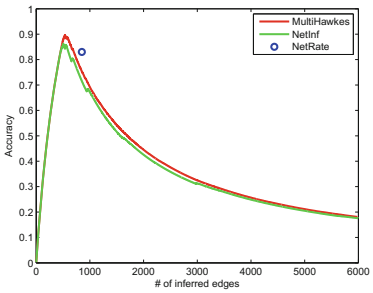
(a) Precision-recall of Random network



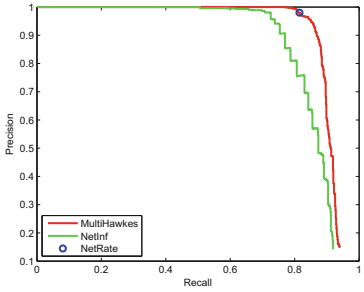
(b) Accuracy of Random network



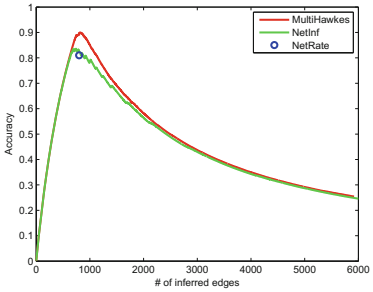
(c) Precision-recall of Hirachichal network



(d) Accuracy of Hirachichal network

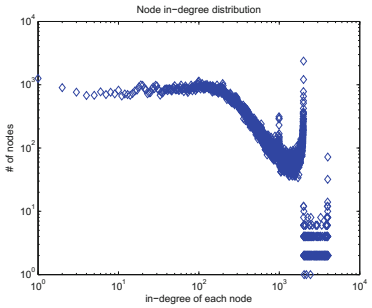


(e) Precision-recall of Core-periphery network

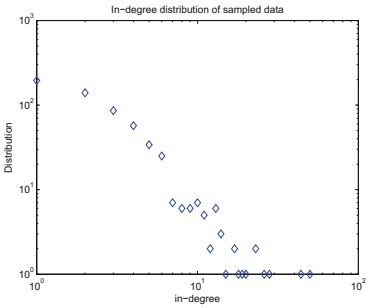


(f) Accuracy of Core-periphery network

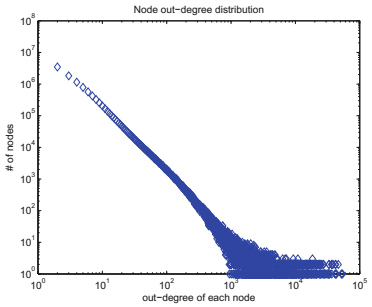
Fig. 4. The precision-recall graphs and accuracy graphs of MultiHawkes compared with NetRate and NetInf for 3 kinds of node networks (random, hierarchical and core-periphery) with time window $T = 10$, 1024 nodes and 5000 cascades.



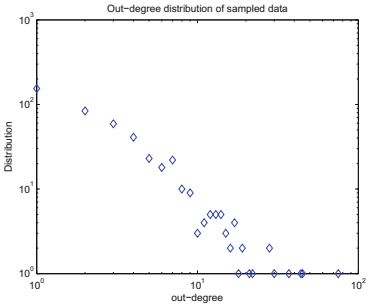
(a) in-degree distribution of the entire dataset



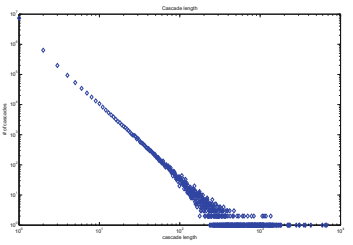
(b) in-degree distribution of the sampled dataset



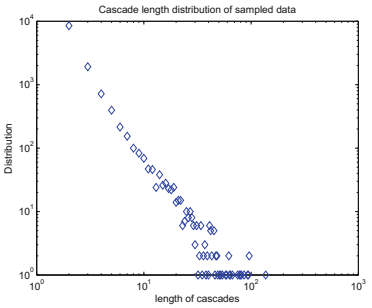
(c) out-degree distribution of the entire dataset



(d) out-degree distribution of the sampled dataset

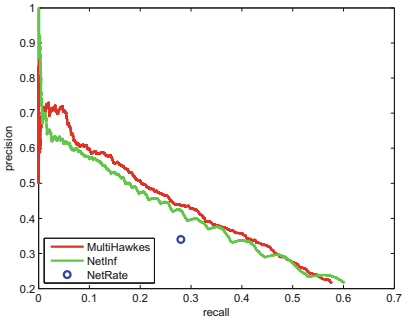


(e) cascade length distribution of the entire dataset

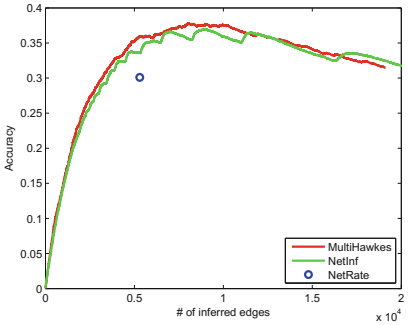


(f) cascade length distribution of the sampled dataset

Fig. 5. The distribution of in-degree, out degree and length of cascades on the entire Sina Weibo dataset and the dataset we sampled.



(a) Precision-recall of sampled Sina Weibo dataset



(b) Accuracy of sampled Sina Weibo dataset

Fig. 6. The precision-recall graphs and accuracy graphs of MULTIHAWKES compared with NETRATE and NETINF on sampled Sina Weibo dataset with time window T=480 h, 987 nodes and 58749 cascades.

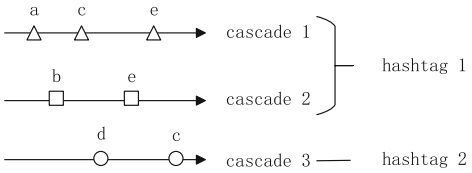
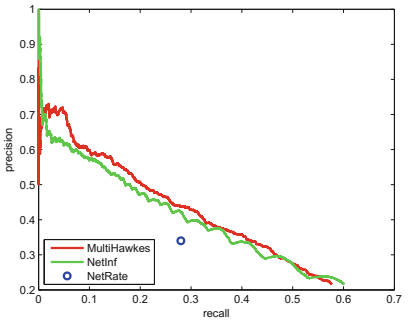
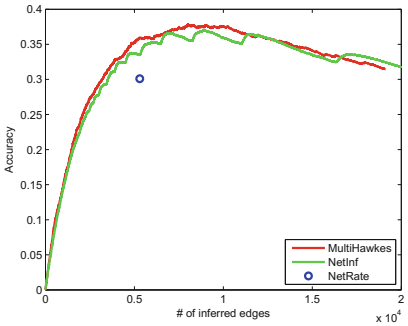


Fig. 7. We take advantage of the hashtags in the microblog data to build connections between cascades.



(a) Precision-recall of sampled Sina Weibo dataset



(b) Accuracy of sampled Sina Weibo dataset

Fig. 8. The precision-recall graphs and accuracy graphs of MULTIHAWKES compared with NETRATE and NETINF on sampled Sina Weibo dataset with time window T = 480 h, 987 nodes and 58749 cascades.

best accuracy point of NetInf is higher than our method. However, our method achieves a relatively stable value of accuracy while NetInf achieves a better accuracy in a relatively narrow region.

6 Conclusions and Future Work

We have presented a fully continuous-time diffusion network model in a new perspective by modeling the temporal pattern of the information cascading process together with the intrinsic characteristics of the information. In particular, the proposed model not only considers the impact of the connectivity of the edges in a network but also takes the popularity of information into account, which enables them to reinforce each other and thus enhances the estimation of both variables. The proposed model is fitted by optimizing the likelihood function in an iterative manner. When experimented with the synthetic data and real-world data, the proposed method typically outperforms NetRate and NetInf, two of the state-of-the-arts methods, in terms of precision, recall, accuracy, and Pearson's correlation. For the real-world social network data, we can discover latent influence relationships and infer the hotness of each cascade based on the timestamps of cascades.

There are several interesting directions in the future work. A more efficient algorithm may be explored to solve the inference problem. Moreover, the context of posts may be used to infer the hotness of a cascade. Our model may also be extended to time-varying model to discover the growth of network structures and the topic trends in social networks.

Acknowledgments. This research was supported by National Natural Science Foundation of China (No. 61003107 and No. 61129001) and the High Technology Research and Development Program of China (No. 2012AA011702).

References

1. Hinton, G., Salakhutdinov, R.: Discovering binary codes for documents by learning deep generative models. Top. Cogn. Sci. **3**(1), 74–91 (2011)
2. Rodriguez, M.G., Leskovec, J., Krause, A.: Inferring networks of diffusion and influence. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, no. 10, pp. 1019–1028 (2010)
3. Myers, S., Leskovec, J.: On the Convexity of Latent Social Network Inference. In: Collection of Advances in Neural Information Processing Systems 23, 1741–1749 (2010)
4. Rodriguez, M.G., Balduzzi, D., Schölkopf, B.: Uncovering the temporal dynamics of diffusion networks. In: Proceedings of ICML, pp. 561–568 (2011)
5. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 137–146 (2003)
6. Cao, L., Gorodetsky, V., Mitkas, P.: Agent mining: the synergy of agents and data mining. IEEE Intell. Syst. **24**(3), 64–72 (2009)

7. Cheong, M., Lee, V.: A study on detecting patterns in twitter intra-topic user and message clustering. In: The 2010 20th International Conference on Pattern Recognition, pp. 3125–3128 (2010)
8. Yanxiang, H., Wen, S., Ye, T., Qiang, C., Lu, L.: Summarizing microblogs on network hot topics. In: The 2011 International Conference on Internet Technology and Applications, pp. 1–4 (2011)
9. Zhang, D., Liu, Y., Lawrence, R.D., Chenthamarakshan, V.: Alpos: a machine learning approach for analyzing microblogging data. In: The 2010 IEEE International Conference on Data Mining Workshops, pp. 1265–1272 (2010)
10. Celikyilmaz, A., Hakkani-Tur, D., Feng, J.: Probabilistic modelbased sentiment analysis of twitter messages. In: The 2010 IEEE International Conference on Spoken Language Technology, Workshop, pp. 79–84 (2010)
11. Wu, Y., Ren, F.: Learning sentimental influence in twitter. In: The 2011 International Conference on Future Computer Sciences and Application, pp. 119–122 (2011)
12. Fan, P., Li, P., Jiang, Z., Li, W., Wang, H.: Measurement and analysis of topology and information propagation on sina-microblog. In: The 2011 IEEE International Conference on Intelligence and Security Informatics, pp. 396–401 (2011)
13. Suh, B., Hong, L., Pirolli, P., Chi, E.H.: Want to be retweeted? large scale analytics on factors impacting retweet in twitter network. In: The 2010 IEEE International Conference on Social Computing/IEEE International Conference on Privacy, Security, Risk and Trust, pp. 177–184 (2010)
14. Agrawal, R., Potamias, M., Terzi, E.: Learning the nature of information in social networks. In: Proceedings of ICWSM (2012)
15. Leskovec, J., Adamic, L.A., Huberman, B.A.: The dynamics of viral marketing. ACM Trans. Web **1**(1) (2007)
16. Crandall, D., Cosley, D., Huttenlocher, D., Kleinberg, J., Suri, S.: Feedback effects between similarity and social influence in online communities. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, no. 9, pp. 160–168 (2008)
17. Bakshy, E., Hofman, J.M., Mason, W.A., Watts, D.J.: Everyone's an influencer: quantifying influence on twitter. In: Proceedings of the 4th ACM International Conference on Web search and data mining, no. 10, pp. 65–74 (2011)
18. Gruhl, D., Guha, R., Liben-Nowell, D., Tomkins, A.: Information diffusion through blogspace. In: Proceedings of the 13th International Conference on World Wide Web, no. 11, pp. 491–501 (2004)
19. Leskovec, J., Backstrom, L., Kleinberg, J.: Meme-tracking and the dynamics of the news cycle. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, no. 10, pp. 497–506 (2009)
20. Romero, D.M., Meeder, B., Kleinberg, J.: Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter. In: Proceedings of the 20th International Conference on World Wide Web, no. 10, pp. 695–704 (2011)
21. Rodriguez, M.G., Schölkopf, B.: Submodular inference of diffusion networks from multiple trees. In: Proceedings of the 29th International Conference on Machine Learning (ICML-12), pp. 489–496 (2012)
22. Du, N., Song, L., Yuan, M., Smola, A.J.: Learning networks of heterogeneous influence. In: Collection of, Advances in Neural Information Processing Systems 25, pp. 2789–2797 (2012)

23. Rodriguez, M.G., Leskovec, J., Schölkopf, B.: Structure and dynamics of information pathways in online media. In: Proceedings of the 6th ACM International Conference on Web Search and Data Mining, no. 10, pp. 23–32 (2013)
24. Du, N., Song, L., Woo, H., Zha, H.: Uncover topic-sensitive information diffusion networks. In: Proceedings of AISTATS, pp. 229–237 (2013)
25. Rodriguez, M.G., Leskovec, J., Schölkopf, B.: Modeling information propagation with survival theory. CoRR (2013)
26. Stomakhin, A., Short, M.B., Bertozzi, A.L.: Reconstruction of missing data in social networks based on temporal patterns of interactions. Inverse Probl. **27**(11), 115013 (2011)
27. Iwata, T., Shah, A., Ghahramani, Z.: Discovering Latent Influence in Online Social Activities via Shared Cascade Poisson Processes. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, no. 9, pp. 266–274 (2013)
28. Zhou, K., Zha, H., Song, L.: Learning triggering kernels for multi-dimensional Hawkes processes. In: Proceedings of ICML (3), no. 9, pp. 1301–1309 (2013)
29. Zhou, K., Zha, H., Song, L.: Learning social infectivity in sparse low-rank networks using multi-dimensional Hawkes processes. In: Proceedings of AISTATS, no. 9, pp. 641–649 (2013)
30. Hawkes, A.G.: Spectra of some self-exciting and mutually exciting point processes. Biometrika **58**(1), 83–90 (1971)
31. Hunter, D.R., Lange, K.: A tutorial on MM algorithms. Am. Stat. **58**(1), 30–37 (2004)
32. Leskovec, J., Chakrabarti, D., Kleinberg, J., Faloutsos, C., Ghahramani, Z.: Kronecker graphs: an approach to modeling networks. J. Mach. Learn. Res. **11**(58), 985–1042 (2010)
33. Erdös, P., Rényi, A.: On the evolution of random graphs. In: Proceedings of Publication of the Mathematical Institute of the Hungarian Academy of Sciences, pp. 17–61 (1960)
34. Clauset, A., Moore, C., Newman, M.E.J.: Hierarchical structure and the prediction of missing links in networks. Nature **453**(7191), 98–101 (2008)
35. Leskovec, J., Lang, K.J., Dasgupta, A., Mahoney, M.W.: Statistical properties of community structure in large social and information networks. In: Proceedings of the 17th International Conference on World Wide Web, pp. 695–704 (2008)

A Spectral Clustering Algorithm Based on Hierarchical Method

Xiwei Chen[1], Li Liu[1(✉)], Dashi Luo[1], Guandong Xu[2], Yonggang Lu[1], Ming Liu[3], and Rongmin Gao[4]

[1] School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, Gansu, People's Republic of China
{chenxw2011,liliu,luodsh12,ylu}@lzu.edu.cn
[2] Advanced Analytics Institute, University of Technology Sydney, Ultimo, NSW 2008, Australia
Guandong.Xu@uts.edu.au
[3] School of Electrical and Information Engineering, The University of Sydney, Sydney, NSW 2006, Australia
ming.liu@sydney.edu.au
[4] School of Pharmacy, Lanzhou University, Lanzhou 730000, Gansu, People's Republic of China
gaorm11@lzu.edu.cn

Abstract. Most of the clustering algorithms were designed to cluster the data in convex spherical sample space, but their ability was poor for clustering more complex structures. In the past few years, several spectral clustering algorithms were proposed to cluster arbitrarily shaped data in various real applications including image processing and web analysis. However, most of these algorithms were based on k-means, which is a randomized algorithm and makes the algorithm easy to fall into local optimal solutions. Hierarchical method could handle the local optimum well because it organizes data into different groups at different levels. In this paper, we propose a novel clustering algorithm called spectral clustering algorithm based on hierarchical clustering (*SCHC*), which combines the advantages of hierarchical clustering and spectral clustering algorithms to avoid the local optimum issues. The experiments on both synthetic data sets and real data sets show that *SCHC* outperforms other six popular clustering algorithms. The method is simple but is shown to be efficient in clustering both convex shaped data and arbitrarily shaped data.

Keywords: Data mining · Clustering · Spectral clustering · Hierarchical clustering

1 Introduction

Many clustering applications can be found in these fields, such as pattern recognition, data mining, machine learning, image analysis and agent mining [1,27], etc.

L. Cao et al. (Eds.): ADMI 2013, LNAI 8316, pp. 111–123, 2014.
DOI: 10.1007/978-3-642-55192-5_9,

Clustering is still an attractive and challenging problem. Traditional clustering algorithms, such as k-means [2], GM_EM [3], etc, while simple, most of them are based on convex spherical sample space, and their ability for dealing with complex cluster structure is poor. When the sample space is not convex, these algorithms may be trapped in a local optimum [4]. In order to solve this problem, the spectral clustering algorithm has been proposed [5].

In recent years, spectral clustering has become one of the most popular modern clustering algorithms, which can cluster arbitrarily shaped data [6]. It is simple to implement, can be solved efficiently by standard linear algebra software, and often outperforms traditional clustering algorithms. Spectral clustering algorithms derive the new features of clustering objects through matrix analysis, and then the new features are used to cluster the original data.

Many scholars have been done on how to improve the performance of spectral clustering. These researches are mainly concentrated on two aspects in the last few years: improving the computational time [7] and improving the accuracy in a particular application, including image retrieval, segmentation and recognition [8,9], social networks mining [10], speech recognizing [11]. Most of these spectral clustering algorithms are based on k-means algorithm. In the k-means algorithm, the clustering problem is treated as an optimization problem, and its objective function is a highly nonlinear and multimodal function. Furthermore, the search direction is always along the direction of decreasing energy, which makes the algorithm easy to fall into local optimal solutions [12]. Only for particular cases of initialization, the algorithm can converge to the global optimal solution. In order to solve the existing problems of k-means algorithm, we propose a novel clustering algorithm called **S**pectral **C**lustering algorithm based on **H**ierarchical **C**lustering (SCHC), which combines the advantages of hierarchical clustering and spectral clustering algorithms to avoid producing the local optimum solutions. Hierarchical clustering algorithm organizes the data into different groups at different levels, and forming a respective tree of clustering. It can be further categorized into agglomerative (bottom-up) method and divisive (top-down) method. The agglomerate algorithms treat data points or data set partitions as sub-clusters in the beginning, and then merge the sub-clusters iteratively until a stop condition is met. Divisive methods begin with a single cluster which contains all the data points, and then partition the clusters based on the dissimilarity recursively until some stop condition is reached [13]. In this paper, we use the agglomerate methods until a specific number of clusters are produced.

The rest of this paper is organized as follows. Section 2 introduces the basic knowledge of graph, and Sect. 3 describes the SCHC clustering algorithm and a hierarchy clustering algorithm which is used in our method, Sect. 4 gives the experimental results, and Sect. 5 concludes the paper with discussion.

2 Basic Knowledge of Graph

Given a set of n data points $x_1, x_2, \ldots, x_n$ with each $x_i \in R^d$, we define an affinity graph $G = (V, E)$ as an undirected graph in which the i^{th} vertex corresponds to

the data point x_i. For each edge $(i, j) \in E$, we associate a weight a_{ij} that encodes the similarity of the data points x_i and x_j. We refer to the matrix $A = (a_{ij})_{i,j=1}^{n}$ of affinities as the similarity matrix.

In the similarity matrix $A = (a_{ij}) \in R^{N \times N}$, the weight of each pair of vertices x_i and x_j is measured by a_{ij},

$$a_{ij} = \begin{cases} \exp(\frac{-||x_i - x_j||^2}{2\sigma^2}), i \neq j, \\ 0, i{=}j \end{cases} \quad i, j = 1, 2, \ldots, n,$$

and it satisfies $a_{ij} \geq 0; a_{ij}{=}a_{ji}$. Where $||x_i - x_j||^2$ can be Euclidean distance, City Block distance, Minkowski distance, or Mahalanobis distance and so on. The degree of vertex x_i is the sum of all the vertex weights adjacent to x_i, which can be defined as $D_{ii} = \sum_{j=1}^{n} a_{ij}, i = 1, 2, \ldots, n$. A diagonal matrix $D = \begin{bmatrix} D_{11} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & D_{nn} \end{bmatrix}$ can be obtained using the degree of vertices. The matrix $L = D - A$ is called Laplacian matrix. The most commonly used Laplacian matrixes are summarized in Table 1. In order to simplify the calculation the unnormalized graph Laplacian matrix is used.

Table 1. Laplacian matrixes types

Unnormalized	$L = D - A$
Symmetric	$L_{Sym} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$
Asymmetric	$L_{As} = D^{-1} L = I - D^{-1} W$

3 Our Method—SCHC

In this section, we present SCHC algorithm, which combines the advantages of spectral clustering and hierarchical clustering method. The normal row vectors T produced by spectral method (line 5) is used as the input for the hierarchical clustering (line 6) that gives the final clustering results (Table 2).

3.1 Graph Partition by the Spectral Clustering Algorithms

Spectral clustering can be interpreted by several different theories, such as figure cut set theory, random migration point and the perturbation theory [14]. But no matter what theory is used, spectral clustering can been converted to the eigenvector problem of Laplacian matrix, and then the eigenvectors are clustered.

Table 2. SCHC clustering algorithm

Input: n data points $\{x_i\}_{i=1}^n, x_i \in R^d$; the number of eigenvectors α; the number of clusters k Output: the clustering results $\{C_1, C_2, \ldots, C_k\}$
1. Construct the similarity matrix A; 2. Calculate the diagonal degree matrix D; 3. Compute the Laplacian matrix: $L = D - A$; 4. Calculate α largest eigenvectors of $L$and construct feature vector space $T = (t_1, t_2, \ldots, t_\alpha) \in R^{N \times \alpha}$; 5. Normalize the row vectors of T; 6. Using the hierarchical clustering algorithm in Sect. 3.2 to cluster the normalized row vectors into k clusters. 7. Output the clustering results $\{C_1, C_2, \ldots, C_k\}$.

The goal of spectral clustering (line 1–5) is to partition the data $\{x_i\}_{i=1}^n, x_i \in R^d$ into k disjoint classes $\{C_1, C_2, \ldots, C_k\}$, such that each x_i belongs to one and only one class, which means

$$\begin{cases} C_1 \cup C_2 \cup \ldots \cup C_k = \{x_i\}_{i=1}^n, x_i \in R^d \\ C_i \cap C_j = \emptyset, i, j = 1, 2, \ldots, k, i \neq j. \end{cases}$$

Different spectral clustering algorithms formalize this partitioning problem in different ways [5,15–17].

3.2 Hierarchical Clustering Algorithm

In the hierarchical clustering process (line 6), in order to determine whether to merge two sub-clusters into a new cluster, the distance between them is used. The detailed process of the hierarchical process is as follows:

(1) Calculate the distance between two data points which are part of the data set $x_1, x_2, \ldots, x_n, x_i \in R^d$, forming a distance matrix dist which is $N \times N$, and generate a unique class label *category, category* = 1,2,...,n for each point x_i, i = 1,2,...,n, and generating an initial value of the number of categories $num\_category = n$.
(2) The Single-Linkage method [18] is used to find the two most similar sub-clusters and they are merged into a single cluster.
(3) Repeating Step (2) until the number of clusters reaches the desired number of clusters, and output the clustering results $\{C_1, C_2, \ldots, C_k\}$.

4 Experimental Results and Analysis

In order to validate the feasibility of the proposed algorithm, we select a number of data sets that contain points in 2D space, and contain clusters of different

shapes, densities, sizes, and noise. Similar data sets can be downloaded from UNIVERSITY OF EASTERN FINLAND (http://cs.joensuu.fi/sipu/datasets/). We compared the results of k-means, FCM [19], KAP [20], GM_EM [3], HC (Hierarchical clustering algorithm), SCKM (Spectral Clustering algorithms based on K-Means) and SCHC in the experiment.

4.1 The Datasets

We use three synthetic datasets in our experiment, the properties of each data set described as follows: The Path-based data set consists of a circular cluster with an opening near the bottom and two Gaussian distributed clusters inside. Each cluster contains 100 data points. The 3-spiral data set consists of 312 points and these points are divided into 3 clusters. Both the Path-based data set and the 3-spiral data set were used in [21]. The Aggregation dataset consists of seven perceptually distinct groups of points and the total number of these points is 788. In fact, the dataset contains features that are known to create difficulties for the selected algorithms, e.g., narrow bridges between clusters, uneven-sized clusters, etc. This data set was used by Aristides Gionis etc. in [22].

In addition, two real datasets called Vehicle Silhouette data set (with 846 points in 18 dimensions) and Balance-scale data set (with 625 points in 4 dimensions) were downloaded from the UC Irvine Machine Learning Repository [23].

Table 3. Character of the five datasets

Datasets		Number of classers	Size
Synthetic datasets	Path-based	3	300
	3-Spiral	3	312
	Aggregation	7	788
Artificial datasets	Vehicle silhouette	4	846
	Balance-scale	3	625

4.2 Evaluation Criteria

There are usually two types of validation indices for evaluating the clustering results: one for measuring the quality by examining the similarities within and between clusters, and the other for comparing the clustering results against an external benchmark.

As a well-known first type index, the DB-Index [24] is used in our experiments. It is defined as

$$\mathrm{DB}-\mathrm{Index}=\frac{1}{K}\sum_{i=1}^{K}\max\frac{\sigma_i+\sigma_j}{dist(C_i,C_j)}, i\neq j,$$

where K is the number of clusters, σ_i is the square root of the intra-cluster inertia of cluster C_i and $dist(C_i, C_j)$ is the distance between the centroids of cluster C_i and C_j.

$$\sigma_i = \sqrt{\sum_{j \in C_i} \frac{dist(j, C_i)^2}{N_i}},$$

where $dist(j, C_i)$ is the distance between data point j and the centroid of cluster C_i, and N_i is the number of data points in cluster C_i. Usually the clustering results with low intra-cluster distances and high inter-cluster distances will produce a low DB-Index. When computing the DB-index, we take the mean position of all the members of a cluster as its centroid instead of using the cluster center given by the algorithm. This will ensure a better comparison because different algorithms usually have different schemes for deciding the centers of clusters. The original cluster centers determined by all the algorithms are shown as filled circles in the figures of the clustering results for reference. If there is only one cluster, a trivial value of zero is given as the DB-Index. On the other extreme, when the number of clusters is close to the number of data points, some clusters will only have one member,and the will be zero for these clusters. As a result, a small DB- Index will be produced. So the DB-Index is more meaningful when the number of clusters is greater than one and much smaller than the number of data points.

For comparing the computed clustering results with a given benchmark, the Adjusted Rand Index is used [25]. Given a set S of n elements, and two groupings (e.g. clusterings) of these points, namely $X = \{X_1, X_2, \ldots, X_r\}$ and $Y = \{Y_1, Y_2, \ldots, Y_s\}$, the overlap between X and Y can be summarized in a contingency table $[n_{ij}]$ where each entry n_{ij} denotes the number of objects in common between X_i and Y_j :$n_{ij} = |X_i \cap Y_j|$.

$X \setminus Y$	Y_1	Y_2	$\cdots$	Y_s	*Sums*
X_1	n_{11}	n_{12}	$\ldots$	n_{1s}	a_1
X_2	n_{21}	n_{22}	$\ldots$	n_{2s}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
X_r	n_{r1}	n_{r2}	$\ldots$	n_{rs}	a_r
Sums	b_1	b_2	$\cdots$	b_s	

The adjusted form of the Rand Index, the Adjusted Rand Index, is

$$AdjustedIndex = \frac{Index - ExpectedIndex}{MaxIndex - ExpectedIndex}$$

More specifically,

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{n}{2}},$$

where n_{ij}, a_i, b_j are values from the contingency table. The Adjusted Rand Index can yield a value between -1 and $+1$ [26]. The maximum value of the Adjusted Rand Index is 1, which means that the two clustering results are exactly the same. When the two partitions are picked at random which corresponds to the null model, the Adjusted Rand Index is 0 [28].

A lower DB-Index or a higher Adjusted Rand Index indicates a better clustering result. When the benchmark is available, both of the indices are measured, otherwise only the DB-Index is used in our experiments.

4.3 Implementation Details

To evaluate the proposed SCHC method, it is compared with six popular clustering methods: k-means, FCM, KAP, GM_EM, HC, and SCKM. The SCHC algorithm is implemented in Matlab R2010a, the other Matlab codes come from Matlab toolbox, or from the Matlab Central website (http://www.mathworks.com/matlabcentral/). All the experiments are run on a desktop computer with an Inter(R) Pentium(R) CPU G620 @2.60 GHz and 4 GB RAM.

The running time of a single execution, the number of clusters produced, the number of iterations, and the validation indices are recorded in Tables 4, 5, 6 and 7. For all of the clustering, we set the number of clusters in Table 3, and for the rest parameters, if any, we used Matlab's defaults. Because kmeans, FCM, GM_EM, and SCKM are all randomized algorithms, they are executed 100 times. For KAP, HC and SCHC that are not randomized algorithms, the results of each experiment are the same. We only select one experimental results, but the run time is still the average time of running 100 times. The minimum and the average value of the DB-Index, and the maximum and the average value of the Adjusted Rand Index are used in our experimental evaluation.

4.4 Results and Analysis

Experiments Using Synthetic Data Sets. For Dataset Path-based containing three clusters of a circular cluster with an opening near the bottom and two Gaussian distributed clusters inside, the results are shown in Table 4 and Fig. 1. It can be seen from Fig. 1 that kmeans, FCM, KAP cannot recognize the circular cluster, Although GM_EM and HC are possible to recognize the peripheral annular, they could not recognize the two Gaussian distributed clusters inside that they are close to each other. SCHC gives a perfect clustering result. By the ARI values in Table 4 we can see that GM_EM and SCKM can sometimes produce good results indicated by the highest ARI index. But they are not stable, and the clustering results of traditional spectral clustering SCKM can produce bad

result as shown in Fig. 1, this may be related to the randomness of the kmeans method used in SCKM. It is noticed from Table 4 that SCHC has produced a much better Adjusted Rand Index than all the other methods while its DB-Index is very high. Because the cluster shape is far beyond globular, the DB-Index is not a good quality criterion for this case.

For Dataset 3-Spiral, it containing three clusters of the same size, the results are shown in Table 5 and Fig. 2. We can see that for the dataset, k-means, FCM, KAP, and GM_EM method produce clustering results far away from the actual results. Although the maximum ARI value of SCKM reaches 1, there is a large variation in its clustering accuracy indicated by the difference between the maximum and the minimum ARI. HC and SCHC methods can produce very good results for this data set. Because the cluster shape is far beyond globular, the DB-Index is not a good quality criterion for this case either.

From Fig. 3, it can be seen that only the SCHC method perfectly identifies all the shapes in this complex data set, while all the other methods fail to identify the optimal result. Through the analysis of Table 6 we derive that although they

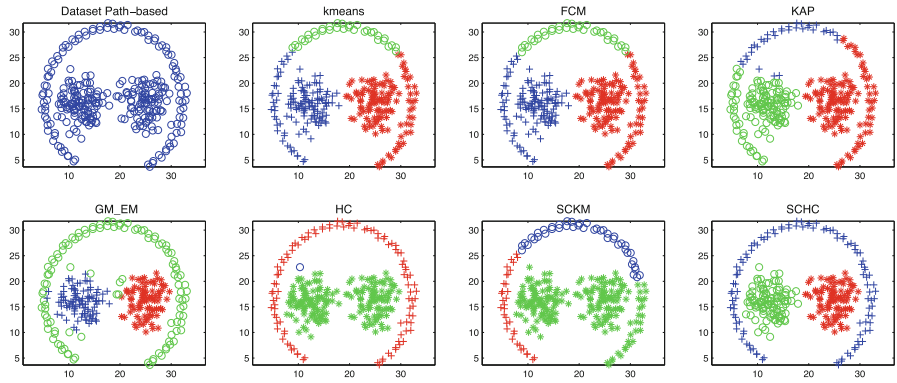


Fig. 1. Clustering results of Dataset Path-based.

Table 4. Clustering results on Dataset Path-based by different Clustering Algorithms

Dataset	Algorithms	#Clusters[a] (#Iter.)[b]	ARI MAX.	ARI AVE.	DB-Index MIN.	DB-Index AVE.	Run time[c]
Path-based	kmeans	3(100)	0.4922	0.4920	0.7531	0.7546	0.0028
	FCM	3(100)	0.4957		0.7660		0.0210
	KAP	3(1)	0.4755		0.7882		1.8239
	GM_EM	3(100)	0.9195	0.8957	2.3876	2.4095	0.5341
	HC	3(1)	0.5875		4.9253		0.0062
	SCKM	3(100)	1	0.8732	1.0623	2.3094	0.0988
	SCHC	3(1)	1		2.5443		0.2945

[a] The number of the clusters produced.
[b] The number of the iterations.
[c] The running time in seconds of a single iteration.

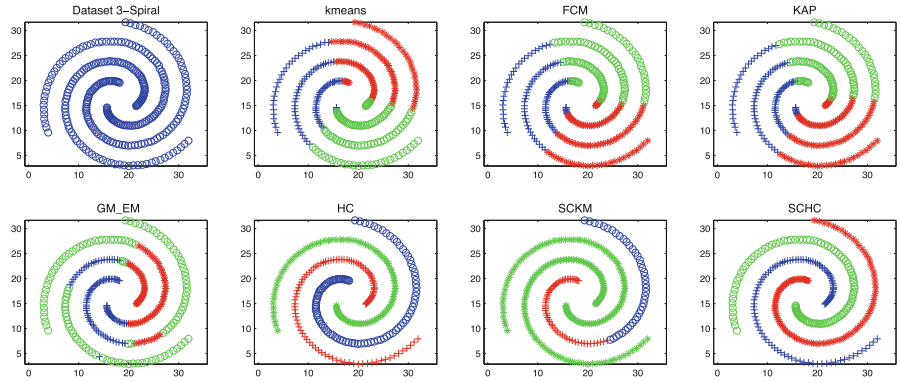


Fig. 2. Clustering results of Dataset 3-Spiral.

Table 5. Clustering results on Dataset 3-Spiral by different Clustering Algorithms

Dataset	Algorithms	#Clusters[a] (#Iter.)[b]	ARI MAX.	ARI AVE.	DB-Index MIN.	DB-Index AVE.	Run time[c]
3-Spiral	kmeans	3(100)	−0.0055	−0.0059	0.9489	0.9546	0.0059
	FCM	3(100)	−0.0057	−0.0062	0.9519	0.9531	0.0377
	KAP	3(1)	−0.0060		0.9527		1.8998
	GM_EM	3(100)	0.0628	0.0541	4.6454	5.0097	1.2336
	HC	3(1)	1		6.1355		0.0087
	SCKM	3(100)	1	0.8382	3.8434	5.7591	0.1138
	SCHC	3(1)	1		6.1355		0.3289

[a] The number of the clusters produced.
[b] The number of the iterations.
[c] The running time in seconds of a single iteration.

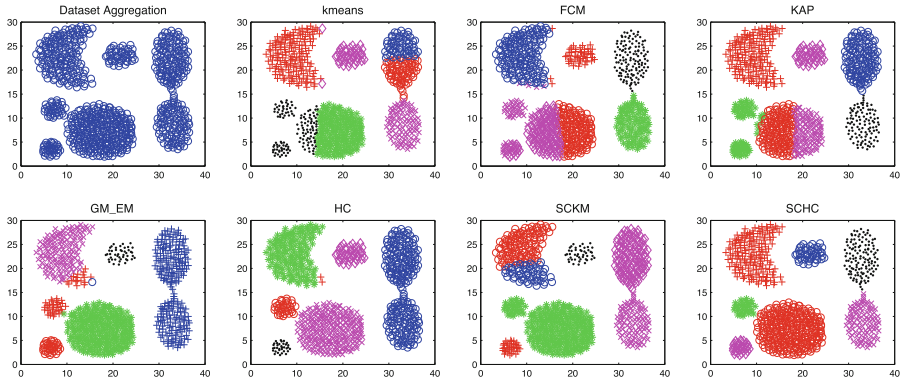


Fig. 3. Clustering results of Dataset Aggregation.

Table 6. Clustering results on Dataset Aggregation by different Clustering Algorithms

Dataset	Algorithms	#Clusters[a] (#Iter.)[b]	ARI MAX.	ARI AVE.	DB-Index MIN.	DB-Index AVE.	Run time[c]
	kmeans	7(100)	0.7782	0.7262	0.7411	0.7991	0.0075
	FCM	7(100)	0.7436	0.6830	0.7927	0.9325	0.0814
	KAP	7(1)	0.7763		0.7760		32.6706
Aggregation	GM_EM	7(100)	0.9840	0.8159	0.5822	1.2308	15.1347
	HC	7(1)	0.8795		0.6351		0.0451
	SCKM	7(100)	0.9971	0.8284	0.5414	0.8014	2.8497
	SCHC	7(1)	1		0.5372		7.5160

[a] The number of the clusters produced.
[b] The number of the iterations.
[c] The running time in seconds of a single iteration.

got ARI values are relatively high, but the precision is not very ideal, there is still much room to improve. The SCHC method produces the maximum ARI value which is 1 and the smallest DB-Index, which illustrate that effectiveness of our method for the data set.

Experiments Using Two Real Data Sets. We have applied the proposed SCHC method and the six other clustering methods to the real data sets from

Table 7. Clustering results on Real Datasets by different Clustering Algorithms

Dataset	Evaluation parameters		Algorithms kmeans	FCM	KAP	GM_EM	HC	SCKM	SCHC
	#Clusters[a] (#Iter.)[b]		4(100)	4(100)	4(1)	4(100)	4(1)	4(100)	4(1)
Vehicle	**DB-Index**	MIN.	1.1589	1.3090	1.8643	1.4523	0.9324	2.7400	0.6438
		AVE.	1.4959	2.1563		1.9912		2.9029	
	Run time[c]		0.0186	0.1173	22.3412	0.3724	0.0953	2.7370	7.6009
	#Clusters[a] (#Iter.)[b]		2(100)	2(100)	2(1)	2(100)	2(1)	2(100)	2(1)
Balance	**DB-Index**	MIN.	1.7480	1.7753	N/A[d]	1.7818	0.8546	3.1167	0.7552
		AVE.	1.7769	1.8710		1.8520		3.2337	
	Run time[c]		0.0136	0.0693	63.3676	0.0862	0.2409	0.7481	2.3452

[a] The number of the clusters produced.
[b] The number of the iterations.
[c] The running time in seconds of a single iteration.
[d] The algorithm cannot achieve the required number of clusters

the UCI Machine Learning Repository [23]. For data set vehicle silhouette comes from http://archive.ics.uci.edu/ml/datasets/statlog+(vehicle+silhouettes), $\alpha = 15$ is used to select the largest eigenvectors of L and 4 clusters have been produced in all 100 iterations for SCKM and SCHC. For data set balance-scale comes from http://archive.ics.uci.edu/ml/datasets/Balance+Scale, $\alpha = 1$ is used to select the largest eigenvectors of L and 2 clusters have been produced in all 100 iterations for SCKM and SCHC. Table 7 shows that the DB-Index of the SCHC results is the smallest among all the methods, which indicates better performance of our approach for the two real data sets compared with other methods.

Running Time Comparison. The experimental results in Tables 4, 5, 6 and 7 show that the speed of SCHC is slightly slower than the average, but it is much higher than affinity propagation (K-AP). The time complexity of the proposed SCHC algorithms is $O(n^3)$,as can be seen from the algorithm description in Sect. 3. This is a drawback of our method. Our future work is to improve the speed of this algorithm while maintaining the clustering accuracy.

5 Conclusion and Discussion

In this paper we have presented a novel spectral clustering method based on hierarchical clustering. The key idea is to use the Hierarchical Clustering instead of the k-means in a traditional spectral clustering. Experiments on real and synthetic data show that the proposed SCHC method generally outperforms state of the art methods by producing more satisfying clustering results. The proposed SCHC method also enjoys several practical implementation advantages. Firstly, it is not a randomized method, so the result can be reproduced for the same data set with the same settings. Secondly, it is simple to implement so it can be easily applied to different kinds of clustering problems.

The drawbacks of SCHC include using two parameters α and k, and a relatively slow speed. Hence, further work needs to be done on how to optimize this algorithm to improve its efficiency and on how to select the two parameters.

Acknowledgement. This work was partially supported by the National Natural Science Foundation of China (grant no.61003240), the Scientific Research Foundation for the Returned Overseas Chinese Scholars(grant order no.44th), and the Opening Project of Guangdong Province Key Laboratory of Computational Science at the Sun Yat-sen University (grant year 2012).

References

1. Qiu, H., Hancock, E.R.: Graph matching and clustering using spectral partitions. J. Pattern Recogn. Soc. **39**(1), 22–24 (2006)
2. Lloyd, P.S.: least squares quantization in PCM. IEEE Trans. Inf. Theory **28**(2), 129–137 (1982)

3. Bishop, C.M.: Pattern Recognition and Machine Learning, Ch. 9. Springer, New York (2006). ISBN 0-387-31073-8
4. Gao, Y., Gu, S., Tang, J.: Research on spectral clustering in machine learning. Comput. Sci. **34**(2), 201–203 (2007)
5. Ng, A.Y., Jordan, M., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: Advances in Neural Information Processing Systems (NIPS) (2002)
6. Ding, S., Zhang, L., Zhang, Y.: Research on spectral clustering algorithms and prospects. In: The 2nd International Conference on Computer Engineering and Technology (ICCET), vol. 6, pp. 149–153, April 2010
7. Chen, W.Y., Song, Y., et al.: Parallel spectral clustering in distributed systems. IEEE Trans. Pattern Anal. Mach. Intell. **33**(3), 568–586 (2011)
8. Wang, C., Wang, J., Zhen, J.: Application of spectral clustering in image retrieval. Comput. Tech. Dev. **19**(1), 207–210 (2009)
9. Ekin, A., Pankanti, S., Hampapur, A.: Initialization-independent spectral clustering with applications to automatic video analysis. In: IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 3, pp. 641–644, May 2004
10. Jiang, Y., Tang, C., et al.: CTSC: core-tag oriented spectral clustering algorithm on Web2.0 tags. In: The Sixth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 09), vol. 1, pp. 460–464, August 2009
11. Bach, F.R., Jordan, M.I.: Spectral clustering for speech separation. In: Automatic Speech and Speaker Recognition: Large Margin and Kernel, Methods, pp. 221–253, January 2009
12. Wang, H., Chen, J., Guo, K.: A genetic spectral clustering algorithm. J. Comput. Inf. Syst. **7**(9), 3245–3252 (2011)
13. Qian, W., Zhou, A.: Analyzing popular clustering algorithms from different viewpoints. J. Softw. **13**(8), 1382–1394 (2002)
14. Tian, Z., Li, X., Ju, Y.: The perturbation analysis of the spectral clustering. Chin. Sci. **37**(4), 527–543 (2007)
15. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 888–905 (2000)
16. Meila, M., Shi, J.: Learning segmentation with random walk. In: Advances in Neural Information Processing Systems (NIPS), pp. 470–477 (2001)
17. Yu, S., Shi, J.B.: Multiclass spectral clustering. In: Ninth IEEE International Conference on Computer Vision, vol. 1, pp. 313–319, October 2003
18. Gower, J.C., Ross, G.J.S.: Minimum spanning trees and single linkage cluster. J. Roy. Stat. Soc. Series C (Applied Statistics) **18**(1), 54–64 (1969)
19. Bezdek, J.C., Ehrlich, R., Full, W.: FCM: the fuzzy c-means clustering algorithm. Comput. Geosci. **10**(2–3), 191–203 (1984)
20. Zhang, X., et al.: K-AP: generating specified K clusters by efficient affinity propagation. In: IEEE 10th International Conference on Data Mining (ICDM), pp. 1187–1192 (2010)
21. Hong, C., Yeung, D.Y.: Robust path-based spectral clustering. Pattern Recogn. **41**(1), 191–203 (2008)
22. Gionis, A., Mannila, H., Tsaparas, P.: Clustering aggregation. In: 21st International Conference on Date of Conference, pp. 341–352, April 2005
23. http://archive.ics.uci.edu/ml
24. Davies, D.L., Bouldin, D.W.: A cluster separation measure. IEEE Trans. Pattern Anal. Mach. Intell. **1**, 224–227 (1979)
25. Hubert, L.J., Arabie, P.: Comparing partitions. J. Classif. **2**, 193–218 (1985)

26. http://en.wikipedia.org/wiki/Rand_index
27. Cao, L., Gorodetsky, V., Mitkas, P.: Agent mining: the synergy of agents and data mining. IEEE Intell. Syst. **24**(3), 64–72 (2009)
28. Lu, Y., Wan, Y.: Clustering by sorting potential values (CSPV): a novel potential-based clustering method. Pattern Recogn. **45**(9), 3512–3522 (2012)

Transitive Identity Mapping Using Force-Based Clustering

H. Van Dyke Parunak[1(✉)] and Sven Brueckner[2]

[1] Soar Technology, 3600 Green Court, Suite 600, Ann Arbor, MI 48105, USA
van.parunak@soartech.com
[2] Axon Connected, 2322 Blue Stone Hills Drive, Suite 20, Harrisonburg, VA 22801, USA
sven.brueckner@axonconnected.com

Abstract. In most information retrieval systems, software processes (whether agent-based or not) reason about passive items of data. An alternative approach instantiates each record as an agent that actively self-organizes with other agents (including queries). Imitating the movement of bodies under physical forces, we describe a distributed algorithm ("force-based clustering," or FBC) for dynamically clustering and querying large, heterogeneous, dynamic collections of entities. The algorithm moves entities in a virtual space in a way that estimates the transitive closure of the pairwise comparisons. We demonstrate this algorithm on a large, heterogeneous collection of records, each representing a person. We have some information about a person of interest, but no record in the collection directly matches this information. Application of FBC identifies a small subset of records that are good candidates for describing the person of interest, for further manual investigation and verification.

Keywords: Clustering · Transitive mapping · Self-organizing information · Active data

1 Introduction

Many real-world search problems require inexact matches against heterogeneous data sources, where no single data source can answer the query. For example:

> An unidentified male visits a clinic, signing in with an illegible signature and partially illegible phone number, then leaves the clinic before being seen. Later, staff discovers that another patient has symptoms of an influenza-like illness consistent with a potentially deadly and highly contagious virus. Staff initiates quarantine to limit close contact until the laboratory confirms diagnosis. In reviewing the sign-in log, staff discover the visitor's entry. Patients and staff cannot recall any supporting information about the unidentified individual, who is at risk and must be identified as quickly as possible.

An anonymous sponsor provided us with eight databases (DBs) (Table 1), containing varying combinations of name, address, phone number, and DB-specific record identifiers (Doc-IDs) for fictitious individuals, but concealing the identity of the POI. SRLU has 130 records, and the others have on the order of 50,000 each. Cursory examination shows that some keys are shared both within and across databases, and

L. Cao et al. (Eds.): ADMI 2013, LNAI 8316, pp. 124–135, 2014.
DOI: 10.1007/978-3-642-55192-5_10,

Table 1. Databases.—A '?' indicates that the field is present in only some records of the DB.

DB name	Available fields										
	Last name	Middle name	Middle initial	First name	Street #	Street	City	State	Zip	Phone	Doc ID
CCCR	X	X		X	X	X	X	X	X		X
CCTR					X	X	X	X	X		X
HPA	X			X		X	X	X	X		
HR					X	X	X	X	X	X	
ID	X	X		X	X	X	X	X	X		X
SRLU	X		X	X	X	X	X	X	X	X	
TR	X		?	X						X	X
WP	X			X	X	X	X	X	X	X	

some names appear to be variant spellings (e.g., "Tom F. Tuk" and "Tolman Fredegar Took" share other information). Only the last two digits of the phone number are illegible, but 104 records have phone numbers that could match the available number, some associated with different names or addresses, suggesting errors in the data. These phone numbers are in states different than the clinic. Our task is to develop a prioritized list of people with contact information whom authorities should contact in order to reach the mystery patient as quickly as possible.

Constructing and reasoning over such scenarios is combinatorially prohibitive, and too slow for emergencies (such as tracking an epidemic or disrupting a terrorist attack. Following the idea of agent mining [17], Force-Based Clustering (FBC) instantiates each record as a software agent in an abstract low-dimensional space (a three-dimensional torus wrapped in four dimensions). The agents compute virtual "forces" among themselves, and move in response to those forces. The transitivity of these forces brings together agents whose similarity may not be documented directly, but that are linked by a chain of similar agents.

Section 2 defines our algorithm. Section 3 relates it to other techniques. Section 4 reports its performance. Section 5 concludes.

2 Technical Approach

We summarize the motivation for and implementation of FBC.

2.1 Motivation

FBC is motivated by physical forces, which show several characteristics:

- If entities are close, they repel one another (mutual exclusion).
- If they are far apart, their interaction rapidly decreases (depending on the physical force involved, as the square or even higher powers of the separation).
- Multiple types of forces can contribute concurrently to interactions.

Each record in our database is an agent. We distribute them in an abstract space, define virtual forces among them, and let them move. Similar records move closer to one another, pulling their neighbors with them (and thus providing transitive closure). We query the system by inserting a query record that contains what we know about the POI, letting the system converge, and retrieving records that end up close to the query. The closer a record is to the query, the higher we rank it in our list of persons to contact. This approach emulates physical movement:

- Extremely close agents repel one another, keeping similar records from collapsing to the same location, in spite of attractive forces among them.
- The decrease of interaction strength with distance means that most interactions are among nearby agents. This locality of interaction facilitates convergence of any digital algorithm for computing agent movement. Physical forces act all at once, but an algorithm must manipulate a subset of agents at each time step. Local interactions reduce the set of agents with which a given agent effectively interacts, allowing their influence to be felt in fewer steps.
- The concept of multiple forces lets us handle heterogeneous records with varying field contents, by defining a "last-name force," an "address force," and a "phone-number-force." Integration of these forces through agent movement allows transitive interactions among records that do not directly overlap. For example, imagine that record A has only address and DB key information (database "CCTR"), B has address and name ("CCCR"), and C has name and phone number ("TR"). The "address force" will bring A and B together, the "name" force will bring B and C together, and as a result, A and C come close together, suggesting a link between the phone number in C with the DB key in A.

2.2 Implementation

Our implementation includes similarity computation, force definition, distributed execution, and convergence detection.

Similarity Computation. From the union of the fields in the databases, we define nine complex attributes in [0, 1], each derived from one or more raw attribute fields. These rules take care of missing simple attributes (e.g., a full-name complex attribute with no middle name) and may also employ external data sources in the similarity calculation. The overall similarity between two records is a weighted sum of the component similarities.

The similarity computations and weights we assign to each complex attribute reflect our understanding about the contribution each can make to the challenge.

Edge.—The data set provides not only $\sim$350 k records, but also a pre-computed set of $\sim$58 k similarity-weighted edges based on the record attributes. The "Edge" complex attribute is the total composite score of two records as recorded in this table. If the table does not specify an edge between two records, their "Edge" similarity is zero.

Source.—Though we do not know the meaning of the various databases, we must combine them to achieve transitive closure. Differences in record structure may reduce the similarity score based on substantive fields. To encourage exploration of cross-

databases similarities, we assign a similarity component of 1 between two records if their sources are different, and 0 if they are the same.

Full Name.—A person's name is the most specific and semantically meaningful identifier. Between two "Full Name" attributes we compute the similarity based on the presence and match of the three component strings (0.5 for last name match, an additional 0.3 for first name match, and a final 0.2 for a middle name match). Our current implementation determines a name component match solely on (ignore-case) string identity; applying Soundex [12] is a natural extension.

US State.—The "State" attribute of a record offers a rough indicator of its geographic location. We define a static similarity measure between all states based on the normalized geographic distance of the latitude/longitude of their capitals. Records with identical state identifiers have a similarity of 1. Records for states with the largest geographic distance of capital cities, or records that do not identify a (known) state, all have a similarity value of 0. Other similarity values are proportional in-between.

Zip.—The US postal code system offers a finer approximation of the geographic location of this record than the US-State attribute. Similarity between two Zip attribute values is first established by identity (similarity = 1). If the Zip codes are not identical but both populated, their similarity is based on the [0, 1] normalized geographic distance of the latitude/longitude coordinates of their respective geodesic centers. Unpopulated records result in a similarity of zero.

Full Address.—The "Full Address" complex attribute combines our various geographic estimates of the location a record references. Complete or partial matches of the components of the Full Address accumulate similarity contributions. Matches for State and Zip entries are provided by their respective complex attribute wrappers. City and street name strings are either identical or not. If street names are identical, then highly similar house numbers provide a small similarity bonus.

Phone.—Our only concrete identifier for the POI is a phone number, but the database contains some duplicate numbers, suggesting either shared phones or erroneous data. We accumulate units of similarity by first assessing the similarity of the area codes, then the prefix similarity, and finally the line code similarity. In this sequence, whenever there is not an identical match, further similarity accumulation will stop. For instance, there is no reason to compare line codes with different prefixes. If the area codes of two records already do not match, we use an external database of latitude/longitude coordinates of major cities or towns in the geographic coverage of the area code to provide a partial area-code similarity measure.

Gender.—The POI is male. While there is no explicit "Gender" field in the databases, the first name of a person, if provided, does provide an estimate of the likely gender. We extracted a database of 48 k international first names that estimates likely gender as one of 5 values: Strongly Female, Possibly Female, Unknown, Possibly Male, Strongly Male. Using string matching with the provided First Name, we assign each record one of these five genders. If the first name is missing, we assign Unknown. This name database gives substantial coverage of the records in the challenge data set, once we add the genders of the main characters of J.R.R. Tolkin's *Lord of the Rings* trilogy.

We manually defined a 5×5 similarity matrix between the gender identities, assigning a similarity of 1 for matching Strongly Female or Male records and decreasing similarity for more distant fields in the gender identity ordering.

ID-DOC.—We do not know the meaning of the ID-DOC keys, but a cursory survey of the data shows that some of these are shared, both within and across databases. Binary string similarity (0 or 1) is assessed.

Force Definition. The force between two agents has two components: one repulsive and one attractive. Each force is computed using the convex distance scaling function

$$g(d, m, s) = \frac{e^{s\frac{m-d}{m}} - 1}{e^s - 1}$$

where

- d is the shortest distance on the torus between the two agents,
- m is the maximum distance on the torus,
- s is a shaping factor; as s increases, force drops off more rapidly.

For similarity φ between two agents, the force is

$$f(d, \varphi) = w_a dg(d, m, s_a)\varphi - w_b mg(d, m, s_b)(1 - \varphi)$$

Figure 1 plots this force for $s_r = 5, s_a = 6, w_a = w_r = 1, \varphi = 0.97$, with repulsion at low separations, low force at large separations, and attraction in between. The force is multiplied by a maximum step length to give the distance that the agent moves in the direction of the agent with which it is being compared. The larger the step length, the more agents move on each iteration, but the more danger there is of thrashing.

Distributed Execution. We apply FBC repeatedly to pairs of points. Convergence is smoothest if step length is modest, which in turn requires each pair of points to be evaluated repeatedly. Application of this algorithm to a large number N of points thus involves $O(N^2)$ operations, which can be prohibitive for very large datasets.

The processes in our experiments execute asynchronously against a centralized DB. Each time a process is invoked, it

- Retrieves C points and their locations from the database (in our experiments, $C = 35$; all except queries are randomly chosen);
- Applies the FBC algorithm to all pairs of points in the sample, and computes new locations for them;
- Writes the points back into the database.

A process remembers the k closest agents to a query agent that it has seen. The results of the clustering search can be retrieved by merging these lists across processes.

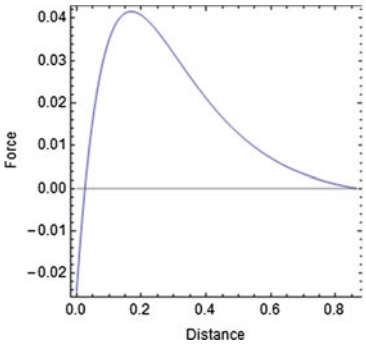


Fig. 1. Example of computed force

FBC scales linearly in both space and time in the number of processors, and offers a quadratic benefit over a naïve algorithm, regardless of the degree of distribution. Let

- N = the number of agents;
- m = the number of processors;
- C = the number of points clustered by each processor at one time;
- d = the maximum step length an agent can take in a processing cycle, expressed as a fraction of the maximum distance D on the torus.

Consider the quadratic benefit first. A naïve approach computes the similarity of each agent to all other agents, for complexity $O(N^2)$. In FBC, an agent is closely attracted to a group of g other agents, and each interaction between two agents approximates g^2 interactions, reducing the complexity by a factor of $(N/g)^2$.

Massive data invites distribution. We expect time complexity to scale linearly with the number of processors. On average, assume that each agent starts out $D/2$ from its optimal location. Then an agent needs $O(1/(2d))$ interactions to reach its destination. For simplicity of analysis, assume that processes run synchronously. The probability of an agent being selected in a processing cohort is mC/N. Each selection yields C interactions, so an agent can anticipate mC^2/N interactions per processing round, thus requiring $N/(2dmc^2)$ processing rounds to move to its optimal location. Each round takes $O(C^2)$ time, so the total processing time is $N/(2dm)$, independent of C.

FBC processes run asynchronously, so two processes may simultaneously move the same point, and only the last one to write to the database will be preserved. The chance of a record being in a given clustering process is C/N. The chance that we get some record—any record—in two processes concurrently is $N^*(C/N)^2 = C^2/N$. The number of possible pairs of processes is ${}_mC_2$, so the probability of collision is $\binom{m}{2} C^2/N$, which for $C = 35$, $N = 350{,}000$, $m = 4$ is about 4 %. While C does not affect time complexity directly, it does affect collision probability quadratically, commending a choice of small C. We do not attempt to detect these collisions, but rely on the incremental any-time nature of the algorithm to correct them over time.

Space complexity is also linear, since our clustering process needs hold in memory only the set of records being clustered. Empirically, we find that 350,000 records is at the limit of a single processor, while smaller cohorts are easily processed.

Convergence Detection. If steps are small enough, incremental distributed processes like FBC converge roughly exponentially [13]. To monitor convergence, we compare the pairwise separation of agents on the torus with their pairwise similarity, similar to Kruskal stress [8] in multi-dimensional scaling. If we were able to capture all the similarity information in the spatial distribution, the rank ordering of distances between agents in space would be the same as their rank ordering in similarity. Sets of two pairs (x_i, y_i), (x_j, y_j) of joint observations from random variables X, Y define four values:

- P is the number of such pairs in which the rank ordering of $x_\bullet$ and $y_\bullet$ is the same.
- Q is the number of pairs in which the rank ordering is different.
- T is the number of pairs in which the x values are tied.

- U is the number of pairs in which the y values are tied.

Given these values, the Kendall τ is

$$\tau = \frac{P - Q}{\sqrt{(P + Q + T) * (P + Q + U)}}$$

By construction, τ ranges from -1 if the variables are anti-correlated to +1 if they are perfectly correlated. We consider only pairs of points one of whose members is a query. Since retrieval in KBC consists of selecting those points that are nearest to the query, this measure accurately reflects the aspects of convergence that are important to us, ignoring dynamics far from the query that have no impact on retrieval.

3 Comparison with Other Technologies

The FBC algorithm merits comparison with a number of other technologies.

3.1 Semantic Analysis

A semantic approach to identity matching reasons explicitly about the semantics of each field. For example, it might represent the insight that if two people have the same address, they probably know one another. Such an approach is standard in classical artificial intelligence, and can bring a great deal of domain knowledge to bear on the problem. However, it is computationally very costly, and thus inappropriate for extremely large datasets. It also requires extensive knowledge engineering, slowing its application to problems that must be solved quickly.

FBC uses domain knowledge, in defining similarity metrics for complex attributes. The definitions discussed in Sect. 3.2 all incorporate our intuitions and semantic understanding of the problem. FBC translates these intuitions into numbers, permitting much faster computation than symbolic manipulation allows.

3.2 Cluster Analysis

Cluster analysis [7] seeks to identify entities that are near to one another by some measure. Centralized methods use a distance matrix of pairwise separation of entities, and many of them require updating this matrix repeatedly. Their complexity is thus at least $O(n^2)$, and in practice they reach their limit with datasets on the order of 10^5 elements (e.g., 202,000 galaxies in [6]). Decentralized approaches [14] typically partition the data, cluster each subset separately, then exchange either cluster parameters (such as centroids) or representative points to estimate a merged clustering.

Cluster analysis differs from FBC in three important ways.

- Constructing and maintaining a distance matrix is difficult with dynamic data. Typically, one cannot add data during clustering. FBC is any-time: it can accept new data while running (though this project does not draw on this feature).

- Cluster analysis views entities as fixed in attribute space, and applies distance measurement to them as passive objects. FBC allows them to move in an abstract low-dimensional space, actively participating in their own organization.
- A consequence of the centralized distance matrix is that all attributes participate equally in global clustering decisions, hindering the analysis of heterogeneous data. FBC allows entities to interact pairwise, drawing only on those attributes that both entities possess. Integration across heterogeneous attribute sets happens by transitive interactions, in which an entity shares some attributes with one entity, other with another, and thus intermediates their interaction.

3.3 Dimensionality-Reduction Algorithms

The movement of FBC entities is reminiscent of some iterative-update algorithms for multi-dimensional scaling (MDS), an instance of methods that reduce the dimensionality of a set of records. In general, dimensionality-reduction algorithms do not handle data that is massive, high-dimensional, dynamic, and heterogeneous. Algorithms for dimensionality reduction of distributed sensor data [4, 16] rely on the homogeneous or near-homogeneous feature spaces of sensory data. Conventional schemes for dimension reduction, such as the linear FastMap algorithm [3] or nonlinear algorithms such as IsoMap [18] or Locally Linear Embedding [15], do not handle diverse or distributed data. Some of these schemes have been adapted to a distributed environment [1, 9–11], but presume homogeneous data and a non-dynamic environment that allows iteration over static data collections on each processor. They commonly make local estimates of globally relevant data items globally, and exchange these estimates iteratively across the network of processors. In high-dimensional data, not all dimensions are relevant to every interaction. Structure among subsets of the data lies on a much lower-dimensional manifold, whose dimensions depend on the query. Systems that exploit this insight [2, 5] require access to all the data in a single process, and so do not support the distribution needed for timely processing of massive data.

4 Evaluation

We discuss preliminary results from experiments with the data set, using only the "Edge" similarity coefficient, and provide an initial assessment of the convergence characteristics of the information matching process with a small artificial data set.

4.1 Results

The information matching process is inherently parallel and can be distributed with gains essentially linear in the number of processors, e.g., in a MapReduce/Hadoop cloud-computing environment. Our experiment used three standard WinTel PCs to execute 4 clustering processes each and an additional PC to run the MySQL database with the 350 k records and their clustering coordinates. In this small setup, we arrived at the results reported here in less than two days execution even though one PC (4 processes) failed due to network problems after less than 8 hours.

The clustering space is a unit (1 × 1 × 1) box with all its 6 faces wrapped. Thus, we can operate in a finite volume without having to address edge-effects. The result is a small island of records, close to the query records, and separated by a distinct "moat" from the mass of irrelevant records.

Figure 2 shows the three queries (phone, address, phone + address) in the upper right, and also records that have edges in the challenge data edge table *and* either set "NY" as their State, "Bethesda" as their city, or "212" as their area code. Adjacent to the queries, a "moat" separates a relatively small set of relevant neighbors from the rest of the data.

This plot illustrates two additional features of FBC.

1. All of the 7,195 records in this figure are generally relevant to the problem, yet FBC locates many of them far from the query, showing its selectivity.
2. There is considerable structure within the "irrelevant" records, suggesting that FBC can process multiple queries at the same time.

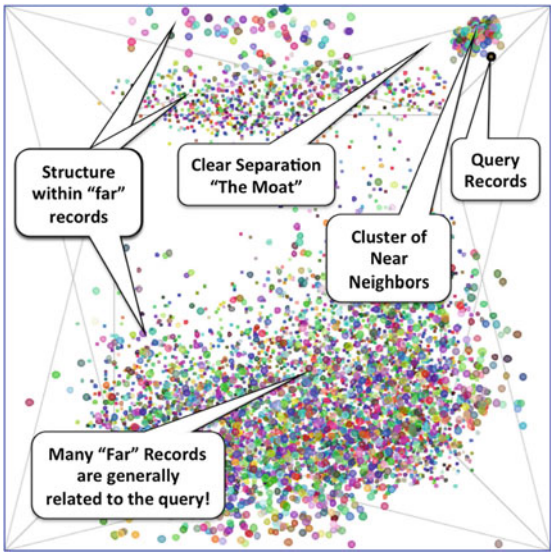


Fig. 2. 7.2 k converged records in clustering space

Figure 3 explores our "transitivity" claim, plotting for all pairs of (record, query) pairs the explicit edge similarity vs. their distance in clustering space. The gross structure of the arrangement shows many records far from the queries

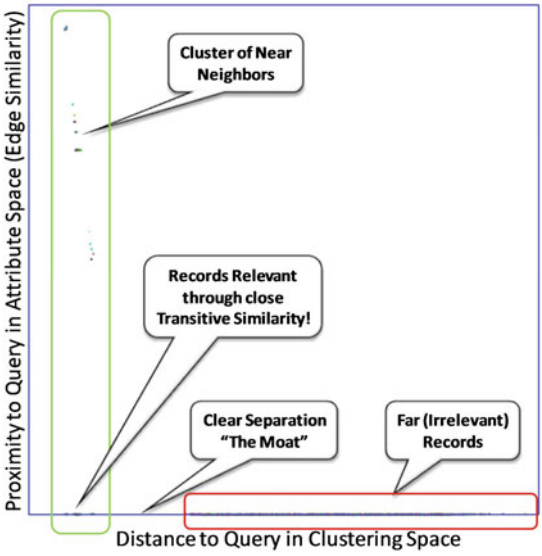


Fig. 3. Correlation between clustering distance and pair-wise edge similarity of data records to query records

(all without direct similarity), the empty space in the mid-region, and the cluster of relevant records near the queries. In the fine structure of the query neighbors, data records with non-zero direct similarity are ordered in clustering distance according to their similarity. Inn confirmation of transitivity, many data records near the query records have no direct similarity to those queries, but have been pulled in by the transitive nature of FBC.

How well do we solve the initial challenge? In our interface, we select one query record, which triggers the collection of a few of the nearest neighbors to the query agents in clustering space. We order these by their cluster-space distance and estimate their match strength from their normalized (by maximum possible distance in space) distance to the selected query record. Selecting any neighbor from that list triggers an excursion into the underlying attribute-similarity space. In an exhaustive recursive process starting at the query record, we are looking for non-trivial (more than one step) transitive paths that lead to the selected data record. A data record with more than 92 % similarity to the query record based on distance in clustering space and ranked #6 among all neighbors of this query, is connected to the query through only two intermediate records (note that these records themselves do not have to be close to the query in clustering space). We discover this path of records from the (incomplete phone number) query to the 6^{th}-nearest neighbor:

- The unidentified male "Mr. X" lives in New York with his wife, Gloriana D. Brandybuck. Their land-line phone number at 3306 Rosewood Lane, New York, NY 10003 is consistent with the clinic records.
- Since they have just married, Gloriana Donnamira Brandybuck's driver's license or credit card (a Doc-ID) is still registered with her old address at 2719 Pin Oak Drive, Manhattan, NY 10018.
- They traveled to Maryland and checked them into a hotel at 18 Wayback Road, Bethesda, MD 20014, using the same Doc-ID.
- The day after their arrival, Mr. X fell ill and decided to visit the medical clinic nearby at 4408 East Madison Ave., Bethesda, MD, 20014.

The sponsor verified the correctness of our solution.

4.2 Convergence Assessment

We claim that FBC can be distributed over many processing units by aggressively sub-sampling the number of clustering interactions that have to be computed. We also claim that this distributed process has exponential convergence characteristics that provide a good answer fast and improved answer if more time is available (any-time characteristic). To assess these claims, we performed an initial experiment with an artificial data set of 350 color (RGB) data records, groups into seven clusters, and one query record. We start the experiment with a random arrangement of the records' agents in cluster space and run to (manually determined) convergence. We repeat the experiment varying C, emulating distributed execution with $C/351$ parallel processes. Our sequential execution of the random sampling ignores the possibility for collisions (the same record moved by more than one process at the same time), which will introduce additional noise.

Figure 4 assesses the impact of parallelization by scaling the x-axis for each data series by C and correcting for the movement of the query record. Thus scaled, the convergence curves trend very closely to each other, suggesting that a nearly linear speed-up with the number of processors may be accomplished in a distributed setting. The additional noise introduced by the sub-sampling with small c actually improves the final quality of the converged result. We hypothesize that, similar to simulated annealing processes for instance, the noise prevents the clustering from falling into sub-optimal stable states and instead drives it closer to the global optimum.

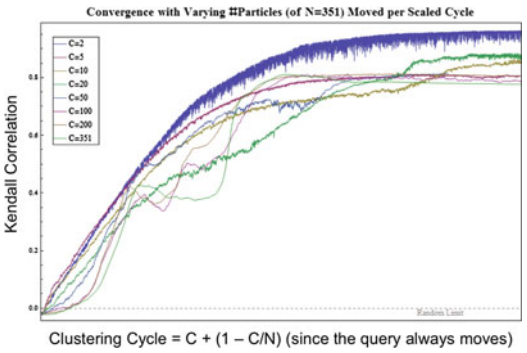


Fig. 4. Kendall correlation over clustering cycle * c.

5 Conclusion

Many problems in epidemiology and domestic security require the ability to discover transitive linkages across heterogeneous databases rapidly, without reasoning explicitly about possible scenarios. Instead of reasoning about the various records, Force-Based Clustering (FBC) turns each record into a software agent that moves in an abstract information space in response to the net "force" it feels from other agents. These forces in turn are defined by generic similarity measures over commonly occurring fields, measures that can readily be defined in advance and applied quickly to available information. The agent interactions can be distributed over many processors to speed the clustering process. Application of this approach to a synthetic data set (provided by an anonymous sponsor external to our research group) allows us to identify the person of interest. Potential extensions include tuning the weights of different features, providing for human direction of the processing, distributing data as well as processing via Hadoop, and exploring dynamically changing data.

References

1. Abu-Khzam, F.N., Samatovaz, N., Ostrouchov, G., Langston, M.A., Geist, A.: Distributed dimension reduction algorithms for widely dispersed data. In: Fourteenth IASTED International Conference on Parallel and Distributed Computing and Systems (IASTED PDCS 2002), pp. 167–174. ACTA Press (2002)
2. Aggarwal, C.C., Yu, P.S.: Finding generalized projected clusters in high dimensional spaces. In: SIGMOD Conference, pp. 70–81 (2000)
3. Faloutsos, C., Lin, K.-I.D.: FastMap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In: ACM SIGMOD, San Jose, CA, pp. 163–174 (1995)

4. Fang, J., Li, H.: Optimal/near-optimal dimensionality reduction for distributed estimation in homogeneous and certain inhomogeneous scenarios. IEEE Trans. Signal Process. **58**(8), 4339–4353 (2010)
5. Hinneburg, A., Aggarwal, C.C, Keim, D.A.: What is the nearest neighbor in high dimensional spaces? In: 26th International Conference on Very Large Data Bases (VLDB 2000), pp. 506–515. Morgan Kaufmann, Cairo (2000)
6. Jang, W., Hendry, M.: Cluster analysis of massive datasets in astronomy. Stat. Comput. **17**(3), 253–262 (2007)
7. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York (1990)
8. Kruskal, J.B.: Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika **29**, 1–27 (1964)
9. Magdalinos, P., Doulkeridis, C., Vazirgiannis, M.: K-landmarks: distributed dimensionality reduction for clustering quality maintenance. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS (LNAI), vol. 4213, pp. 322–334. Springer, Heidelberg (2006)
10. Magdalinos, P., Doulkeridis, C., Vazirgiannis, M.: a novel effective distributed dimensionality reduction algorithm. In: SIAM Feature Selection for Data Mining Workshop (SIAM-FSDM'06), Bethesda, MD (2006)
11. Magdalinos, P., Vazirgiannis, M., Valsamou, D.: Distributed knowledge discovery with non linear dimensionality reduction. In: Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) PAKDD 2010. LNCS, vol. 6119, pp. 14–26. Springer, Heidelberg (2010)
12. NARA: The Soundex Indexing System. National Archives and Records Administration. http://www.archives.gov/research/census/soundex.html (2007)
13. Parunak, H.V.D., Brueckner, S.A., Sauter, J.A., Matthews, R.: Global convergence of local agent behaviors. In Proceedings of Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS05), pp. 305–312. ACM (2005)
14. Rajaraman, A., Ullman, J.D.: Mining of Massive Datasets. Cambridge University Press, Cambridge (2011)
15. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**(5500), 2323–2326 (2000)
16. Roy, O., Vetterli, M.: Dimensionality reduction for distributed estimation in the infinite dimensional regime. IEEE Trans. Inf. Theory **54**(4), 1655–1669 (2008)
17. Cao, L., Gorodetsky, L., Mitkas, P.: Agent mining: the synergy of agents and data mining. IEEE Intell. Syst. **24**(3), 64–72 (2009)
18. Tenenbaum, J.B., Silva, Vd, Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290**, 2319–2323 (2000)

Author Index