Francesco Buccafurri
Giovanni Semeraro (Eds.)

LNBIP 61

E-Commerce and Web Technologies

11th International Conference, EC-Web 2010
Bilbao, Spain, September 2010
Proceedings

Springer

Lecture Notes in Business Information Processing 61

Francesco Buccafurri
Giovanni Semeraro (Eds.)

E-Commerce and Web Technologies

11th International Conference, EC-Web 2010
Bilbao, Spain, September 1-3, 2010
Proceedings

Springer

Volume Editors

Francesco Buccafurri
Università degli Studi Mediterranea di Reggio Calabria
DIMET Dept.
via Graziella, loc. Feo di Vito, 89122 Reggio Calabria, Italy
E-mail: bucca@unirc.it

Giovanni Semeraro
Università degli Studi di Bari
Dipartimento di Informatica
Via E. Orabona, 4, 70126 Bari, Italy
E-mail: semeraro@di.uniba.it

Library of Congress Control Number: 2010932190

ACM Computing Classification (1998): J.1, K.4.4, I.2.11, H.3.5, H.4

ISSN 1865-1348
ISBN-10 3-642-15207-4 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-15207-8 Springer Berlin Heidelberg New York

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180 5 4 3 2 1 0

Preface

After the lesson learned during last years and following the successful edition of EC-Web 2009, for its 11th edition EC-Web tried to provide a clearer description of the electronic commerce universe focusing on some relevant topics. The main focus was not only on Internet-related techniques and approaches. The aim of EC-Web 2010 was to also cover aspects related to theoretical foundations of e-commerce, business processes as well as new approaches exploiting recently emerged technologies and scenarios such as the Semantic Web, Web services, SOA architectures, mobile and ubiquitous computing, just to cite a few. Due to their central role in any realistic e-commerce infrastructure, security and privacy issues were widely considered, without excluding legal and regulatory aspects. The choice of the above relevant topics directly reflects the fact that electronic commerce (EC), in the last few years, has changed and evolved into a well-established and founded reality both from a technological point of view and from a scientific one. Nevertheless, together with its evolution, new challenges and topics have emerged as well as new questions have been raised related to many aspects of EC.

Keeping in mind the experience of the last edition of EC-Web, we maintained, for its 11th edition, the structure and the scientific organization of EC-Web 2009, aiming to highlight the autonomous role of the different (sometimes heterogeneous) aspects of EC, without missing their interdisciplinary scope. Thus, we organized the conference into four "mini-conferences," each for a relevant area of EC and equipped with Area Chairs. Both the submission and the review process reflected the organization into the four tracks, namely: Agent-Based Electronic Commerce (Chairs: Helder Coelho - Fernando Lopes), Service-Oriented E-Commerce and Business Processes (Chair: Florian Daniel), Recommender Systems (Chairs: Marco de Gemmis - Pasquale Lops) and E-Payment, Security and Trust (Chair: Barbara Masucci).

We received a broad spectrum of submissions and we are confident that the papers that were finally selected for publication and presentation contribute to a better understanding of EC issues and possibilities in the Web 2.0 and Web 3.0 eras. We are grateful to all authors for their submissions. All papers were reviewed by at least three reviewers, either members of the Program Committee or external experts in the field. In all, 45 papers were submitted and the Program Committee selected the 22 papers published in this volume. We received submissions from 18 different countries located in four continents, namely, Austria, Australia, Belgium, Canada, France, Germany, Greece, Ireland, Italy, Portugal, Romania, Slovakia, Spain, Taiwan, The Netherlands, Turkey, the UK, and the USA.

The three invited speakers provided a fundamental contribution to the success of the conference. Specifically, Ricardo Baeza-Yates outlined the science and the technology behind Web advertising, while Tommaso Di Noia and Azzurra Ragone presented different semantic-based approaches to matchmaking and negotiation in electronic markets, showing how semantics can lead to a new generation of EC systems.

We wish to thank Track Chairs Helder Manuel Ferreira Coelho, Florian Daniel, Marco de Gemmis, Fernando Lopes, Pasquale Lops and Barbara Masucci for their valuable contribution and support as well as all the Program Committee members of each track and external reviewers. Our thanks also go to Roland Wagner and to Gabriela Wagner for their great support in every single step of the organization. We do not forget Amin Anjomshoaa, who supported us with ConfDriver and changed the system according to our needs, and Leo Iaquinta, Cataldo Musto and Fedelucio Narducci, who served as Web masters and reacted promptly to our requests. We are very grateful to them all.

September 2010

Giovanni Semeraro
Francesco Buccafurri

Organization

Program Chairs

Francesco Buccafurri	Università degli Studi Mediterranea di Reggio Calabria, Italy
Giovanni Semeraro	University of Bari "Aldo Moro", Italy

Track Chairs

Agent-Based Electronic Commerce

Helder Manuel Ferreira Coelho	University of Lisbon, Portugal
Fernando Lopes	National Research Institute, Lisbon, Portugal

Service Oriented E-Commerce and Business Process

Florian Daniel	University of Trento, Italy

Recommender Systems

Marco de Gemmis	University of Bari "Aldo Moro", Italy
Pasquale Lops	University of Bari "Aldo Moro", Italy

E-Payment, Security and Trust

Barbara Masucci	University of Salerno, Italy

Program Committee

Agent-based Electronic Commerce

Holger Billhardt	University Rey Juan Carlos, Spain
Luis Botelho	Lisbon University Institute (ISCTE), Portugal
Miguel Carmona	University of Alcalá, Spain
Helder Coelho	University of Lisbon, Portugal
Edith Elkind	Nanyang Technological University, Singapore
Alberto Fernández	University Rey Juan Carlos, Spain
Masabumi Furuhata	Japan Advanced Institute of Science and Technology, Japan
Nicola Gatti	Politecnico di Milano, Italy
Massimiliano Giacomin	University of Brescia, Italy
Koen Hindriks	Delft University of Technology, The Netherlands
Joris Hulstijn	Vrije University, The Netherlands

Wojtek Jamroga	University of Luxembourg, Luxembourg
Sverker Janson	Swedish Institute of Computer Science, Sweden
Souhila Kaci	Artois University, France
Paulo Leitão	Polytechnic Institute of Bragança, Portugal
Fernando Lopes	National Research Institute (LNEG), Portugal
Paulo Novais	University of Minho, Portugal
Nir Oren	King's College London, UK
Gabriella Pigozzi	University of Luxembourg, Luxembourg
Alberto Sardinha	Lancaster University, UK
Murat Sensoy	University of Aberdeen, UK
Mathijs Weerdt	Delft University of Technology, The Netherlands
Dongmo Zhang	University of Western Sydney, Australia

Service Oriented E-Commerce and Business Processes

Cinzia Cappiello	Politecnico di Milano, Italy
Sven Casteleyn	Vrije Universiteit Brussel, Belgium
Marco Comuzzi	Eindhoven University of Technology, The Netherlands
Alfredo Cuzzocrea	Italian National Research Council, Italy
Paolo Giorgini	University of Trento, Italy
Chang Heng	Huawei Technologies, Shenzhen, P.R.China
Heiko Ludwig	IBM TJ Watson Research Center, USA
Ralph Mietzner	University of Stuttgart, Germany
Hamid Motahari	HP Labs, Palo Alto, USA
Emmanuel Pigout	SAP Research, France
Azzurra Ragone	Politecnico di Bari, Italy
Florian Rosenberg	CSIRO ICT Centre, Australia
Michael Weiss	Carleton University, Ottawa, Canada
Uwe Zdun	Vienna University of Technology, Austria
Christian Zirpins	University of Karlsruhe, Germany

Recommender Systems

Giambattista Amati	Fondazione Ugo Bordoni, Italy
Sarabjot Singh Anand	University of Warwick, UK
Liliana Ardissono	University of Torino, Italy
Giuliano Armano	University of Cagliari, Italy
Pierpaolo Basile	University of Bari "Aldo Moro", Italy
Bettina Berendt	KU Leuven, Belgium
Shlomo Berkovsky	CSIRO, Australia
Robin Burke	De Paul University, USA
Ivan Cantador	Universidad Autónoma de Madrid, Spain
Pablo Castells	Ciudad Universitaria de Cantoblanco, Spain
Federica Cena	University of Turin, Italy
Antonina Dattolo	University of Udine, Italy
Rosta Farzan	Carnegie Mellon University, Pittsburgh, USA
Alexander Felfernig	University of Klagenfurt, Austria

Michele Gorgoglione	Polytecnico di Bari, Italy
Dietmar Jannach	Dortmund University of Technology, Germany
Robert Jäschke	University of Kassel, Germany
Alípio Mário Jorge	University of Porto, Portugal
Alfred Kobsa	University of California, Irvine, USA
Francisco J Martin	Strands Inc.
Bhaskar Mehta	Google Inc.
Alessandro Micarelli	Roma Tre University, Rome, Italy
Stuart E. Middleton	University of Southampton, UK
Bamshad Mobasher	De Paul University, USA
Olfa Nasraoui	University of Louisville, USA
Cosimo Palmisano	Aizoon srl, Turin, Italy
Gabriella Pasi	Bicocca University, Milan, Italy
Roberto Pirrone	University of Palermo, Italy
Azzurra Ragone	Polytecnico di Bari, Italy
Francesco Ricci	Free University of Bozen-Bolzano, Italy
Shilad Sen	Macalester College, USA
Carlo Tasso	University of Udine, Italy
Eloisa Vargiu	University of Cagliari, Italy
Markus Zanker	University of Klagenfurt, Germany

E-Payment, Security and Trust

Mikaël Ates	Entr'Ouvert, Free Software Company, France
Anna Lisa Ferrara	University of Salerno, Italy
Matthew Green	Independent Security Evaluators, USA
Audun Jøsang	University of Oslo, Norway
Seny Kamara	Microsoft Research, USA
Gianluca Lax	Università Mediterranea di Reggio Calabria, Italy
Jose A. Onieva González	Universidad de Malaga, Spain
George Stephanides	University of Macedonia, Greece
Allan Tomlinson	University of London, UK

External Reviewers

Jonathan Gemmell	DePaul University, Chicago, USA
Silvia Calegari	University of Milano-Bicocca, Italy
Umberto Panniello	Polytecnico di Bari, Italy
Marcos A. Domingues	University of Porto, Portugal

Table of Contents

Recommender Systems 1

Recommender Systems 2

E-Payment, Security and Trust

Service Oriented E-Commerce and Business Processes

Invited Talk

Agent-Based Electronic Commerce 1

Agent-Based Electronic Commerce 2

Recommender Systems 3

Invited Talk

Recommender Systems 4

Resource Recommendation in Collaborative Tagging Applications

Jonathan Gemmell, Thomas Schimoler, Bamshad Mobasher, and Robin Burke

Center for Web Intelligence
School of Computing, DePaul University
Chicago, Illinois, USA
{jgemmell,tschimoler,mobasher,rburke}@cdm.depaul.edu

Abstract. Collaborative tagging applications enable users to annotate online resources with user-generated keywords. The collection of these annotations and the way they connect users and resources produce a rich information space for users to explore. However the size, complexity and chaotic structure of these systems hamper users as they search for information. Recommenders can assist the user by suggesting resources, tags or even other users. Previous work has demonstrated that an integrative approach which exploits all three dimensions of the data (users, resources, tags) produce superior results in tag recommendation. We extend this integrative philosophy to resource recommendation. Specifically, we propose an approach for designing weighted linear hybrid resource recommenders. Through extensive experimentation on two large real world datasets, we show that the hybrid recommenders surpass the effectiveness of their constituent components while inheriting their simplicity, computational efficiency and explanatory capacity. We further introduce the notion of information channels which describe the interaction of the three dimensions. Information channels can be used to explain the effectiveness of individual recommenders or explain the relative contribution of components in the hybrid recommender.

Keywords: Collaborative Tagging, Information Channel, Hybrid Recommender.

1 Introduction

Collaborative tagging applications such as Citeulike[1] and LastFM[2] allow users to annotate online resources with arbitrary tags. These tags aid users as they organize, share and navigate online content. Tagging applications have many advantages over more traditional web applications. Resources are categorized by several tags, rather than a single branch of a hierarchy. Tagging systems also benefit from the opinions of many users rather than a dominant view provided by a few 'experts.' They are consequently more nimble, able to adjust to a changing vocabulary and absorb trends quickly.

While recommendation algorithms are a staple of many online applications, the complex data structure makes them even more critical in tagging systems. Resources, tags

[1] www.citeulike.org

[2] www.last.fm

F. Buccafurri and G. Semeraro (Eds.): EC-Web 2010, LNBIP 61, pp. 1–12, 2010.

or even other users can be recommended in a variety of contexts. Moreover, the recommenders must contend with a three-dimensional information space.

To better understand the flow of information within tagging data, we propose the notion of information channels and provide two metrics for their evaluation. Information channels describe the level of interaction between users, resources and tags. The quality of these interactions can impact the effectiveness of various recommendation algorithms.

Leveraging the three-dimensional information requires an integrative technique. Graph models and tensor factorization, for example, have been successfully used for the recommendation of tags. These approaches have two drawbacks: they often rely on computationally complex algorithms which may not be practically scalable to real online systems; and they produce models which are difficult to interpret.

In this work, we turn our attention to resource recommendation and propose the use of linear weighted hybrid recommenders. This integrative technique combines the output of several component recommenders. The hybrid maintains the simplicity, computational efficiency and explanatory powers of its components.

The rest of the paper is organized as follows. In Section 2 we present related work on recommendation techniques. Section 3 introduces information channels. We present our linear weighted hybrid scheme in Section 4. Our experimental results and evaluation are offered in Section 5.

2 Related Work

One of the first techniques to demonstrate the value of an integrative approach was FolkRank [8], an adaptation of the well known PageRank algorithm for collaborative tagging data. While this approach produces excellent results in tag recommendations, its computational requirements make it ill suited for large scale deployment. This approach works best when it can triangulate elements from one dimension (i.e. tags) given elements from the two other dimensions (i.e. a users and a resource). For resource recommendation, however, the input consists solely of a user. FolkRank is unable to effectively exploit its graph model and produces inferior results.

Tensor factorization is another integrative solution for making recommendations in tagging applications. Tucker decomposition is one such example that factors the three dimensional tagging data into three features spaces and a core residual tensor [18,19]. This method predicts the relevance of elements from one dimension of the data given elements from the other two dimensions. Unlike FolkRank the time required to produce a recommendation is quite fast. However, the time required to build the model is far too great to be applicable to any real world application.

A pair-wise interaction tensor factorization model has also been proposed which offers far more reasonable running times in both the building of the model and the production of recommendations [12,13]. It has been used to optimize the ranking of tags given the user-resource pairs in the data. Tags may then be recommended for a new user-resource pair. While this model captures user-tag and resource-tag interactions, it does not directly capture the relation between users and resources making it ill suited for resource recommendation. It remains unclear if the algorithm can be adapted to identify user-resource interactions without breaking certain assumptions inherent in the model. All these techniques are unable to explain why a recommendation has been made.

Our previous work on tag recommendation [3,4] has demonstrated that hybrid algorithms [1] can integrate the three dimensions of the data. Linear weighted hybrids were constructed from a mixture of popularity based, model based and collaborative filtering algorithms. The most successful hybrids were those that incorporated component recommenders that relied on complimentary dimensions of the data. In this work we extend this approach to resource recommendation.

3 Information Channels

We begin this section by discussing the data model. We then discuss how an understanding of the data model facilities the notion of information channels, the interaction between users, resources and tags. We provide two metrics for analyzing the strengths of information channels and briefly discuss how they can be used to explain the performance of recommenders.

At the heart of a collaborative tagging application is the annotation; a user describes a resource with one or more tags. A collection of annotations results in a complex network of interrelated users, resources and tags [10]. Collaborative tagging data can be described as a four-tuple: $\langle U, R, T, A\rangle$, where, U is a set of users; R is a set of resources; T is a set of tags; and A is a set of annotations. An annotation contains a user, resource and all tags the user applied to the resource.

The data may be viewed as a hyper-graph [11] with users, tags and resources represented as nodes and the annotations represented as hyper-edges. Alternatively, it can be viewed as a three dimensional matrix, *URT*, in which an entry *URT(u,r,t)* is 1 if u tagged r with t. Aggregate projections of the data can be constructed, reducing the dimensionality but sacrificing information [11]. The relation between resources and tags can be defined as $RT(r,t)$. In this work we define $RT(r,t)$ as the number of users that have applied t to r.

This notion strongly resembles the "bag-of-words" vector space model [14] and is analogous to the idea of *term frequency* common in information retrieval. A similar two dimensional projection can be constructed for UT, in which an entry contains the number of times a user has applied a tag to any resource. Finally, UR is a binary matrix indicating whether or not a user has annotated a resource. An alternative approach would be to define an entry in the matrix as the number of tags a user has applied to a resource. Our previous work and continued experimentation has shown that the binary model for UR produces better results.

The interaction between users, resources and tags suggests several information channels which may be exploited by resource recommenders. For example, the channel from resources to tags produces a highly descriptive model of the resources, while the channel from resources to users provides an alternative model. The manner in which users interact with the system will result in information channels with varying predictive power. If the users of a tagging system are motivated to organize their resource for later retrieval, then the channel between resources and tags may become well developed. Alternatively, if the emphasis of the system is on sharing resources with friends, the resource-user channel may become dominant. An understanding of the information channels of a collaborative tagging application may inform the selection of recommendation algorithms or the design of hybrid recommenders. To that end, we present two metrics for evaluating the agreement between two information channels: RMSE and ANA.

Each resource, r, may be modeled as a vector over the multi-dimensional space of tags, where a weight, $w(t_i)$, in dimension i corresponds to the prominence of a particular tag t_i:

$$\boldsymbol{r^t} = \langle w(t_1), w(t_2) ... w(t_{|T|}) \rangle \tag{1}$$

Similarly, a resource can be modeled as a vector over the space of users to produce $\boldsymbol{r^u}$, where each weight, $w(u_i)$, corresponds to the importance of a particular user u_i. Analogous vector models can be constructed for users ($\boldsymbol{u^r}$,$\boldsymbol{u^t}$) and tags ($\boldsymbol{t^u}$,$\boldsymbol{t^r}$). We draw the weights directly from the previously constructed aggregate projections UR, UT and RT. The model of a user, resource or tag is defined as a row or column taken from the one of the projections.

Given that any element of the tagging data may be modeled as a vector over two dimensions, our first metric evaluates how well these dimensions agree. We adapt the well known *root mean square error* for this purpose and define it for resources as:

$$RMSE(R) = \sqrt{\frac{\sum_{i,j} (\sigma(\boldsymbol{r_i^u}, \boldsymbol{r_j^u}) - \sigma(\boldsymbol{r_i^t}, \boldsymbol{r_j^t}))^2}{m}} \tag{2}$$

where m is the number of resource pairs and σ is the similarity between the ith and jth resources. Several techniques exist to calculate the similarity between vectors. In this work we employ cosine similarity. Intuitively, if $RMSE(R)$ is low then we can say that there is general agreement between the $\boldsymbol{r^u}$ and $\boldsymbol{r^t}$ models. Information captured in the resource-user channel might therefore be useful in making predictions in the resource-tag channel. On the other hand, if $RMSE(R)$ is high then the two models often disagree on the similarity between resources and the resource-user information will have little predictive power for connecting resources and tags. Equivalent calculations can be made for $RMSE(U)$ and $RMSE(T)$.

RMSE equally penalizes a disagreement in similarities whether it occurs when two resources are quite alike or when they are otherwise different. The former might negatively impact a recommendation algorithm such as user-based collaborative filtering which focuses on the similarity among users while the later would not.

We therefore propose *average neighborhood agreement* or *ANA*. A neighborhood, $N^u(r)$, of the k most similar resources to r is constructed using the cosine similarity and the $\boldsymbol{r^u}$ models. A second neighborhood, $N^t(r)$, is constructed using the $\boldsymbol{r^t}$ models. We then calculate $ANA(R)$:

$$ANA(R) = \frac{\sum_{r \in R}(|N^u(r) \cap N^t(r)|)/k}{|R|} \tag{3}$$

ANA(U) and *ANA(T)* can be similarly calculated. If $N^u(r)$ and $N^t(r)$ are populated with many of the same elements it indicates that the $\boldsymbol{r^u}$ and $\boldsymbol{r^t}$ models are capturing similar information. As with RMSE, ANA could be used to explain how information in one information channel relates to information from another information channel. Unlike *RMSE*, *ANA* is focused on the most similar elements and is unaffected by discrepancies between otherwise dissimilar resources. Taken together these metrics can be used to describe the information channels of a collaborative tagging application and explain the performance of recommendation algorithms.

4 Hybrid Resource Recommendation

In this section we first explore the expected input and output of our recommenders. We then present several simple methods which will be used in our hybrid models and for comparative purposes. Lastly, we discuss how these components can be aggregated into a linear weighted hybrid resource recommender.

4.1 Resource Recommendation

Recent work has explored resource recommendation in collaborative tagging applications [5,6,8,17]. As in traditional recommendation algorithms the input is a user, u, and the output is a set of items, in this case a set of recommended resources, R_r. Unlike traditional algorithms that normally assume a two-dimensional relationship between users and resources, recommendation in a collaborative tagging system must also contend with tags. This dimension adds new opportunities and additional complexity.

For each of our resource recommenders we assume that it accepts a user-resource pair and returns a score, $\phi(u, r)$, describing the relevance of the resource to the user. Given u, the resources are sorted by their corresponding scores and the top n resources are recommended to the user.

Popularity Model. Perhaps the simplest recommender is one which merely recommends the most popular resources. We call this approach Pop and define its score as:

$$\phi(u, r) = \sum_{v \in U} \theta(v, r) \tag{4}$$

where $\theta(v, r)$ is 1 if v has annotated r and 0 otherwise. While Pop is not an effective recommender, it does serve as a baseline and may contribute to the hybrid recommender. In the rare case that a recommender is unable to populate a recommendation set, we rely on Pop to complete the task.

User-Based Collaborative Filtering. User-based collaborative filtering [7,9,16] works under the assumption that users who have agreed in the past are likely to agree in the future. A neighborhood of the most similar users is identified through a similarity metric. For any given resource the weighted sum can then be calculated as:

$$\phi(u, r) = \sum_{v \in N} \sigma(u, v)\theta(v, r) \tag{5}$$

where N is the k nearest neighbors to u and $\sigma(u, v)$ is the cosine similarity between the users u and v. As before $\theta(v, r)$ is 1 if v has annotated r and 0 otherwise.

Collaborative tagging application offer unique opportunities to model the user. When users are modeled as resources we call this approach KNN_{ur}. When users are modeled as tags we call this technique KNN_{ut}.

Item-Based Collaborative Filtering. Item-based collaborative filtering [2,15] relies on discovering similarities among resources rather than among users. We may model the resources as a vector over the user space. We call this model KNN_{ru}. When relying on tags, the vector contains the frequency with which a resource has been annotated with the tags. We call this model KNN_{rt}. Given r we define N as the k nearest resources drawn from the user profile and then define the relevance of r for the user u as:

$$\phi(u,r) = \sum_{s \in N} \sigma(r,s)\theta(u,s) \tag{6}$$

If a user has annotated resources similar to r then $\phi(u,r)$ will be high. Otherwise if the user does not have similar resources in his profile the score will be low.

Tag Model Similarity. Given that we may define both users and resources as a vector over the tag space, we may directly measure the cosine similarity between the two elements. We call this model *TagSim* and define its measure as:

$$\phi(u,r) = \frac{\sum_{t \in T} RT(r,t) \times UT(u,t)}{\sqrt{\sum_{t \in T} RT(r,t)^2} \times \sqrt{\sum_{t \in T} UT(u,t)^2}} \tag{7}$$

This method works under the assumption that the frequency with which a user employs a tag measures his interest in the topic described by that tag. Moreover, we assume that the frequency of the tags applied to the resource adequately describe the resource. If these two models are similar we can infer a relation between the user and resource.

4.2 Linear Weighted Hybrid Recommenders

Hybrid recommenders have a long tradition in recommendation [1]. We employ a linear weighted model which aggregates the results of several components. The constituent recommenders are freed from the burden of covering all the available informational channels and instead focus on only a few. A successful hybrid creates a synergistic blend of its constituents producing results superior to what they achieve alone.

In order to ensure that the scores for each recommendation approach are on the same scale, we normalize the scores $\phi(u,r) \in \Phi$ to 1 producing Φ'. A hybrid resource recommender will accept a user, u and query its component recommenders, $c \in C$, for each resource, r, then combine the results in the linear model:

$$\phi_h(u,r) = \sum_{c \in C} \alpha_c \phi'_c(u,r) \tag{8}$$

where α_c is the strength given to the recommender c. As usual the resources are sorted by their relevance score and the top n are returned. As additional recommenders are added to the hybrid its complexity grows. The challenge then becomes how to ascertain the correct α for each component in order to maximize the effectiveness of the hybrid.

We employ a hill climbing technique, because of its speed, popularity and simplicity. We first initialize the α vector with random positive numbers constrained such that the sum of the vector equals 1. We then randomly modify the vector and test the result to ascertain if it achieves better results. If the result is improved we accept the change, otherwise we usually reject it. We occasionally accept a change to the α vector even when it does not improve the results in order to more fully explore the α space. We continue until the vector stabilizes. To ensure we have not discovered a local maximum, we repeat the experiment several times from different starting points.

5 Experimental Results

In this section we describe the methods used to gather and process our datasets. Our testing methodology is then outlined. We separate the analysis of the experiments into a discussion on the metrics, an evaluation of the component recommenders and finally the results of the hybrids.

5.1 Datasets

We provide an extensive evaluation using data collected from two large real world tagging applications. On all datasets we perform p-core processing. Users, resources and tags are removed from the dataset in order to produce a residual dataset that guarantees each user, resource and tag occur in at least p annotations. We define an annotation to include a user, a resource, and every tag the user has applied to the resource.

Citeulike is a popular online tool used by researchers to manage and catalogue journal articles. The site owners make their dataset freely available to download. On 2/17/2009 the most recent snapshot was taken. A p-core of 5 was used. The dataset contains 2051 users, 5376 resources, 3343 tags and 105,873 annotations.

LastFM users upload their music profile, create playlists and share their musical tastes online. We selected 100 random users from the system and recursively explored the "friend" network. Only about 20% of the users had annotated a resource. Users have the option to tag songs, artists or albums. The tagging data here is limited to album annotations. This larger dataset allowed a p-core of 20. It contains 2368 users, 2350 resources, 1141 tags and 172,177 annotations.

5.2 Methodology

We employ five-fold cross validation. Each user's annotations were divided equally among five folds. Four folds were used as training data to build the component recommenders. The fifth was used to train the model parameters and ascertain the optimal weights of the component in the hybrids. The fifth fold was then discarded and we performed four fold cross validation on the remaining folds. The results were averaged over each user, then over the final four folds.

Given a user, the recommenders are evaluated on their ability to recommend resources found in the user's holdout set, R_h. Recall is a common metric for evaluating the utility of recommendation algorithms. It measures the percentage of items in the holdout set that appear in the recommendation set. Recall is a measure of completeness and is defined as: $r = |R_h \cap R_r|/|R_h|$.

Table 1. $RMSE$ and ANA metrics of Citeulike and LastFM

	RMSE(U)	RMSE(R)	RMSE(T)	ANA(U)	ANA(R)	ANA(T)
Citeulike	0.111	0.145	0.145	0.343	0.327	0.543
LastFM	0.417	0.383	0.121	0.153	0.293	0.225

Precision is another common metric for measuring the usefulness of recommendation algorithms. It measures the percentage of items in the recommendation set that appear in the holdout set. Precision measures the exactness of the recommendation algorithm and is defined as: $p = |R_h \cap R_r|/|R_r|$.

5.3 Experimental Results

Here we report the evaluation of RMSE and ANA on the datasets. Then we report the experimental results of the individual recommenders. Finally we discuss the performance and composition of the hybrids.

Information Channels. Table 1 reports the *RMSE* and *ANA* values on the two datasets. A low $RMSE(U)$ value indicates agreement between the $\boldsymbol{u^r}$ and $\boldsymbol{u^t}$ models. A high $ANA(U)$ represents that the two models produce similar neighborhoods.

In Citeulike the $RMSE(U)$ value is relatively low indicating that the $\boldsymbol{u^r}$ and $\boldsymbol{u^t}$ models are similar. $RMSE(R)$ and $RMSE(T)$ are also low in Citeulike. This agreement is likely due to the motivation of its users. Citeulike is a tagging platform for journal articles. Its users are often focused on particular topics, use domain driven tags and form tightly knit research communities. The result is a highly interrelated tapestry of users, resources and tags.

An evaluation of the ANA metrics demonstrate these trends in further detail. In each case k was selected such that the neighborhood included 10% of the elements. Other values were explored and these produced identical trends.

For Citeulike the *ANA(U)* value is 0.343 indicating that there is moderate overlap when generating neighborhoods based on $\boldsymbol{u^r}$ and $\boldsymbol{u^t}$ models. *ANA(R)* shows a similar result and *ANA(T)* demonstrates the greatest agreement. These relatively large values reinforce the notion that users in Citeulike often concentrate on their research domain and use tags relevant to that domain.

In the context of resource recommendation these observation suggest that tag information could play a meaningful role in matching users with resources. Moreover, because it reveals a strong relation between all three dimensions of the collaborative tagging application, it supports the use of an integrative approach which utilizes multiple information channels.

In contrast the $RMSE(U)$ and $RMSE(R)$ of LastFM is relatively large. This may be due to the fact that many users of this system do not store or organize their music within LastFM using tags. Instead users upload their listening habits through a process called 'scrobbling'. The system records which songs they listened to and how often they have listened to them. Since the song titles are uploaded in batches and the user is not actively engaged in the music when it is added to his profile, the annotation process becomes a burdensome additional task which is often neglected. Rather than using

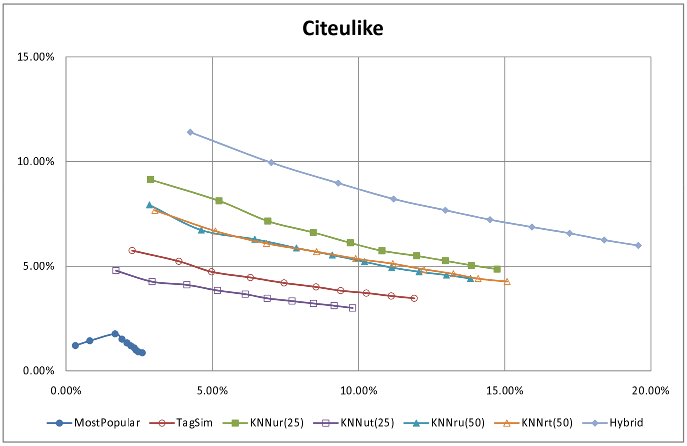


Fig. 1. The recall (x-axis) and precision (y-axis) plotted for recommendations sets of size one through ten for the component and hybrid recommenders on Citeulike

the application for organizing and exploring music through the tag space, users often employ the system to find new music and friends through the resource and user space. Visual examination of the tag space reveals that when the users do annotate albums, the tags are often overly generic, such as "rock," or not descriptive of the resource, such as "album i own." This analysis is supported by the low *ANA* values for LastFM revealing a less structured collaborative tagging application.

In terms of resource recommendation, this evidence suggests that tags would offer diminished utility in resource recommendation. Similarly, an integrative recommender utilizing multiple dimensions would not achieve a significant boost in performance over those that focus solely on the user-resource relation.

Resource Recommenders. Figure 1 reports the recall and precision of the recommendation algorithms in Citeulike. In all cases k was tuned in increments of five from five to one-hundred. The best result is reported in parenthesis. In Citeulike KNN_{ur}, the user-based collaborative filtering approach that models users as a vectors over the resource space is the strongest individual recommender. Given the nature of the recommendation task, a technique which is strongly invested in user-resource connections is predictably valuable to the hybrid.

The second most successful recommender is KNN_{rt}. The efficacy of this approach over KNN_{ru} reveals that in this dataset and for resource recommendation, tags rather than users are the better model for resources. KNN_{ut} is the worst performing individual recommender, suggesting that users are better modeled by resource than by tags. *TagSim* performs moderately well. However, all the recommenders perform far better than the baseline and are tightly grouped, suggesting that while some outperform others they are all provide some utility. The efficacy of several recommenders based on different dimensions of the data was suggested by the low *RMSE* and relatively high *ANA* values.

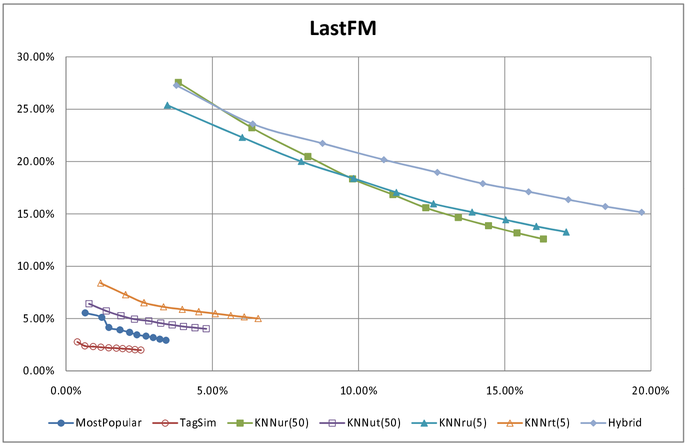


Fig. 2. The recall (x-axis) and precision (y-axis) plotted for recommendations sets of size one through ten for the component and hybrid recommenders on LastFM

Table 2. Contribution of the individual components in Citeulike and LastFM

	Pop	$TagSim$	KNN_{ur}	KNN_{ut}	KNN_{ru}	KNN_{rt}
Citeulike	0.217	0.184	0.270	0.025	0.162	0.142
LastFM	0.006	0.153	0.410	0.005	0.425	0.001

In LastFM KNN_{ur} and KNN_{ru} perform well. The results are found in Figure 2. Both these recommenders concentrate on the user-resource relation. In contrast to Citeulike, there is a wide gulf between these approaches and the remaining techniques. KNN_{ut}, KNN_{rt} and *TagSim* all perform nearly as bad or worse as the baseline recommender which simply employs the popularity of a resource. These techniques rely on tag information and their ineffectualness confirms the metric driven hypotheses generated by the high *RMSE* and low *ANA* values: the tag space is weakly developed.

Hybrid Recommenders. In both datasets the hybrid recommender surpasses the individual components. As suggested by the metrics, the Citeulike hybrid more dramatically outperforms its constituent components than the LastFM hybrid outperforms its constituents. Since the Citeulike data is more organized, particularly in the tag dimension, an integrative approach which exploits multiple dimensions of the data is able to demonstrate greater predictive power.

The individual contributions of the components are reported in Table 2 and add further weight to this analysis. In Citeulike many of the components are well represented in the hybrid. The only exception is KNN_{ut} which we earlier discovered to be the worst individual recommender. Its contribution to the hybrid was 2.5%. The remaining contributions vary from 16.2% to 21.7%. This reliance on multiple dimensions confirms our notion that the *RMSE* and *AMA* metrics describe Citeulike as a well organized application with several strong information channels that can be used for making recommendations.

The LastFM hybrid is almost exclusively composed of KNN_{ur} and KNN_{ru}. Once again this reliance a single dimension confirms the observations made by the *RMSE* and *AMA* metrics. The tag space in LastFM offers little additional information for resource recommendation.

6 Conclusions

In this paper we have proposed a linear weighted hybrid for resource recommendation in collaborative tagging applications. Our results motivate several conclusions. First, the experiments provide further evidence that not all collaborative tagging applications are equal. The different ways users interact with a system can dramatically affect the strength of its information channels. Second, metrics such as *RMSE* and *AMA* can characterize some of the differences between systems and explain how various simple recommenders would perform relative to one another. Third, an evaluation of *RMSE* and *AMA* can predict which collaborative tagging applications would benefit most from an integrative approach such as hybrid recommenders. Finally, weighted linear hybrids provide a simple and efficient means to aggregate information channels into a single framework. These hybrids can offer explanatory support and can be designed to focus on the information channels most relevant to the tagging system. These characteristics are not shared by other integrative approaches such as graph models and tensor factorization. It is not clear whether or not these other integrative models could be extended for resource recommendation.

Acknowledgments

This work was supported in part by the National Science Foundation Cyber Trust program under Grant IIS-0430303 and a grant from the Department of Education, Graduate Assistance in the Area of National Need, P200A070536.

References

1. Burke, R.: Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction 12(4), 331–370 (2002)
2. Deshpande, M., Karypis, G.: Item-Based Top-N Recommendation Algorithms. ACM Transactions on Information Systems 22(1), 143–177 (2004)
3. Gemmell, J., Ramezani, M., Schimoler, T., Christiansen, L., Mobasher, B.: A Fast Effective Multi-Channeled Tag Recommender. In: ECML/PKDD 2009 Discovery Challenge Workshop, Part of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, pp. 59–63 (2009)
4. Gemmell, J., Schimoler, T., Ramezani, M., Christiansen, L., Mobasher, B.: Improving FolkRank With Item-Based Collaborative Filtering. Recommender Systems & the Social Web (2009)
5. Gemmell, J., Shepitsen, A., Mobasher, B., Burke, R.: Personalization in Folksonomies Based on Tag Clustering. Intelligent Techniques for Web Personalization & Recommender Systems (2008)

6. Gemmell, J., Shepitsen, A., Mobasher, B., Burke, R.: Personalizing Navigation in Folksonomies Using Hierarchical Tag Clustering. In: Song, I.-Y., Eder, J., Nguyen, T.M. (eds.) DaWaK 2008. LNCS, vol. 5182, pp. 196–205. Springer, Heidelberg (2008)
7. Herlocker, J., Konstan, J., Borchers, A., Riedl, J.: An Algorithmic Framework for Performing Collaborative Filtering. In: 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, p. 237. ACM, New York (1999)
8. Hotho, A., Jaschke, R., Schmitz, C., Stumme, G.: Information Retrieval in Folksonomies: Search and ranking. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 411–426. Springer, Heidelberg (2006)
9. Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., Riedl, J.: GroupLens: Applying Collaborative Filtering to Usenet News. Communications of the ACM 40(3), 87 (1997)
10. Mathes, A.: Folksonomies-Cooperative Classification and Communication Through Shared Metadata. In: Computer Mediated Communication (Doctoral Seminar), Graduate School of Library and Information Science, University of Illinois Urbana-Champaign (December 2004)
11. Mika, P.: Ontologies are us: A unified model of social networks and semantics. Web Semantics: Science, Services and Agents on the World Wide Web 5(1), 5–15 (2007)
12. Rendle, S., Schmidt-Thieme, L.: Factor Models for Tag Recommendation in BibSonomy. In: ECML/PKDD 2008 Discovery Challenge Workshop, Part of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, pp. 235–243 (2009)
13. Rendle, S., Schmidt-Thieme, L.: Pairwise Interaction Tensor Factorization for Personalized Tag Recommendation. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining, pp. 81–90. ACM, New York (2010)
14. Salton, G., Wong, A., Yang, C.: A Vector Space Model for Automatic Indexing. Communications of the ACM 18(11), 613–620 (1975)
15. Sarwar, B., Karypis, G., Konstan, J., Reidl, J.: Item-Based Collaborative Filtering Recommendation Algorithms. In: 10th International Conference on World Wide Web, p. 295. ACM, New York (2001)
16. Shardanand, U., Maes, P.: Social Information Filtering: Algorithms for Automating Word of Mouth. In: SIGCHI Conference on Human Factors in Computing Systems, pp. 210–217. ACM Press/Addison-Wesley Publishing Co., New York (1995)
17. Shepitsen, A., Gemmell, J., Mobasher, B., Burke, R.: Personalized Recommendation in Social Tagging Systems using Hierarchical Clustering. In: ACM Conference on Recommender Systems, pp. 259–266. ACM, New York (2008)
18. Symeonidis, P., Nanopoulos, A., Manolopoulos, Y.: Tag recommendations based on tensor dimensionality reduction. In: Proceedings of the 2008 ACM conference on Recommender systems, pp. 43–50. ACM, New York (2008)
19. Symeonidis, P., Nanopoulos, A., Manolopoulos, Y.: A Unified Framework for Providing Recommendations in Social Tagging Systems Based on Ternary Semantic Analysis. IEEE Transactions on Knowledge and Data Engineering (2009)

Combining Collaborative and Content-Based Techniques for Tag Recommendation

Cataldo Musto, Fedelucio Narducci, Pasquale Lops, and Marco de Gemmis

Department of Computer Science, University of Bari "Aldo Moro", Italy
{musto,narducci,lops,degemmis}@di.uniba.it

Abstract. The explosion of collaborative platforms we are recently witnessing, such as social networks, or video and photo sharing sites, radically changed the Web dynamics and the way people use and organize information. The use of tags, keywords freely chosen by users for annotating resources, offers a new way for organizing and retrieving web resources that closely reflects the users' mental model and also allows the use of evolving vocabularies. However, since tags are handled in a purely syntactical way, the annotations provided by users generate a very sparse and noisy tag space that limits the effectiveness of tag-based approaches for complex tasks. Consequently, systems called tag recommenders recently emerged, with the purpose of speeding up the so-called tag convergence, providing users with the most suitable tags for the resource to be annotated.

This paper presents a tag recommender system called STaR (Social Tag Recommender), which extends the social approach presented in a previous work [14] with a content-based approach able to extract tags directly from the textual content of HTML pages.

Results of experiments carried out on a large dataset gathered from Bibsonomy, show that the use of content-based techniques improves the predictive accuracy of the tag recommender.

Keywords: Recommender Systems, Web 2.0, Collaborative Tagging Systems, Folksonomies, Tag Recommender.

1 Introduction

The recent explosion of collaborative platforms radically changed the nature of the services offered by the Web. For many years Internet users have been relegated to a passive role in the Web dynamics: they were only viewers of the information created by other users (e.g. web site administrators) and they were not able to contribute in any way to enrich the available content. Nowadays this strong dichotomy has been replaced by a new and more democratic vision, in which users are more and more involved, because they are able to:

1. produce new content (the so-called User Generated Content);
2. enrich already available content with novel metadata.

F. Buccafurri and G. Semeraro (Eds.): EC-Web 2010, LNBIP 61, pp. 13–23, 2010.

In platforms like YouTube, Flickr or Wikipedia users contribute in the first way, by sharing resources (new videos, photos, ...) that other users can enjoy, while in the collaborative tagging systems, the contribution of the community falls into the second category.

The idea behind the concept of tagging is simple: a user enjoys a resource (an image, a web site, etc.) and, according to her mental model, identifies those terms that better describe the information conveyed by that resource. The same resource can be annotated by several users: some of them will reuse the tags already assigned to that resource, while some others will adopt new tags. This process allows building a socially-constructed classification schema, called folksonomy [18]. The inceptive idea behind folksonomies is that the more a tag is used by the community to annotate the target resource, the more is the likelihood the tag correctly describes its content.

Recently the use of folksonomies gained more attention because of their simplicity: using tags, users can freely model the information without the constraints of a predefined lexicon or hierarchy [12]. However, the simplicity of the approach has also an important drawback: the information managed by folksonomies is modeled in a simple syntactical way. Therefore, as stated by Golder et. al. [8], collaborative tagging systems suffer from the classic problems of Information Retrieval (IR) systems like polysemy, synonymy and level variation.

The *polysemy* refers to situations where tags can have multiple meanings: for example a resource tagged with the term *turkey* could indicate a news in an online newspaper about politics or a recipe for Thanksgiving Day.

When multiple tags share the same meaning we refer to it as *synonymy.* In collaborative tagging systems we can also have simple morphological variations, for example we can find *blog*, *blogs*, *web log*, to identify a common blog, but also tags semantically similar such as resources tagged with *arts* versus *cultural heritage.*

Finally, the *level variation* problem refers to the phenomenon of tagging at different level of abstraction: some people can annotate a web page containing a recipe for roast turkey with the tag *roastturkey*, but also with a simple *recipe.*

These drawbacks hinder the use of folksonomies for tasks more complex than the simple browsing of resources. In order to avoid these problems, in the last years many tools have been developed to facilitate the user in the task of tagging, by also speeding up the tag convergence [6]: these systems are known as tag recommenders. These systems work in a very simple way:

1. a user posts a resource;
2. depending on the approach, the tag recommender analyzes some information related to the resource (usually metadata);
3. the tag recommender processes this information and produces a list of recommended tags;
4. the user freely chooses the most appropriate tags to annotate the resource.

This paper presents the tag recommender STaR. When developing the model, we tried to point out two concepts:

- resources with similar content should be annotated with similar tags;
- a tag recommender needs to take into account the previous tagging activity of users, increasing the weight of the tags already used to annotate similar resources.

We applied STaR to the task of recommending tags for bookmarks. In this work we enriched the approach we proposed in [14] by integrating some heuristics for extracting tags from the HTML content of web pages to be annotated. This could allow producing recommendations even when no similar bookmarks are available, and the process of extracting tags from similar resources inevitably fails.

The paper is organized as follows. Section 2 analyzes related work, while Section 3 depicts the general system architecture and the recommendation approach implemented in STaR. Details about the experimental evaluation are given in Section 4. Finally, conclusions and future works are drawn in the last section.

2 Related Work

Usually tag recommenders are broadly divided into three classes: *content-based*, *collaborative*, and *graph-based* approaches.

A content-based tag recommender exploits textual content by adopting Information Retrieval-related techniques [1] in order to extract relevant terms (unigrams or bigrams) to label a resource. Brooks et. al [5], for example, developed a tag recommender system that automatically suggests tags for a blog post by extracting the top three terms exploiting the TF/IDF scoring [15]. Another work exploiting the TF/IDF measure is KEA [19]. This system applies a supervised Bayesian classifier able to extract relevant phrases from textual content.

Other systems exploit ontologies for recommending tags: in [2], terms are extracted from the document and subsequently, surfing the ontology, more abstract and conceptual tags are suggested.

The collaborative approach for tag recommendation, instead, presents some analogies with collaborative filtering methods [4]. These systems compute the relevance of a tag to be suggested by exploiting the community behavior. In the model proposed by Mishne and implemented in AutoTag [13], the system suggests tags based on the other tags associated with similar posts in a given collection. The recommendation process is performed in three steps: first, the tool finds similar posts and extracts their tags. All the tags are then merged, building a general folksonomy that is filtered and re-ranked. The top-ranked tags are suggested to the user, who selects the most appropriate ones to attach to the post. In [7] the authors propose an adaptation of the classical K-nearest neighbor approach to create a set of recommended tags. The neighbors are defined looking for users tagging the same resources with the same tags. In this way tags used by similar users are boosted in the set of recommended tags.

The problem of tag recommendation through graph-based approaches has been firstly addressed by Jäschke et al. in [9]. They compared some recommendation techniques including collaborative filtering, PageRank and FolkRank. The key idea behind FolkRank algorithm is that a resource which is tagged by important tags from important users becomes important itself. The same concept holds for tags and users, thus the approach uses a graph whose vertexes mutually reinforce themselves by spreading their weights. The evaluation showed that FolkRank outperforms other approaches. Schmitz et al. [16] proposed association rule mining as a technique that might be useful in the tag recommendation process.

In literature we can find also some hybrid methods integrating two or more approaches (mainly, content and collaborative ones) in order to reduce their typical drawbacks and point out their qualities. In [11] the authors presents a tag recommender that suggests tags based on the resource content, resource related tags and user profile tags. The content exploited by the recommender is the title of the resource (and the URL, if available). Then, the system extends the set of words extracted from the title with tags related to the title as well as with tags occurring in the user and resource profile. The user profile is composed by all the tags used by that specific user to label resources. Instead, the resource profile groups tags assigned to a resource by other users. Another similar approach is presented in [10] where the authors exploit three kinds of information sources: the description of the resources, their folksonomies, and the tags previously used by the same person. In order to remove inappropriate candidate tags, a filtering method and a weighting scheme for assigning different importance to sources are applied.

3 STaR: A Social Tag Recommender System

A collaborative tagging system is a platform composed of users, resources and tags that allows users to freely assign tags to resources, while the tag recommendation task for a given user and a specific resource can be described as the generation of a set of tags according to some relevance model. In our approach these tags are generated from a ranked set of *candidate tags* from which the top n elements are suggested to the user.

STaR is a tag recommender system, developed at the University of Bari. The inceptive idea behind STaR is to improve the model implemented in systems like AutoTag [13]. Although we agree with the idea that resources with similar content could be annotated with similar tags, in our opinion the approach proposed in [13] presents three important drawbacks:

1. The tag re-ranking formula simply performs a sum of the occurrences of each tag among all the folksonomies, without considering the similarity with the resource to be tagged. In this way tags often used to annotate resources with a low similarity level could be ranked first.
2. The proposed model does not take into account the previous tagging activity performed by users. If two users bookmarked the same resource, they will receive the same suggestions since the folksonomies built from similar resources are the same.
3. When there are not similar resources available for suggesting similar tags, the recommendation task fails.

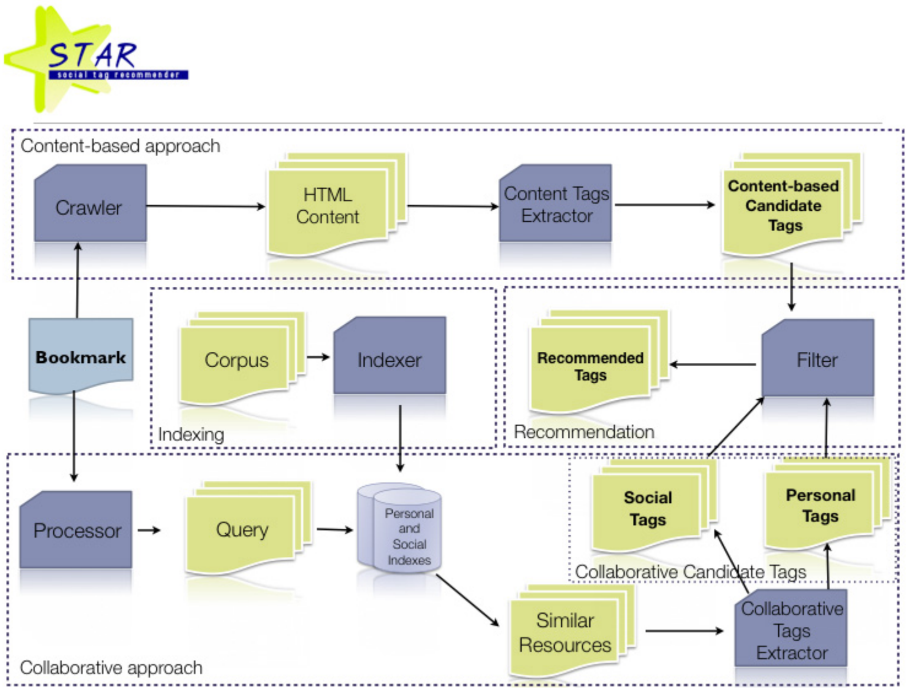


Fig. 1. Architecture of STaR

We will try to overcome these drawbacks by proposing an approach based on the analysis of similar resources, also capable of weighting more the tags already selected by the user during her previous tagging activity. Furthermore, we also integrated some heuristics to directly extract tags from the HTML content of a Web page.

Figure 1 shows the general architecture of STaR. In our system we integrated two different approaches to recommend tags: the content-based and the collaborative one. The collaborative approach needs a preprocessing step that, given a corpus of documents, produces a Social and a Personal indexes that are queried in order to produce a set of *Collaborative Candidate Tags*.

In the same way a set of *Content-based Candidate Tags* is produced by crawling the HTML content of the web page. Finally, the recommendation step merges the results obtained by the aforementioned approaches.

In the next section the recommendation model implemented in STaR is thoroughly analyzed.

3.1 Indexing Step

Given a collection of resources already annotated (tagged), called *corpus*, a preprocessing step is performed by the *Indexer* module, which exploits Apache

Lucene[1] to perform the indexing step. The corpus is composed by *bookmarks*, so we indexed the title of the web page, the description provided by users and the tags used to annotate it. We build an index for each user, called *Personal Index*, that stores the information on her previously tagged resources and an index for the whole community, called *Social Index*, that stores the information about all the resources previously tagged by the community.

3.2 Tag Extraction Step

STaR implements two different approaches to recommend tags for a new bookmark: a *collaborative* approach, and a *content-based* one.

- **Collaborative approach**
  After the Indexing step, STaR retrieves the most similar bookmarks (to that to be annotated) and collects the most relevant tags used by the community to annotate them.

  More specifically, given a user u and a resource r to be annotated, STaR returns the resources whose similarity with r is greater or equal than a threshold β whose value is dependent from the corpus used for the evaluation. To perform this task, the *Processor* gets data about both the user (the tags she uses more) and the resource (metadata like the title of the web page) and submits a query against the *Social Index*. If the user is recognized by the system, since it has previously tagged some other resources, the same query is submitted against her own *Personal Index*, as well. The *Collaborative Tags Extractor* builds a set of *Collaborative Candidate Tags* by extracting tags assigned to the most similar resources retrieved from the indexes. For each tag, a score is computed by weighting the similarity score returned by Apache Lucene with the normalized occurrence of the tag. This list is finally ranked and the top n tags are suggested to the user.

  In order to improve the performance of the Lucene Querying Engine the original Lucene Scoring function has been replaced with an Okapi BM25 implementation[2]. In [14] we showed that the BM25 integration improves the overall accuracy of STaR. Further details on the collaborative approach implemented in STaR are provided in [14].
- **Content-based approach**
  The collaborative approach fails when STaR does not retrieve any bookmark similar to the one the user should annotate.

  To tackle this problem, we implemented a content-based approach for extracting tags by analyzing the HTML source of a web page. The idea behind this strategy is to identify a set of candidate tags, and to choose the more promising ones by analyzing the content of the web page. Starting from the URL of the new bookmark posted by the user, the *Crawler* retrieves the web page and its HTML source. Next, the *Content Tags Extractor* extracts from the URL the string representing the domain name (for example from

[1] http://lucene.apache.org

[2] http://nlp.uned.es/%7Ejperezi/Lucene-BM25/

the URL *www.wikipedia.it*, we extract *wikipedia*). This represents the first element in the set of *Content-based Candidate Tags*. Afterwards, we extract other candidate tags by analyzing the content stored in the HTML tags $\langle title \rangle$ and $\langle meta \rangle$. As regards the tag $\langle meta \rangle$, we extract the value of the attributes *keywords* and *description*, because in our opinion they are the most significant. This list is filtered by deleting stopwords and verbs.

When the set of *Content-based Candidate Tags* is built, the algorithm assigns a score to each tag, according to several heuristics. First, the number of occurrences of each candidate tag in the HTML source is taken into account (values are normalized in a range between 0 and 1). Then, a different weight is assigned to each source according to the following values (this is a heuristics that needs to be evaluated):

- URL: 0.2
- TITLE: 0.3
- META KEYWORDS: 0.25
- META DESCRIPTION: 0.25

The score assigned to each tag is the weighted sum of the normalized occurrence of the tag in each source multiplied for the weight of the same source.
More formally, given $S = \{s_1, ..., s_n\}$ the set of sources, $Cand = \bigcup_{s_i \in S} Cand_s$, where $Cand_s$ is the set of candidate tags coming from the source s, the score of the tag t is computed by the following formula:

$$score(t) = \sum_{s_i \in S} w_{s_i} \cdot n_t \tag{1}$$

where s_i identifies the source of the candidate tag, w_{s_i} is the weight of the source s_i, and n_t is the normalized occurrence of the tag t in the HTML source ($n_t = 0$ if the tag t does not occur in the source s_i).

3.3 Tag Recommendation Step

After the tag extraction step, two sets of tags, *Content-based Candidate Tags* and *Collaborative Candidate Tags* are available. The set *Collaborative Candidate Tags* contains the set *Social Tags* and *Personal Tags*.

The *Filter* component removes from each set those tags whose score is under a certain threshold, and decides which set to exploit to provide recommendations. The *Filter* can suggest just the tags belonging to a specific set, or it can implement a specific strategy to combine different sets.

The strategy implemented in this work is a *cascade* one: the *Filter* starts from a unique set of tags (*Social*, *Personal*, or *Content-based*) and, if that set is empty, it uses the others. For example, it starts with the *Personal Tags* set, but if the user is new in the community, that set is empty; thus, *Social Tags* set is used. If this set is empty too, for example because there are not similar resources to the one to be annotated, the last chance is to exploit tags extracted by the content-based approach.

The *Filter* can implement other strategies: for example a merge of the three sets of tags can be defined. Possible strategies will be investigated in future works.

4 Experimental Evaluation

We designed two different experimental sessions to evaluate the performance of STaR. In the first session we evaluated the predictive accuracy of the single sets of tags, *Social Tags*, *Personal Tags* and *Content-based Tags*, while in the second session we investigated whether the combination of these sets helps STaR to produce better recommendations.

4.1 Description of the Dataset

In the experiment we adopted the dataset used for the Content-based recommendation task of the ECML-PKDD 2009 Discovery Challenge[3]. We just used the *263,004 bookmark posts* submitted by 3,617 different users. For each of the 235,328 different URLs were also provided some textual metadata (such as the title of the resource, the description and so on).

We evaluated STaR by comparing the real tags (namely, the tags a user adopts to annotate a new resource) with the suggested ones. The accuracy was finally computed using classical Information Retrieval metrics, such as Precision (Pr), Recall (Re) and F1-Measure (F1) [17].

4.2 Experimental Session

In the first experiment we evaluated the predictive accuracy of the single sets of tags produced by the *Content-Based Tags Extractor* and the *Collaborative Tags Extractor*. In this experiment the threshold β was set equal to 0.20. Results are presented in Table 1.

Table 1. Predictive accuracy of the single sets of tags

Recommendation set	Pr	Re	F1
Personal Tags	11.20	8.34	9.58
Social Tags	13.80	11.28	12.41
Content-based Tags	14.36	17.90	15.90

The effectiveness of the collaborative approach (Personal Tags and Social Tags) is strictly dependent on the amount of the available resources already annotated. Since the approach based on Personal Tags builds the set of candidates tags on the ground of the resources the user annotated in the past, we observed a significant loss in recall with respect to the content-based approach (-9.56). In

[3] http://www.kde.cs.uni-kassel.de/ws/dc09/

the approach based on Social Tags the recommendation model analyzes the set of resources already tagged by all the users in the community, and this reduces the loss of recall with respect to the content-based approach (-6.62).

As regards the overall accuracy of the content-based approach in terms of F1, it outperforms the collaborative one (+3.49 for Social Tags and +6.32 for Personal Tags). This can be explained by observing that the content-based model is totally independent from the number of the resources already annotated in the corpus, and also because it does not suffer of the cold-start problem. The content-based method based on the extraction of tags from the HTML content of the web page obtained a Precision comparable to that of Social Tags, but with a significant improvement in recall (+6.62).

To sum up, the main outcome of the first experiment is that the content-based approach proposed in the paper is a valuable strategy for improving the performance of the tag recommender.

In the second experiment we investigated whether some strategies for combining collaborative and content-based approaches are useful to improve the predictive accuracy of STaR.

We run the content-based (respectively, the collaborative) tag extraction algorithm when the collaborative (respectively, the content-based) one is not able to return any tag. As regards the collaborative model, this happens when no similar documents are retrieved by querying the corpus containing previously annotated resources, while in the content-based model, this happens when the HTML source does not include any meta-tag or when the words contained in the set of *Content-based Candidate Tags* do not occur in the HTML source code. Results are presented in Table 2.

Table 2. Predictive accuracy of different strategies for combining collaborative and content-based approaches

Configuration	**Pr**	**Re**	**F1**
Personal+Social Tags *(baseline)*	14.28	16.53	15.32
Personal+Content-based Tags	13.64	15.51	14.52
Content-based+Personal Tags	17.79	15.07	16.32
Social+Content-based Tags	11.50	13.96	12.61
Personal+Social+Content-based Tags	14.51	16.68	15.52
Content-based+Personal+Social Tags	**19.21**	**16.12**	**17.53**

The Personal+Social Tags approach is used as baseline since it represents the configuration of STaR used to participate to the ECML-PKDD 2009 Discovery Challenge [14]. The first outcome of the experiment is that the combination of techniques for recommending tags is a valuable strategy. Indeed, the baseline outperforms the results obtained using just Personal Tags (F1: +5.74) or Social Tags (F1: +2.91). It is also worth to note that the order different techniques are used matters. The configuration of the experiment adopting the content-based tags first and the set of personal tags afterwards outperforms the configuration adopting personal tags first and content-based tags subsequently (F1: +1.8). This result is coherent with the last two results reported Table 2 (F1: +2.01). Another interesting result is that using configuration exploiting content-based tags as the first strategy gives the best results. This is in line with results report in Table 1, in which the best performance of STaR was obtained by adopting the set of content-based tags.

5 Conclusions and Future Work

In this paper we presented STaR, a tag recommender system integrating a social approach with a content-based one able to extract tags by analyzing the textual content of resources to be annotated. The inceptive idea behind our work was to discover similarity among resources in order to exploit communities and user tagging behavior. In this way our recommender system was able to suggest tags for users and items still not stored in the training set. We also enriched this collaborative approach by extracting tags from HTML source code by adopting simple heuristics: we showed that this improves the predictive accuracy of the recommendation model.

We are planning to run several experiments related to the content-based extraction algorithm in order to tune the several systems parameters. Finally, we will investigate about other possible strategies for combining and merging tags coming from the different sources: for example a simple linear combination of the relevance scores could be adopted.

Furthermore, since tags usually suffer of typical Information Retrieval problem (polysemy, synonymy, etc.) we will try to establish if the integration of Word Sense Disambiguation [3] tools or a semantic representation of documents could improve the performance of the tag recommender.

References

1. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley, Reading (1999)
2. Baruzzo, A., Dattolo, A., Pudota, N., Tasso, C.: Recommending new tags using domain-ontologies. In: Web Intelligence/IAT Workshops, pp. 409–412 (2009)
3. Basile, P., Degemmis, M., Gentile, A.L., Lops, P., Semeraro, G.: UNIBA: JIGSAW algorithm for Word Sense Disambiguation. In: Proceedings of the 4th ACL International Workshop on Semantic Evaluations (SemEval-2007), Prague, Czech Republic, June 23-24, 2007, pp. 398–401. Association for Computational Linguistics (2007)
4. Billsus, D., Pazzani, M.J.: Learning collaborative information filters. In: Proceeding of the 15th International Conference on Machine Learning, pp. 46–54. Morgan Kaufmann, San Francisco (1998)
5. Brooks, C.H., Montanez, N.: Improved annotation of the blogosphere via autotagging and hierarchical clustering. In: WWW '06: Proceedings of the 15th International Conference on World Wide Web, pp. 625–632. ACM, New York (2006)

6. Cattuto, C., Schmitz, C., Baldassarri, A., Servedio, V.D.P., Loreto, V., Hotho, A., Grahl, M., Stumme, G.: Network properties of folksonomies. AI Communications 20(4), 245–262 (2007)
7. Gemmell, J., Schimoler, T., Ramezani, M., Mobasher, B.: Adapting k-nearest neighbor for tag recommendation in folksonomies. In: ITWP (2009)
8. Golder, S., Huberman, B.A.: The Structure of Collaborative Tagging Systems. Journal of Information Science 32(2), 198–208 (2006)
9. Jäschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., Stumme, G.: Tag recommendations in folksonomies. In: Hinneburg, A. (ed.) Workshop Proceedings of Lernen - Wissensentdeckung - Adaptivität (LWA 2007), September 2007, pp. 13–20 (2007)
10. Ju, S., Hwang, K.: A weighting scheme for tag recommendation in social bookmarking systems. In: Eisterlehner, F., Hotho, A., Jäschke, R. (eds.) ECML PKDD Discovery Challenge 2009 (DC'09), CEUR Workshop Proceedings, September 2009, vol. 497, pp. 109–118 (2009)
11. Lipczak, M., Hu, Y., Kollet, Y., Milios, E.: Tag sources for recommendation in collaborative tagging systems. In: Eisterlehner, F., Hotho, A., Jäschke, R. (eds.) ECML PKDD Discovery Challenge 2009 (DC'09), CEUR Workshop Proceedings, September 2009, vol. 497, pp. 157–172 (2009)
12. Mathes, A.: Folksonomies - cooperative classification and communication through shared metadata (December 2004), http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html
13. Mishne, G.: Autotag: a collaborative approach to automated tag assignment for weblog posts. In: WWW '06: Proceedings of the 15th International Conference on World Wide Web, pp. 953–954. ACM, New York (2006)
14. Musto, C., Narducci, F., de Gemmis, M., Lops, P., Semeraro, G.: Star: a social tag recommender system. In: Eisterlehner, F., Hotho, A., Jäschke, R. (eds.) ECML PKDD Discovery Challenge 2009 (DC'09), CEUR Workshop Proceedings, Bled, Slovenia, vol. 497, pp. 215–227 (2009)
15. Salton, G.: Automatic Text Processing. Addison-Wesley, Reading (1989)
16. Schmitz, C., Hotho, A., Jäschke, R., Stumme, G.: Mining association rules in folksonomies. In: Data Science and Classification (Proc. IFCS 2006 Conference), Studies in Classification, Data Analysis, and Knowledge Organization, July 2006, pp. 261–270. Springer, Heidelberg (2006)
17. Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys 34(1), 1–47 (2002)
18. Vander Wal, T.: Folksonomy coinage and definition. Website (Februar 2007), http://vanderwal.net/folksonomy.html
19. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann Publishers, San Francisco (1999)

Category Recommendation in User Specified Structure

Ying Zhou, Xiaochen Huang, and Shirley Priyanka Lee

School of Information Technologies
The University of Sydney. NSW 2006, Australia
ying.zhou@sydney.edu.au, xhua2034@uni.sydney.edu.au,
slee7800@uni.sydney.edu.au

Abstract. Tagging has become a main tool for Internet users to describe and advertise various web resources. The relatively flat structure of the tag space poses lots of challenges in tag based query engines. Many data-centric algorithms have been proposed to discover structures from the flat tag space and to improve query results. At the same time, lots of social networking sites start to provide mechanisms allowing users to specify simple hierarchical structures. The group concept in Flickr is a good example of such user specified structure. Users can create a group with predefined themes. Other users can add their resources to several related groups voluntarily or by invitation. These groups are analogue to categories in a cataloguing system. This user specified structure would ideally improve the precision of tag based query. However, categories can be created by any user and public categories like groups are open for users to add resources in. More often than not, a simple category title does not give enough information on its content. In this paper we propose two algorithms, traditional IR cosine similarity approach and frequent pattern matching approach, to recommend categories to a given resource. We evaluate our algorithms using groups and photos from Flickr. Both algorithms achieve promising results in terms of precision in general. We also analyse strength and weakness of the two algorithms with respect to features of test data. We believe such recommendation mechanism is an important complement to any user specified hierarchical structure.

Keywords: user specified structure; pattern matching; cosine similarity; social tagging system.

1 Introduction

Cataloguing and classification are integral components of any information system dealing with large amount of data. Typically a taxonomy is used to exclusively classify an item to an unambiguous category which is in turn within a more general one [3]. Some examples of taxonomies include the Dewey decimal classification for libraries, and computer file systems for organizing electronic files [6]. In addition to the relatively rigid taxonomic classification a more relaxed option is to annotate content with keywords (called *tags*) for future navigation,

F. Buccafurri and G. Semeraro (Eds.): EC-Web 2010, LNBIP 61, pp. 24–35, 2010.

filtering and search. For example, Figure 1 shows a photograph of a cat from South Africa in a file system classification. The same photograph as tagged in Figure 2 allows more attributes of the photograph to be categorised. In web 2.0 era, many user-uploaded web resources are described by a list of keywords supplied either by the resource owner or other users. Such web platforms for users to manage and share resources are known as social tagging systems.

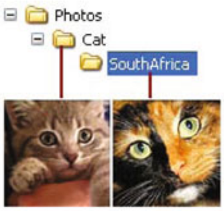


Fig. 1. Typical File System Classification

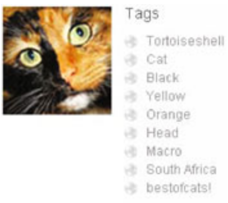


Fig. 2. Folksonomy example

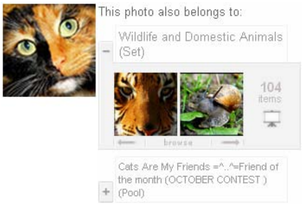

This photo also belongs to:
Wildlife and Domestic Animals (Set)
104 items
browse
Cats Are My Friends =^..^=Friend of the month (OCTOBER CONTEST) (Pool)

Fig. 3. Sets and Groups Example

In social tagging systems, users are free to use any text as tags to describe an item. They have recently gained immense popularity. A good example of such system is Flickr [2], the online photo sharing website that allows users to upload, share, organise and view photos. Since its launch in 2004, Flickr has become one of the largest photo sharing communities on the Internet. As of October 2009, Flickr claims to host over 4 billion photos, indexed by over 20 million unique tags. As users contribute more tags, a form of classification scheme takes shape. Such scheme is now commonly referred to as 'folksonomy' [6]. Different from taxonomy, folksonomy is flat and its categories are ambiguous most of the time.

As tags contain high density information, it is likely that tag based query always return results from various perspectives. Clustering would help users to locate desirable results efficiently. In addition to automatic and unsupervised clustering feature, many social tagging systems provide tools for users to explicitly classify and organize resources. For instance, Flickr provides both user-level and site-level classification tools. A user can organize his/her photos into overlapping collections and sets. A user can also create groups and invite other users to join the group and put related photos in group pool. All those activities produce a hierarchical structure on top of the relatively flat folksonomy. Figure 3 shows the same cat photo belongs to a set called "Wild life and domestic animals" defined by the owner. It also belongs to many group pools such as "Cat are my friends", "Best of Cats". Group titles, set titles and photo tags are categories in a user specified hierarchical structure.

User specified category has been a new venue for users to organize and promote their resources. Large number of categories have been created in sites like Flickr. Given a web resource, there may be many categories that are related with it. Different from categories in a taxonomy system, the meaning of user defined categories are derived from the resources associated with it. As users are allowed to put the same resource in various categories, categories may overlap in terms of both meanings and contents. A simple keyword query may result in thousands of groups in Flickr returned, and the top few ones may not have any semantic relation. It is not easy for a user to find out whether a photo is more related with "apple lover" or "an apple a day", or some other similar named categories. To make effective use of such user specified structure, we need models to summarize and recommend categories to users based on what are currently in the category.

In this paper we propose several algorithms to rank and recommend user defined categories to a given web resource. The contributions of the paper are as follows:

- We propose and implement two recommendation algorithms, traditional IR cosine similarity approach and frequent pattern matching approach.
- We carefully design a test data collection of interrelated categories with different features, such as large and small categories, focused and diverse categories from Flickr.
- We illustrate that both algorithms achieve promising recommendation precision, while the frequent pattern matching approach is more efficient than cosine similarity approach. In addition, we show and prove that category features have different impacts on result precision.

2 Literature Review

Liu et al. [4] proposed an unsupervised tag ranking scheme, which ranks the tags associated with a given photo according to their relevancy. Groups are then recommended based on the top 3 tags of a photo in the ranked tag list. Group recommendation is presented as a possible application area of the tag ranking scheme in this paper.

Chen et al. [1] designed and implemented a group recommendation system called "Sheep Dog" which uses a Support Vector Machine classifier to classify a photo to a pre-defined list of concepts. Photo query and group query APIs provided by Flickr are used to obtain training dataset for the SVM classifier. Once the photos concepts are identified, the top 'n' concept names are taken as keywords to recommend groups. This approach relies mainly on visual features of the photo. It also rely on a manually selected "popular" concepts such as "animals", "architecture", "dog", "cat", "snake", "portrait", etc. This kind of supervised learning with a pre-defined category list is not suitable for dynamic and open system as Flickr or other social tagging system.

Negoescu et al. [7] conducted an extensive study on Flickr groups and behaviour of users with respect to sharing photos in groups. They also apply probabilistic latent semantic analysis (PLSA) to model groups by latent topics. Each group is considered as a document, and the content of the document are the tags associated with the photos inside the group. PLSA derives the implicit topics of the group documents collection by computing two important probabilities: the probability a tag represents a topic and the probability a group contains a topic.

In keyword based group search, the first probability is used to find relevant topics of query terms while the second one is used to find top groups containing those topics. The topic based approach may provide better ranking than the simple keyword search against group title or description in certain cases where the query contains some popular terms. However it might not be effective for queries with less popular terms as those terms would have small probabilities with all topics. In addition, the optimal value of the model parameter, number of topics, can be different for different datasets.

3 Category Recommendation Algorithms

The problem we try to solve can be described informally as: given an annotated resource and a collection of categories, rank the categories based on how closely its semantic matches the resource. We propose two methods to solve this problem. The Cosine Similarity Model (CSM) takes traditional IR approach. It tackles the problem at a macro level by treating each category as a document consisting of tags from all resources inside the category. It captures the overall category feature without considering features of individual resources inside the category. We also address the problem at a micro level from data mining perspective using Frequent Pattern Matching (FPM) method. This is done through examining tag co-occurring patterns of individual resource inside the category and evaluating an external resource's similarity with the category based on pattern matching.

3.1 Cosine Similarity Model

Vector space model places an important role in information retrieval applications. The basic idea involves three elements: term space, document vector and query vector. In this model, a document is treated as a bag of words and all distinct words in the document collection form a term space. Documents and query are then represented as vectors in the term space. For each query, a document is assigned a score calculated as the distance between the document vector and the query vector. These are essential in locating and ranking relevant documents for a given query.

Our cosine similarity model takes this approach by constructing a term space from distinct tags appearing in all categories. A category is mapped as a document; the tag set of a given resource is mapped as a query. Such mapping enables us to represent both categories and the query as vectors in the term space. The weight of a tag t in a query q is denoted as $w_{q,t}$; while the weight of a tag in a category c is denoted as $w_{c,t}$. Different heuristics have been proposed in literature to quantify the similarity between a query and a document. These include various ways of computing $w_{q,t}$, $w_{c,t}$ and the combining similarity function [8,5]. In this study, we adopt the normalized formulation as term frequency component and do not include the *idf* component in $w_{c,t}$. The exclusion of *idf* has practical implication as both the category itself and the category collection are not stable at all. The computation of $w_{q,t}$ follows the binary match formulation since no duplicate tags can appear in the same query photo in Flickr. Cosine measure is used to compute the distance of category vector $\overrightarrow{V}(c)$ and query vector$\overrightarrow{V}(q)$. Let $f_{d,c}$ denotes frequency of a term t in category c, we have:

$$w_{c,t} = log_{10} f_{t,c}, w_{q,t} = \begin{cases} 1 \text{ if } t \in q \\ 0 \; otherwise \end{cases} \tag{1}$$

$$Sim(q,c) = \frac{\sum_{t \in q} w_{q,t} w_{c,t}}{\sqrt[2]{\sum_{t \in q} w_{q,t}^2} \sqrt[2]{\sum w_{c,t}^2}} = \frac{\sum_{t \in q} w_{c,t}}{\sqrt[2]{|q|} \sqrt[2]{\sum w_{c,t}^2}} \tag{2}$$

3.2 Frequent Pattern Matching Model

Treating category as a bag of words misses the natural grouping structure of tags formed by resources inside the category. Even in similar bags of words, certain tags may co-occur more frequently than others. Ignoring such patterns can cause problems. Figure 4 illustrates an example where cosine similarity model will assign same scores to the two very different categories. In this example, category *apple tree* and *New York City* both share two tags with the query photo. The cosine similarity between the query photo and the two categories are the same. However, it is obvious that category *Apple tree* is more closely related with the query photo. The differences between categories are reflected in the tag co-occurrence patterns. The query tags *apple* and *tree* co-occur repeatedly in category *apple tree*, while all query tags do not co-occur in category *New York City*.

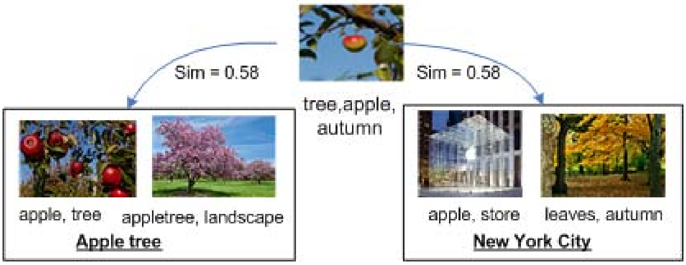


Fig. 4. Problems of Cosine Similarity Model

The simple example shows that co-occurrence patterns of tags inside a category are good indicators on how closely a query resource and a category match. Our frequent pattern matching model is developed based on this observation. In the preprocessing stage, frequent co-occurrence patterns are extracted from each category. An inverted tag-pattern index is set up for efficient pattern matching in the query stage. When a query resource is given, matching pattern is found for each related category and an FP score is computed based on matching portion and pattern support.

The tag pattern of a given resource would match partially with many frequent patterns of a category. It is reasonable to assume that large overlapping indicates close match between two patterns. Hence, patterns inside a category with the longest overlapping portion are selected as candidate patterns. Among all candidate patterns, the pattern with the highest support becomes the final matching pattern. Matching patterns from different categories may vary in length. Longer frequent pattern normally has lower support than the shorter frequent pattern has. To mitigate such effect, the FP score of each category is thus computed as overlapping count adjusted by the support percentage of matching pattern. We have

$$FpScore(q, c) = |mp \cap q| + SupportPercent(mp) \tag{3}$$

Table 1 shows the process of computing FpScore based on inverted tag-pattern index. We use the same query and groups that are shown in Figure 4 as an example. We assume different support percentages for the involved patterns. A query is issued for each tag in the query to find related (pattern, group, support) tuples. After scanning the inverted-pattern list for all tags in the query. We get the final score for group 1 as 2.5 and group 2 as 1.5.

Table 1. FPScore computation

tag(s)	pattern tuple	FPScore computation
apple, tree →	([***apple, tree***], g1, 0.5)	g1 ← 2.5
apple	([***apple,*** *store*], g2, 0.5)	g2 ← 1.5
autumn →	([*leaves,* ***autumn***], g2, 0.3)	

4 Experiment

The experiment is carried out on a PC equipped with quad CPU and 4G memory. MySQL 5.1 is used as local storage for test data and results. Both algorithms are implemented in Java language.

4.1 Experimental Data

Our test data collection includes several user specified categories downloaded from Flickr using APIs provided. Each category is represented by a group in Flickr. All photos as well as their tags in group pools are downloaded. The photos without any tag information are discarded. The data collection took place on early January, 2010. In total there are over 1.8 million photos included in the test collection. They belong to over 95.7 thousand users and 38 groups. Around 1 million unique tags are used to tag the photos in the collection.

We select groups with various features around a few predefined themes. The main goal is to create a sample of groups that overlaps in textual content such as titles and tags but not in actual photo content. Such sample can test an algorithm's discriminative power. The themes include *apple*, *portrait*, *animal* and so on. Within each theme, we include general and narrow concepts. For instance, we have a group representing *apple* in general, a group focusing on wide range of *Apple products* and several groups concentrating on *MacBook*, *iPod* and so on. We also include groups with subjective themes and groups about different cameras. The sizes of these groups vary from a few hundred photos to over 700 thousand photos.

Table 2. Test Data Collection Summary

	feature	**count/percent**
group size as number of photos	over 100K	3 groups
	between 10K and 100K	10 groups
	between 1K and 10K	19 groups
	less than 1K	6 groups
tag-group distribution (overall)	maximum sharing: all groups	8 tags
	tags shared by least 2 groups	23.6%
tag-group distribution (top 5)	unique tags	110
	tags shared by least 2 groups	26.4%
	groups having unique top tags	4 groups
	group sharing all top tags	4 groups
tag-group distribution (top 10)	unique tags	203
	tags shared by at least 2 groups	33.5%
	groups sharing all top tags	7 groups
	groups sharing half or more top tags	29 groups
photo-group distribution	maximum sharing: 6 groups	6 photos
	appear only in 1 group	96.5%

Table 2 shows the summary of group sizes and tag, photo distribution among groups. There are 8 tags appear in all groups; overall 248.9K tags appear in at least two groups. We also analyse frequent tags in the groups. We extract top 5 and top 10 tags from each group and examine the tag-group and group-tag distribution. Quite a few groups share all their top tags with other groups. 29 groups share half or more top 10 tags with other groups. These show significant tag overlapping within our test data. The photo sharing among groups are relatively small. 96.5% of the photos appear only in one group. Such distribution satisfies our main goal for the group sample as test data.

We run two experiments on the test data. The first experiment involves a small independent test suite and human evaluators are used to provide relevance judgment. A second experiment divides each group into a training set and a test set and examine if the algorithm can successfully add the test portion back to the original group. A much larger test suite is used in this experiment.

4.2 Independent Test Suite

In this experiment, the test suite consists of 173 photos downloaded from Flickr by tag based queries. Tags are chosen according to the main themes of our data set and each tag is used as a separate query. The top relevant photos of these queries that focus on one and only one theme form the test suite. For example, if a photo belongs to theme *MacBook*, it will not be a photo of *MacBook Pro*. Photos happen to appear in our test data are excluded to ensure the independence of our test suite.

Three measures are used to evaluate the quality of recommendation: precision, R-precision [5] and top-1-precision scores. Precision (P) is defined as the fraction of retrieved groups that are relevant. Our data set contains relatively small number of groups, as a result the relevant groups for each query photo is quite limited. This may lead to a relatively low precision for both CSM and FPM. R-precision (R-P) is a more appropriate measure. It is defined as the fraction of relevant items that are retrieved among the top r retrieved results. In this case r is set to the number of relevant groups agreed by the evaluators which makes R-precision very similar to *recall*. However, considering the limited number of relevant groups for each query photo among which most of them are likely to be retrieved by either algorithm, we believe that R-precision is better than recall in terms of depicting the capability that an algorithm suggests relevant groups as its top recommendations. While precision and R-precision tend to overlook ranking of the recommended groups, top-1-precision (T1-P) can better reflect an algorithm's discriminative power. It is defined as the precision that an algorithm recommends the top 1 evaluator-suggested group as its own top 1. As each test photo contains only one theme, and each of these themes has only one group that fit it the best, T1-P is the best approach to examine the accuracy of algorithms differentiating between themes.

The user evaluation is run in two sessions. Session I focuses on photo content. Information provided for user evaluation includes the title, picture and tags of the test photos. A list of groups is also given to the evaluators to identify related ones for each photo. Group title, topic photos (if available) and top 5 tags of that group are provided to assist the judgment. All the camera groups are not included in the list given. As most of the groups have an unbiased, content based central concept, it is relatively easy for an evaluator to associate those groups with a photo by the information we provide. In Session II, evaluators are asked to add possible relevant camera groups based on tag information. As a result, these camera groups will always rank lower than the content-based groups in user evaluation.

We precompute patterns, group vector length and inverted index for the entire test data. FPM has around 111K index compared with 1.6M index for CSM. The average query response time for FPM is 0.5 seconds in contrast to 10s in CSM. It is clear that in terms of space cost and query response time, FPM outperforms CSM with large margin, as it requires substantially less storage for precomputed patterns and the response time is much faster.

The relevant groups identified by the evaluators are used to compute the three types of precision scores. Figure 5 shows the results. In general, CSM has much lower *P* and *T1-P* while the *R-P* is comparable with FPM. We further divide the test suite into focused and less focused sets according to their themes. There are 137 photos with content clearly associated with a focused theme, such as *MacBook*. The remaining 36 photos correspond to themes like *nature* and *portrait* and are put in subjective-themed groups by the evaluators. We notice that the *precision* of CSM is constantly much lower than FPM, as it tends to recommend much more groups than FPM. It is also interesting to observe that FPM has higher *R-P* and *T1-P* in recommending groups with focused content; while CSM has better *R-P* and *T1-P* in recommending subjective-themed groups. Such observation is consistent with the algorithms' underlying feature. Groups with focused content often have many long and expressive patterns with high support percentage, while groups with less focused content often have much less number of patterns and the patterns are much shorter with a quite low support percentage. This leads to FPM ranking some less focused groups that are loosely related to the photos lower than those focused groups or even failing to recommend them. We also compare the discriminative power of CSM and FPM by selecting several themes that share similar tag distribution or top tags, such as [*cat, Big Cat*] and [*MacBook, MacBook Pro*]. We can see from Figure 6 that FPM outperforms CSM in terms of *T1-P*. It illustrates the pitfall of considering tag's single occurrence rather than their co-occurrence. Large groups are likely to be ranked as the top few ones in the CSM recommendation list. For example, *nature*, *city* and *landscape* are top tags in some extremely large camera groups. Although these tags also appear within other groups, when they present in a photo, CSM always highly recommend the camera groups regardless of omission of camera information in the photo's tags. Therefore, we come to a conclusion that CSM is prone to larger groups with wider topics and it has a poor accuracy when differentiating two or more themes that share some common features. In contrast, FPM tends to put content-focused groups (with longer patterns) in front of those less-focused large group.

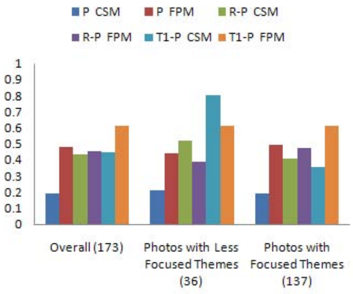


Fig. 5. General comparison between CSM and FPM

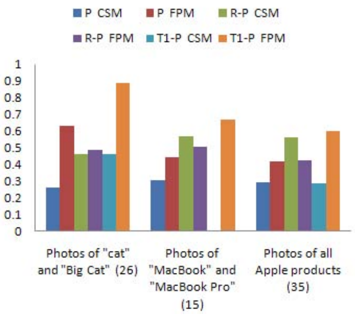


Fig. 6. Discriminative power comparison between CSM and FPM

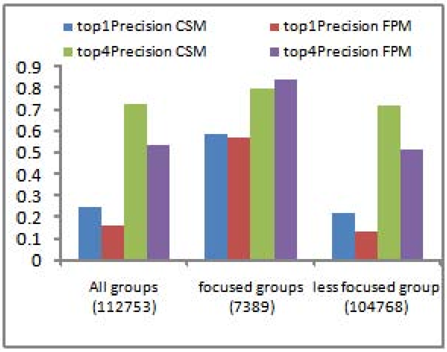
In summary, CSM is in favour of large and less focused groups. It tends to recommend those groups when there is certain tag overlapping. Such bias sometimes increases the recommendation noise for photos that are only remotely related with those groups. On the other hand, FPM performs better in focused groups especially in separating different concepts sharing similar textual content. However, it tends to misrepresent those large and less focused groups.

4.3 Build-in Test Suite

We are able to get the date that a photo is added to a particular group from Flickr API. Based on the date information, we extract photos added within the last two months dating back from the collection time as the test suite. The remaining photos in each group form the new test data collection. Over 108K photos are included in the test suite. Majority of the individual test photo belongs to only one group. Around 4.5% of the photos belong to 2 or more group. The maximum number of groups shared by an individual photo is 4. We compute *top1precision* and *top4precision* as measure. These precisions are defined on group rather than on individual query. *top1precision* is simply the percentage of photos in a group that has its original group as top 1 in the recommendation list. *top4precision* is the percentage of photos that has its original group included in top 4 of the recommendation list.

Table 3. Storage, Execution Cost and Precision Comparison

	FPM	CSM
Index number	129K	1.46M
index size	9.6M	37.5M
Execution Time (s)	1.2K	214.7K
Per Query response time (s)	0.01	1.98



For each group, there is a certain percentage of photos fail to appear in the top list of neither algorithm. Three types of photos fall in this set: (1) A photo with only one tag and the tag is the frequent tag of many groups. For instance, photos with only a tag *apple* always have the same recommendations regardless of their original groups; (2) A photo with a short list of unusual tags. for instance, a photo tagged with [*adayatthezoo*], a tag appear only twice in one group and once in another group does not get any recommendation from FPM and a relative random recommendation by CSM; (3) A photo with tag list closer to other groups than its original one. For instance, a photo of iPod tagged with [*colour, art, cards, crafts*] is put into groups with *portrait* theme by both algorithms. All such cases are examples of lack of expressive tags and cannot be processed accurately by text-only algorithms.

Photos that are missed by FPM but appearing on CSM recommendation list have relatively low occurring tags. Photos missed by CSM but appearing on FPM top recommendation list often have camera and subjective theme groups presented as noise. Others have the typical symptom as described in Figure 4. For instance, a photo with tags like [*streetphotography, color, apple, street, newyork, women, people, city*] from a group about *New York City* has many groups related with *apple* recommended by CSM.

4.4 Parameter Analysis

Support threshold in FPM algorithm is the only parameter involved. Typically, support threshold is set as a percentage of total transaction number. Considering the large variation of group sizes, it is difficult to define a one-size-fit-all percentage value. A threshold value at 2.5% would require around 100 photos to have the same tag(s) in a small group. It would require around 20K photos to have the same tag(s) in a large group. To make the patterns generated comparable among groups of different sizes, we set the threshold value so that top 100 tags of each group form the large 1-item set. We generate two sets of frequent patterns based on 2.5% threshold and on top 100 tag threshold. We run FPM on both FP sets using the build-in test suite. Figure 7 highlights the difference. The chart on left panel shows the overall *top4precision* obtained using two thresholds. Support based on top 100 tags obtains higher precision, but the improvement is not very big. We also show a few individual groups' performance using the two thresholds. We select one group with specific concept (sc), one group with general concept (gc), one group with subjective theme (st) and three camera groups (c1-c3). The improvement on precision is more prominent in less focused groups. The chart on the right panel shows the number of patterns and distinct tags involved for those groups. We can see that the top 100 tag based support significantly increased the number of patterns in all groups. This is partly due to the number of tags involved in the frequent pattern computation. For many groups, especially less focused groups, using 2.5% threshold results in a very limited set of tags.

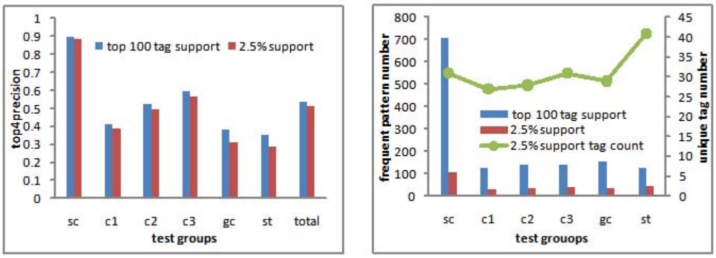


Fig. 7. Comparision between different support values

5 Conclusion

In this paper we propose and implement two algorithms for recommending categories to a given resources based on the textual information. The baseline algorithm (CSM) takes traditional IR approach while another algorithm (FPM) takes data mining approach. Overall, the CSM approach favours large and the less focused group while the FPM algorithm performs better in separating various concepts with similar textual content. We run two experiments on Flickr group data and give detailed analyse of the algorithms performance with respect to group features. The proposed algorithm can be used in group and other user-specified structure.

References

1. Chen, H., Chang, M., Chang, P., Tien, M., Hsu, W.H., Wu, J.: SheepDog - Group and Tag Recommendation for Flickr Photos by Automatic Search-based Learning. In: Proceeding of the 16th ACM international conference on Multimedia, Vancouver, British Columbia, Canada, pp. 737–740 (2008)
2. Flickr, `http://www.flickr.com`
3. Golder, S., Huberman, B.A.: The structure of collaborative tagging systems, `http://www.hpl.hp.com/research/idl/papers/tags/tags.pdf`
4. Liu, D., Hua, X., Yang, L., Wang, M., Zhang, H.: Tag Ranking. In: Proceedings of the 18th International Conference on World Wide Web, Madrid, Spain, pp. 351–360 (2009)
5. Manning, C.D., Raghavan, P., Schtze, H.: An Introduction to Information Retrieval. Cambridge University Press, Cambridge (2009)
6. Mathes, A.: Folksonomies - Cooperative Classification and Communication through Shared Metadata, `http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html`
7. Negoescu, R., Gatica-Perez, D.: Analyzing Flickr Groups. In: Proceeding of the 2008 International Conference on Content-Based Image and Video Retrieval, Niagara Falls, Ontario, Canada, pp. 417–426 (2008)
8. Zobel, J., Moffat, A.: Exploring the similarity space. ACM SIGIR Forum 32(1), 18–34 (Spring 1998)

Semantic Tag Cloud Generation via DBpedia

Roberto Mirizzi[1], Azzurra Ragone[1,2],
Tommaso Di Noia[1], and Eugenio Di Sciascio[1]

[1] Politecnico di Bari – Via Orabona, 4, 70125 Bari, Italy
mirizzi@deemail.poliba.it, {ragone,dinoia,disciacio}@poliba.it
[2] University of Trento – Via Sommarive, 14, 38100 Povo (Trento), Italy
ragone@disi.unitn.it

Abstract. Many current recommender systems exploit textual annotations (tags) provided by users to retrieve and suggest online contents. The text-based recommendation provided by these systems could be enhanced (i) using unambiguous identifiers representative of tags and (ii) exploiting semantic relations among tags which are impossible to be discovered by traditional textual analysis. In this paper we concentrate on annotation and retrieval of web content, exploiting semantic tagging with DBpedia. We use semantic information stored in the DBpedia dataset and propose a new hybrid ranking system to rank keywords and to expand queries formulated by the user. Inputs of our ranking system are (i) the DBpedia dataset; (ii) external information sources such as classical search engine results and social tagging systems. We compare our approach with other RDF similarity measures, proving the validity of our algorithm with an extensive evaluation involving real users.

Keywords: content-based recommendation, RDF ranking, DBpedia.

1 Introduction

Many content-based recommender systems exploit textual annotation (e.g., tags) to recommend content to web users. Moreover, in the current Web 2.0 we also see e-commerce sites giving the possibility to users to tag content/products they want to sell/buy, in order to make them easily retrievable by other users willing to buy/sell that content or products[1]. However, the limits of pure textual-based recommender systems are well know: as semantic relations among keywords are not taken into account, they cannot recognize different keywords with the same meaning (synonymy), as well as the fact that a single word can have different meanings (polysemy). This is the main reason why proposed content is sometimes not in topic with what the user is looking for. Pure textual approaches do not allow to face problems such as synonymy, polysemy, homonymy, context analysis, nor to discover particular relations as hyponymy and hyperonymy[2]. Several semantic-based systems exploits ontological information [8,5,4]

[1] Just to cite an example, Amazon allows users to tag products to improve the search and recommendation process (http://www.amazon.com/gp/tagging/cloud/).

[2] www.wikipedia.org/wiki/{Synonym|Polysemy|Homonym|Hyponymy}

F. Buccafurri and G. Semeraro (Eds.): EC-Web 2010, LNBIP 61, pp. 36–48, 2010.

to overcome the above mentioned issues. Unfortuantely, the main problem of such approaches is that it is very laborious to maintain an ontology regularly updated. Projects like DBpedia[3] may solve the issue of having a semantic source of information regularly updated and which covers a wide range of fields, being based on Wikipedia. Indeed, DBpedia is a community effort to extract structured information from Wikipedia and to make this information available on the Web as a RDF dataset, allowing to pose sophisticated SPARQL queries to Wikipedia. Terms from DBpedia can be used to annotate and represents web contents. Compared to other subject hierarchies and taxonomies, DBpedia has the advantage that each term/resource is endowed with a rich description including abstracts in more than 90 languages. Another advantage, compared to static hierarchies, is that DBpedia evolves as Wikipedia changes. Moreover, each concept in DBpedia is referred by its own URI. This allows to precisely get a resource with no ambiguity. For example, the American corporation *Google Inc.* headquartered in California is referred to as the resource identified by the URI http://dbpedia.org/resource/Google, whereas the American comic strip *Barney Google* created in 1919 by Billy DeBeck is referred to as the URI http://dbpedia.org/resource/Barney_Google_and_Snuffy_Smith.

The main idea behind our approach is the following: keywords can be mapped to corresponding DBpedia resources. After this mapping, we are able to associate a well defined semantics to keywords and we can enrich the "meaning" of the keywords by exploiting the ontological nature of DBpedia. Main contributions of this work are:

- A tool for the semantic annotation of web resources, useful in both the tagging phase and in the retrieval one (see Section 2).
- A novel *hybrid* approach to rank resources on DBpedia w.r.t. a given keyword. Our system combines the advantages of a *semantic-based* approach (relying on a RDF graph) with the advantages of *text-based* IR approaches as it also exploits the results coming from the most popular search engines (Google, Yahoo!, Bing) and from a popular social bookmarking system (Delicious). Moreover, our ranking algorithm is enhanced by textual and link analysis (abstracts and wikilinks in DBpedia coming from Wikipedia).
- A *relative* ranking system: differently from PageRank-style algorithms, each node in the graph has not an importance value per se, but it is ranked w.r.t. its neighborhood nodes. That is, each node has a different importance value depending on the performed query. In our system we want to rank resources w.r.t. a given query by retrieving a ranking list of resources. For this reason we compute a weight representing a similarity relation between resources, instead of a weight for the single resource, as in PageRank-style algorithms.
- An extensive evaluation of our algorithm with real users and comparison w.r.t. other four different ranking algorithms, which provides evidence of the quality of our approach.

The remainder of the paper is structured as follows: in Section 2 we introduce our motivating scenario and present a first implementation of the whole system,

[3] http://dbpedia.org/

which is detailed in Section 3. Then, in Section 4 we show and discuss the results of experimental evaluation. In Section 5 we discuss related work with respect to our approach. Conclusion and future work close the paper.

2 Not Only Tag: A Tool for Tag Cloud Generation

In this section we describe a semantic social tagging system *Not Only Tag* – *NOT* (available at `http://sisinflab.poliba.it/not-only-tag`, see Figure 1)) that can be used to recommend similar tags to users in the annotation and retrieval process of web resources.

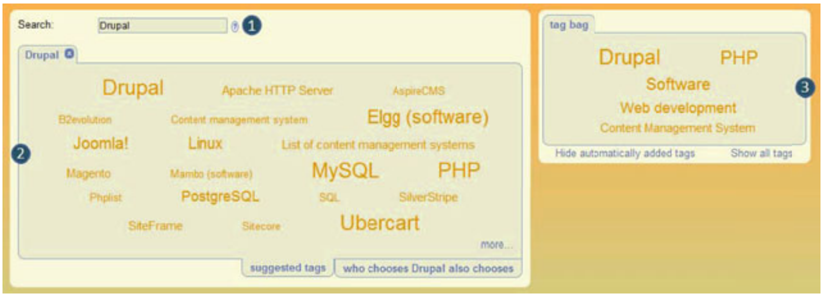


Fig. 1. Screenshot of *Not Only Tag* system

The interaction with the system is very simple and intuitive. Let us suppose the user wants to annotate a software component. The user starts typing some characters (let us say "*Drup*") in the text input area (marked as *(1)* in Figure 1) and the system suggests a list of `DBpedia` URIs whose labels or abstracts contain the typed string. Then the user may select one of the suggested items. We stress here that the user does not suggest just a keyword but a `DBpedia` resource identified by a unique URI. Let us suppose that the choice is the tag *Drupal*, which corresponds to the URI `dbpres:Drupal`.

The system populates a tag cloud (as shown in Figure 1 *(2)*), where the size of the tags reflects their relative **relevance** with respect to *Drupal* in this case (how the relevance is determined is explained in Section 3). We may see that the biggest tags are *Ubercart*, *PHP*, *MySQL*, *Elgg* and *Joomla!*. If the user goes with the mouse pointer over a tag, the abstract of the corresponding `DBpedia` resource appears in a tooltip. This is useful because it allows for a better understanding of the meaning of that tag. When the user clicks on a tag, the corresponding cloud is created in a new tab. Thanks to this feature the user can also navigate the `DBpedia` subgraph in an intuitive way.

The user can collects suggested tags she consider relevant for her campaign by a drag and drop operation of the tag in her tag-bag area (indicated by *(3)* in Figure 1). Once the user selects a tag, the system automatically enriches this area with concepts related to the dropped tag. For example, in the case of *Drupal*, its most related concepts are *PHP*, *Software*, *Web Development*, *Content*

Management System and so on. These new keywords represent the corresponding Wikipedia Categories showed in the Wikipedia page of *Drupal*. Also the tags appearing in the personal tag bag area are sized according to their relevance. Thanks to the `RDF` nature of `DBpedia`, they can be easily computed via nested `SPARQL` queries. In `DBpedia`, for each URI representing Wikipedia category there is a `RDF` triple having the URI as subject, `rdf:type` as property and `skos:Concept` as object. For a further deeper expansion of (semantic) keywords in the tag bag, we also exploit the `skos:broader` and `skos:subject` properties within `DBpedia`. These two properties are used to represent an ontological taxonomy among Wikipedia categories. In particular, `skos:broader` links a category (subject) to its super-category while `skos:subject` relates a resource to its corresponding Wikipedia category. Finally, the `SPARQL` query used to compute the expanded cloud related to a given resource is recursively repeated for all the related categories.

3 An Hybrid Algorithm to Rank DBpedia Resources

In this section we describe our hybrid ranking algorithm *DBpediaRanker*[4], used to rank resources (tags) in `DBpedia` w.r.t. a given keyword. This algorithm computes the *relevance* of `DBpedia` resources (tags) w.r.t. a given keyword and so it allows to determine the *size* of the words in the tag cloud of *NOT* (see Section 2). In a nutshell, *DBpediaRanker* explores the `DBpedia` graph and queries external information sources in order to compute a *similarity value* for each pair of resources reached during the exploration.

The graph browsing, and the consequent ranking of resources, is performed *offline* and, at the end, the result is a weighted graph where nodes are `DBpedia` resources and weights represent the similarity value between any pair of nodes. The graph so obtained will then be used at *runtime*:

- in the *annotation phase*, to suggest *similar* tags to users annotating e.g. their software components;
- in the *retrieval phase*, to display components annotated with tags semantically related to the ones used in the query.

The similarity value between any pair of resources in the `DBpedia` graph is computed querying *external information sources* (search engines and social bookmarking systems) and exploits *textual* and *link analysis* in `DBpedia`. For each pair of resource nodes in the explored graph, we perform a query to each external information source: we search for the number of returned web pages containing the labels of each nodes individually and then for the two labels together (as explained in Section 3.2). Moreover, we look at, respectively, **abstracts** in Wikipedia and **wikilinks**, i.e., links between Wikipedia pages. Specifically, given two resource nodes a and b, we check if the label of node a is contained in the abstract of node b, and vice versa. The main assumption behind this check is that if a `DBpedia` resource name appears in the abstract of another `DBpedia` resource

[4] For a more detailed description of the system the interested reader can refer to `http://sisinflab.poliba.it/publications/2010/MRDD10a/`

it is reasonable to think that the two resources are related with each other. For the same reason, we also check if the Wikipedia page of resource a has a link to the Wikipedia page of resource b, and vice versa. In the following we will present in details all the components of our system, whose architecture is sketched in Figure 2.

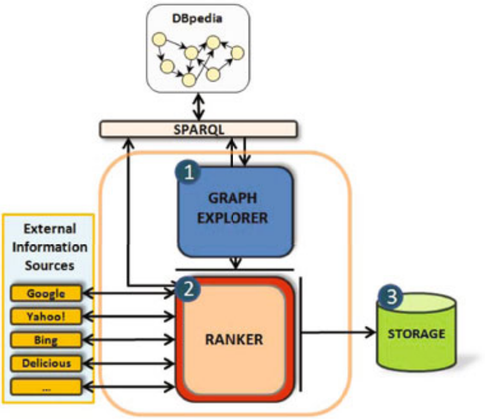


Fig. 2. The ranking system *DBpediaRanker*

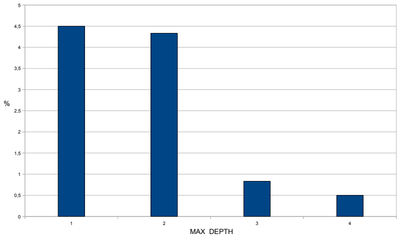


Fig. 3. Evaluation for MAX_DEPTH. It represents the average percentage (y axis) of the top-10 resources related to 100 seeds within a distance of 1 to 4 hops (x axis).

3.1 Graph Explorer

This module queries `DBpedia` via its `SPARQL` endpoint. Given a `DBpedia` URI [5], the explorer looks for other URIs connected to it via a set of predefined properties. The properties of `DBpedia` to be explored can be set in the system before the exploration starts. In our initial setting, we decided to select only the `SKOS` properties `skos:subject` and `skos:broader` Indeed, these two properties are not specific of a particular context and are very popular in the `DBpedia` dataset. Hence, they can be used as a good starting point. Moreover, we observed that the majority of nodes reached by other properties were also reached by the selected properties, meaning that our choice of `skos:subject` and `skos:broader` properties does not disregard the effects of potentially domain-specific properties.

Given a root URI, this is explored up to a predefined distance, that can be configured in the initial settings. We found through a series of experiments that setting this distance, that we call MAX_DEPTH, equal to 2 is a good choice. Indeed, resources within two hops are still highly correlated to the root URI, while going to the third hop this correlation quickly decreases. Indeed, we noticed that if we set $MAX\_DEPTH = 1$ (this means considering just nodes directly linked) we loose many relevant relation between pairs of resources. On the other hand, if we set $MAX\_DEPTH > 2$ we have too many non relevant resources.

[5] From now on we use the words *URI* and *resource* indistinctly.

In order to find the optimal value for MAX_DEPTH, we initially explored 100 seed nodes up to a $MAX\_DEPTH = 4$. After this exploration was completed, we retrieved the top-10 (most similar) related resources for each node (see Section 3.2). The results showed that on the average the 85% of the top-10 related resources where within a distance of one or two hops. The resources two hops far from the seeds where considered as the most relevant the 43% of times ($\sigma = 0.52$). On the contrary the resources above two hops were rarely present among the first results (less than 15% of times). In figure 3 the average percentage of top-10 related resources w.r.t. to the distance from a seed (MAX_DEPTH) is shown.

3.2 Ranker

Here we describe the ranker, the core component of the whole system. Given any pair of resources in the DBpedia graph it determines a similarity value between them; this similarity value is the weight associated to the edge between the two resources.

Given two URIs uri_1 and uri_2 in the same graph-path, it compares how much they relate with each other exploiting information sources external to DBpedia such as search engines and social tagging systems.

The aim of this module is to evaluate how strong is a semantic connection between two DBpedia resources using information taken from external sources. In our current implementation we consider as external sources both (i) web search engines (Google, Yahoo! and Bing) and (ii) social tagging systems (Delicious), plus (iii) Wikipedia-related information contained in DBpedia. Given two DBpedia resources uri_1 and uri_2, we verify how many web pages contain (or have been tagged by) the value of the `rdfs:label` associated to uri_1 and uri_2. Then we compare these values with the number of pages containing (or tagged by) both labels. We select more than one search engine because we do not want to bind the result to a specific algorithm of a single search engine. Moreover, we want to rank a resource not only with respect to the popularity of related web pages on the web, but also considering the popularity of such resources among users (e.g., in Delicious). In this way we are able to combine two different perspectives on the popularity of a resource: the one related to the words occurring within web documents, the other one exploiting the social nature of the current web. Through formula (1) we evaluate the related similarity of two URIs uri_1 and uri_2 with respect to a given external information source $info\_source$.

$$sim(uri_1, uri_2, info\_source) = \frac{p_{uri_1,uri_2}}{p_{uri_1}} + \frac{p_{uri_1,uri_2}}{p_{uri_2}} \qquad (1)$$

Given the information source $info\_source$, p_{uri_1} and p_{uri_2} represent the number of documents containing (or tagged by) the `rdfs:label` associated to uri_1 and uri_2 respectively, while p_{uri_1,uri_2} represents how many documents contain (or have been tagged by) both the label of uri_1 and uri_2. It is easy to see that the formula is symmetric and the returned value is in $[0, 2]$. The number of documents containing both labels has to be always lower or equal to those containing only one of the two labels. Otherwise we set the value $\frac{p_{uri_1,uri_2}}{p_{uri_1}} = 0$.

Ranker does not use only external information sources but exploits also further information from `DBpedia`. In fact, we also consider Wikipedia hypertextual links mapped in `DBpedia` by the property `dbpprop:wikilink`. Whenever in a Wikipedia document w_1 there is a hypertextual link to another Wikipedia document w_2, in `DBpedia` there is a `dbpprop:wikilink` from the corresponding resource URIs uri_1 and uri_2. Hence, if there is a `dbpprop:wikilink` from uri_1 to uri_2 and/or vice versa, we assume a stronger relation between the two resources. We evaluate the strength of the connection as follow:

$$wikiS(uri_1, uri_2) = \begin{cases} 0, \text{ no } \texttt{wikilink} \text{ between } uri_1 \text{ and } uri_2; \\ 1, \texttt{wikilink} \text{ only from } uri_1 \text{ to } uri_2; \\ 1, \texttt{wikilink} \text{ only from } uri_2 \text{ to } uri_1; \\ 2, \texttt{wikilink} \text{ both from } uri_1 \text{ to } uri_2 \text{ and} \\ \quad \text{vice versa;} \end{cases}$$

Furthermore, given two resources uri_1 and uri_2, we check if the `rdfs:label` of uri_1 is contained in the `dbpprop:abstract` of uri_2 (and vice versa). Let n be the number of words composing the label of a resource and m the number of words composing the label which are also in the abstract, $abstractS(uri_1, uri_2) = \frac{m}{n}$, with $\frac{m}{n}$ in [0,1] as $m \leq n$. At the end, the similarity value between uri_1 and uri_2 is computed as the sum of the functions:

$$\begin{aligned} &sim(uri_1, uri_2, google) + sim(uri_1, uri_2, yahoo) + sim(uri_1, uri_2, bing) + \\ &+ sim(uri_1, uri_2, delicious) + wikiS(uri_1, uri_2) + abstractS(uri_1, uri_2) \end{aligned} \tag{2}$$

4 Evaluation

In the experimental evaluation we compared our *DBpediaRanker* algorithm with other four different algorithms; some of them are just a variation of our algorithm but lack of some key features.

Algo2 is equivalent to our algorithm, but it does not take into account textual and link analysis in `DBpedia`.

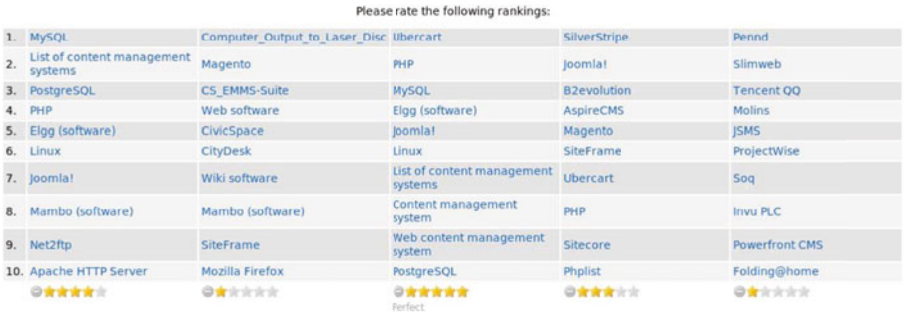
Algo3 is equivalent to our algorithm, but it does not take into account external information sources, i.e., information coming from search engines and social bookmarking systems.

Algo4, differently from our algorithm, does not exploit textual and link analysis. Moreover, when it queries external information sources, instead of the formula (1), it uses the *co-occurrence* formula: $\frac{p_{uri_1,uri_2}}{p_{uri_1}+p_{uri_2}-p_{uri_1,uri_2}}$

Algo5 is equivalent to *Algo4*, but it uses *similarity distance* [1] instead of co-occurrence.

We did not choose to use either co-occurrence or similarity distance in *DBpediaRanker* since they do not work well when one of the two resources is extremely more popular than the other, while formula (1) allows to catch this situation.

In order to assess the quality of our algorithm we conducted a study where we asked to participants to rate the results returned by each algorithm. For each query, we presented five different rankings, each one corresponding to one of the



Please rate the following rankings:

1.	MySQL	Computer_Output_to_Laser_Disc	Ubercart	SilverStripe	Pennd
2.	List of content management systems	Magento	PHP	Joomla!	Slimweb
3.	PostgreSQL	CS_EMMS-Suite	MySQL	B2evolution	Tencent QQ
4.	PHP	Web software	Elgg (software)	AspireCMS	Molins
5.	Elgg (software)	CivicSpace	Joomla!	Magento	JSMS
6.	Linux	CityDesk	Linux	SiteFrame	ProjectWise
7.	Joomla!	Wiki software	List of content management systems	Ubercart	Soq
8.	Mambo (software)	Mambo (software)	Content management system	PHP	Invu PLC
9.	Net2ftp	SiteFrame	Web content management system	Sitecore	Powerfront CMS
10.	Apache HTTP Server	Mozilla Firefox	PostgreSQL	Phplist	Folding@home
	★★★★☆	★☆☆☆☆	★★★★★ Perfect	★★★☆☆	★☆☆☆☆

Fig. 4. Screenshot of the evaluation system. The five columns show the results for, respectively, *Algo3*, *Algo4*, *DBpediaRanker*, *Algo2* and *Algo5*.

ranking methods. The result lists consisted of the top ten results returned by the respective method. In Figure 4, results for the query *Drupal* are depicted. Looking at all the results obtained with our approach (column 3), we notice that they really are tightly in topic with *Drupal*. For example, if we focus on the first three results, we have *Ubercart*, that is the popular e-commerce module for *Drupal*, *PHP*, which is the programming language used in *Drupal*, and *MySQL*, the most used DBMS in combinance with *Drupal*. The other results are still very relevant, we have for example *Elgg* and *Joomla!*, that are the major concurrents of *Drupal*, and *Linux*, which is the common platform used when developing with *Drupal*. It is very likely that a user who knows *Drupal*, also knows the languages and technologies our measure returned.

We point out that even if we use external information sources to perform substantially a textual search (for example checking that the word *Drupal* and the word *Ubercart* appear more often in the same Web pages with respect to the pair *Drupal* and *PHP*), this does not mean that we are discarding semantics in our search and that we are performing just a string comparison. Indeed, we are *not* performing just a keyword-based search: this is still more evident if we consider the best results our system returns if the query is *PHP*. In fact, in this case no node having the word *PHP* in the label appears in the first results. On the contrary the first results are *Zend Framework* and *Zend Engine*, that are respectively the most used web application framework when coding in PHP and the heart of PHP core. *PHP-GTK* is one of the first resources that contains the word *PHP* in its label and is ranked only after the previous ones.

During the evaluation phase, the volunteers were asked to rate the different ranking algorithms from 1 to 5 (as shown in Figure 4), according to which list they deemed represent the best results for each query. The order in which the different algorithms were presented varied for each query: e.g., in Figure 4 the results for *DBpediaRanker* algorithm appear in the third column, a new query would show the results for the same algorithm in a whatever column between the first and the last. This has been decided in order to prevent users to being influenced by previous results. For the same reason the columns do not have the name of the ranking measure.

The area covered by this test was the *ICT* one and in particular *programming languages* and *databases*.

The test was performed by 50 volunteers during a period of two weeks. The users were Computer Science Engineering master students (last year), Ph.D. students and researchers belonging to the ICT scientific community. For this reason, the testers can be considered IT domain experts. During the testing period we collected 244 votes. It means that each user voted on average about 5 times. The system is still available at `http://sisinflab.poliba.it/evaluation`. The user can search for a keyword in the ICT domain by typing it in the text field, or she may directly select a keyword from a list below the text field that changes each time the page is refreshed. While typing the resource to be searched for, the system suggests a list of concepts obtained from `DBpedia`.

If the service does not return any result, it means that the typed characters do not have any corresponding resource in `DBpedia`, so the user can not vote on something that is not in the `DBpedia` graph. It may happen that after having chosen a valid keyword (i.e., an existing resource in `DBpedia`) from the suggestion list, the system says that there are no results for the selected keyword. It happens because we limited the exploration of the `RDF` graph to nodes belonging to *programming languages* and *databases* domain, while the URI lookup web service queries the whole `DBpedia`. For the sake of simplicity, in this first experiment we decided not to filter results from the URI lookup web service with the nodes in our context. Occasionally it may happen that a keyword belonging to IT domain gives no results, this could happen because the selected resource has not yet been analyzed by *DBpediaRanker*.

In all other cases the user will see a screenshot similar to the one depicted in Figure 4. Moving the mouse pointer on a cell of a column, the cells in other columns having the same label will be highlighted. This allows the user to better understand how differently algorithms rank the same resource and in which positions the same labels are in the five columns. Clicking on a concept, the corresponding Wikipedia page will open in an iframe. This facilitates the user to obtain more information about the clicked concept.

Finally, the user can start to rate the results of the five algorithms, according to the following scale: (i) one star: *very poor*; (ii) two stars: *not that bad*; (iii) three stars: *average*; (iv) four stars: *good*; (v) five stars: *perfect*. The user has to rate each algorithm before sending her vote to the server. Once rated the current resource, the user may vote for a new resource if she wants. For each voting we collected the time elapsed to rate the five algorithms: on the average it took about 1 minute and 40 seconds ($\sigma = 96.03$ s). The most voted resources were `C++`, `MySQL` and `Javascript` with 10 votings.

In Figure 5 we plotted the mean of the votes assigned to each method. Error bars represent standard deviation. *DBpediaRanker* has a mean value of 3.91 ($\sigma = 1.0$). It means that, on the average, users rated it as *Good*. Examining its standard deviation, we see that the values are within the range of $\sim 3 \div 5$, i.e., the ranks are comprised between *Average* and *Perfect*. In order to determine if the differences between our method and the others are statistically significant we used the Wilcoxon test [13]. From the Wilcoxon test we can conclude that not only our algorithm performed always better than the others, but also that

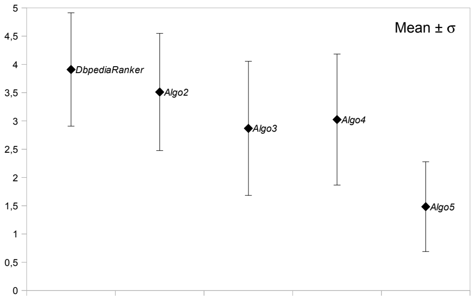


Fig. 5. Average ranks

the (positive) differences between our ranking and the others are statistically significant. Indeed, the *z-ratio* obtained by comparing *DBpediaRanker* algorithm with *Algo2*, *Algo3*, *Algo4* and *Algo5* is respectively 4.93, 8.71, 7.66, 12.89, (with $p < 0.0001$). By comparing these values with the critical value of z[6], we can reject the null hypothesis (correlated rankings), and say that the differences between our algorithm and the others are statistically significant.

5 Related Work

Nowadays, a lot of websites expose their data as RDF documents; just to cite a few: the *DBPL* database, *RDF book mashup*, *DBtune*, *MusicBrainz*[7]. Therefore, it would be very useful to have some metrics able to define the relevance of nodes in the RDF graph, in order to give back to the user a *ranked* list of results, ranked w.r.t. the user's query. In order to overcome this limit several PageRank-like [11] ranking algorithms have been proposed [2,7,9,6]. They seem, in principle, to be good candidates to rank resources in an RDF knowledge base. Yet, there are some considerable differences, that cannot be disregard, between ranking web documents and ranking resources to which some semantics is attached. Indeed, the only thing considered by the PageRank algorithm is the origin of the links, as all links between documents have the same relevance, they are just hyperlinks. For RDF resources this assumption is no more true: in an RDF graph there are several types of links, each one with different relevance and different semantics, therefore, differently from the previous case, an RDF graph is not just a graph, but a directed graph with labels on each edge. Moreover an RDF resource can have different origins and can be part of several different contexts and this information cannot be disregarded, instead it should be exploited in some way in the ranking process. *Swoogle* [2] is a semantic web search engine and a metadata search provider, which uses the *OntologyRank* algorithm, inspired by the PageRank algorithm. Differently from *Swoogle*, that ranks RDF documents which

[6] http://faculty.vassar.edu/lowry/ch12a.html

[7] http://www.informatik.uni-trier.de/~ley/db/, http://www4.wiwiss.fu-berlin.de/bizer/bookmashup/, http://dbtune.org/, http://musicbrainz.org/

refer to the query, our main task is to rank RDF resources *similar* to the query. Nonetheless, we borrowed from Swoogle the idea of browsing only a predefined subset of the semantic links. Similarly to our approach also the *ReConRank* [7] algorithm explores just a specific subgraph: when a user performs a query the result is a topical subgraph, which contains all resources related to keywords specified by the user himself. In the subgraph it is possible to include only the nodes *directly* linked to the particular root node (the query) as well as specify the number n of desired hops, that is how far we want to go from the root node. The *ReConRank* algorithm uses a PageRank-like algorithm to compute the relevance of resources, called *ResourceRank*. However, like our approach, the *ReConRank* algorithm tries to take into account not only the relevance of resources, but also the "context" of a certain resource, applying the *ContextRank* algorithm [7]. Our approach differs from [7] due to the semantic richness of the DBpedia graph (in terms of number of links) the full topical graph for each resource would contain a huge number of resources. This is the reason why we only explore the links skos:subject and skos:broader. Hart et al. [6] exploit the notion of naming authority, introduced by [9], to rank data coming from different sources. In order to achieve this aim they use an algorithm similar to PageRank, adapted to structured information such as the one contained in an RDF graph. However, as for PageRank, their ranking measure is absolute, i.e. it does not depend on the particular query. In our case, we are not interested in an absolute ranking and we do not take into account naming authority because we are referring to DBpedia: the naming authority approach as considered in [6] loses its meaning in the case of a single huge source such as DBpedia. Mukherjea et al. in [10] presented a system to rank RDF resources inspired by [9]. As in the classical PageRank approach the relevance of a resource is decreased when there are a lot of outcoming links from that, nevertheless such an assumption seems not to be right in this case, as if an RDF resource has a lot of outcoming links the relevance of such a resource should be increased not decreased. In our approach, in order to compute if a resource is within or outside the context, we consider as *authority* URIs the most popular DBpedia categories. Based on this observation, URIs within the context can be interpreted as *hub* URIs. *TripleRank* [3], by applying a decomposition of a 3-dimensional tensor that represents an RDF graph, extends the paradigm of two-dimensional graph representation, introduced by HITS, to obtain information on the resources and predicates of the analyzed graph. In the pre-processing phase they prune dominant predicates, such as dbpprop:wikilink, which, instead, have a fundamental role as shown in the experimental evaluation. Moreover in [3] they consider only objects of triples, while we look at both directions of statements. Finally, as for all the HITS-based algorithms, the ranking is just based on the graph structure. On the contrary we also use external information sources. *Sindice* [12], differently from the approaches already presented, does not provide a ranking based on any lexicographic or graph-based information. It ranks resources retrieved by SPARQL queries exploiting external ranking services (as Google popularity) and information related to hostnames, relevant statements, dimension of information sources. Differently from our approach, the main task of Sindice is to return RDF triples (data) related to a given query. Kasneci et al. [8] present a semantic search engine *NAGA*. It extracts information from several sources on the web and, then, finds relationships between the extracted entities. The system answers to queries about relationships already collected in it, which at the moment of the writing are around one hundred. Differently from our system, in order to query NAGA the user has to know all the relations that can possibly link two entities and has to learn a specific query language, other than know the exact name of the label she is looking for; while we do not require any technical knowledge to our users, just the ability to use tags. We do not collect information from the entire Web, but we rely on the `Linked Data` cloud, and in particular on `DBpedia` at the present moment.

6 Conclusion and Future Work

In this paper we presented a novel system for semantic tag generation and retrieval. We motivated our approach in a scenario of annotation of web resources, showing how exploiting semantic information in `DBpedia` it is possible both (i) to help users in the process of tag selection, and (ii) to enhance the retrieval process of previously annotated content, displaying the most relevant resources w.r.t. the keywords (tags) specified by the user. We described the components of our system and showed the validity of our approach through experimental results supported by extensive users evaluation. Currently, we are mainly investigating on how to extract more fine grained contexts and how to enrich the context extracting not only relevant resources but also relevant properties.

Acknowledgment

This research has been supported by Apulia Strategic projects PS_092, PS_121, PS_025.

References

1. Cilibrasi, R., Vitányi, P.: The Google Similarity Distance. IEEE Transactions on Knowledge and Data Engineering 19(3), 370–383 (2007)
2. Ding, L., Finin, T., Joshi, A., Pan, R., Cost, S.R., Peng, Y., Reddivari, P., Doshi, V., Sachs, J.: Swoogle: a search and metadata engine for the semantic web. In: CIKM '04, pp. 652–659 (2004)
3. Franz, T., Schultz, A., Sizov, S., Staab, S.: Triplerank: Ranking semantic web data by tensor decomposition. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 213–228. Springer, Heidelberg (2009)
4. Gabrilovich, E., Markovitch, S.: Harnessing the expertise of 70,000 human editors: Knowledge-based feature generation for text categorization. J. Mach. Learn. Res. 8, 2297–2345 (2007)
5. Gates, S.C., Teiken, W., Cheng, K.-S.F.: Taxonomies by the numbers: building high-performance taxonomies. In: CIKM '05: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, pp. 568–577. ACM, New York (2005)

6. Harth, A., Kinsella, S., Decker, S.: Using naming authority to rank data and ontologies for web search. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 277–292. Springer, Heidelberg (2009)
7. Hogan, A., Harth, A., Decker, S.: ReConRank: A Scalable Ranking Method for Semantic Web Data with Context (2006)
8. Kasneci, G., Suchanek, F.M., Ifrim, G., Ramanath, M., Weikum, G.: Naga: Searching and ranking knowledge. In: ICDE 2008 (2008)
9. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. In: Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 668–677 (1998)
10. Mukherjea, S., Bamba, B., Kankar, P.: Information Retrieval and Knowledge Discovery utilizing a BioMedical Patent Semantic Web. IEEE Trans. Knowl. Data Eng. 17(8), 1099–1110 (2005)
11. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank Citation Ranking: Bringing Order to the Web. Technical report (1998)
12. Tummarello, G., Delbru, R., Oren, E.: Sindice.com: Weaving the Open Linked Data. The Semantic Web, 552–565 (2008)
13. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics Bulletin 1(6), 80–83 (1945)

Social Networks as Data Source for Recommendation Systems

Mathias Bank and Juergen Franke

Daimler AG, Research & Development,
D-89081 Ulm, Germany
mathias.bank@daimler.com

Abstract. Reviews and review based rankings are widely used in recommendation systems to provide potential customers quality information about selected products. During the last years, many researchers have shown that these reviews are neither objective nor do they represent real quality values. Even established ranking methods designed to fix this problem have been shown to be unreliable. In this work, user generated content of fora, weblogs and similar trustworthy social networks is proposed as an alternative data source. It is shown how this data can be used to calculate a satisfaction and relevance measure for different product features to provide potential customers reliable quality information. The method is evaluated in the automotive domain using J.D. Power's established Initial Quality Study to ensure providing meaningful quality-related data.

Keywords: social networks, reviews, recommendation system.

1 Introduction

During the last 10 years, the e-commerce sector has shown that online shopping portals have a great growth potential which has not even been stopped by the economic crisis. Nevertheless, in a highly competitive environment it is simply not enough to offer product catalogs only. Today, potential customers expect more comfort: In addition to high quality product information they want to be guided through the large number of possible products. Recommendation systems have been successfully used by many different e-commerce portals to suggest relevant products by providing product rankings or experience reviews of other customers [1,2].

There are different types of systems to convert browsers into buyers, improve cross-selling or enforce customer loyalty [1]. Reviews and review based rankings are the most widely used recommendation system which is supposed to provide relevant quality information based on the experience of other customers. Recent work has shown that this simple and widely used method does not reflect real quality values [3,4] and that even different rating methods (e.g. amazon's "helpful" ratings) cannot assure quality relevant reviews [5,6]. Nevertheless, 78% of Internet users rely on recommendations from consumers and 61% trust customer opinions posted online [7].

F. Buccafurri and G. Semeraro (Eds.): EC-Web 2010, LNBIP 61, pp. 49–60, 2010.

In this work, a new recommendation system is presented that is designed to provide reliable and quality relevant product information. Instead of relying on a small number of locally available and possibly manipulated data, it utilizes user generated content in Internet fora, web-logs and other social networks. The very large amount of mostly unbiased comments (section 2) is used to provide the customer a satisfaction and relevance measure on different abstraction levels. The proposed system provides these measures for the complete life-cycle of a selected product and is also able to analyze quality changes over time by selecting different time slots. This is a very important information for long-run products like cars or hotels.

2 User Generated Content as Recommendation Source

In most community systems, review based recommendation systems are realized using free-text fields in addition to star-ratings. While shop owners can increase their revenue offering comment functionality [4,8], customers are able to read product experience of other customers [3]. This win-win situation leads to a highly accepted recommendation system that is realized in nearly every online shop system and thus comments are becoming an accepted form of cultural expression [3].

Unfortunately, customer reviews are not objective at all. Recent work has shown that most of the reviews posted in online markets are bimodal [9]: they are either allotted an extremely high rating or an extremely low rating. The additional available average numerical star-rating does not convey a lot of information in this situation so that the users have to read free-text comments. It has been shown that most of these comments are positively biased. In some situations, they are misused for advertisement, promotion and communication issues or they are simply duplicated [3]. A measure is needed to distinguish high quality reviews from low quality reviews. Different shopping systems (e.g. amazon.com) have introduced a peer-reviewed measure, in which each customer can specify if a review was helpful. It has been shown that these assessments are influenced by a number of factors [5,6]. Nevertheless, studies have used these votes to train ranking models [10], which in consequence reflect this bias. Different approaches try to solve this problem by using extractable meta data like objectivity, subjectivity and readability to create an objective quality measure [8,5].

Previously mentioned methods do not take time into account, so that it is not possible to identify quality problems over time: The review based recommendation systems assume that there are no quality changes. This is surely not correct for products which are sold for a long time like cars or hotels. Additionally, these methods require a product to have reviews. New or low-traffic items do not have such reviews so that customers may prefer other products. Shop owners and product manufacturers can counteract this by adding some baits but there are also paid or manipulated comments [3]. This not-recommended method undermines trust and thus harms the long-term reliability.

In the following section, a new recommendation system is proposed that is able to fix these vulnerabilities. Instead of relying only on locally available comments that are possibly manipulated to increase sales, the algorithm takes user generated content of fora, weblogs and other social networks into account. Currently, there are more than 475 million active Internet users in the world. More than 38% want to start their own web-log and 34% of blog authors are writing their opinions on products and brands [11]. It is assumed that this data is not as biased as product reviews because there are no ulterior financial motives. Nevertheless, the data source has to be selected carefully. During the last years, especially weblogs have attracted attention due to paid content which is why in the USA the Federal Trade Commission FTC regulates weblogs in their use of paid content with the beginning of December 1st, 2009[1]. An analysis of 1,196 discussions in Internet fora has shown however, that most of these discussions are focused on topics the customer needs help or advice (fig. 1). These discussions provide a lot of quality information. A carefully selected data source of Internet fora and trustworthy weblogs is the base for the proposed recommendation system. It is not recommended to track "the whole" Internet.

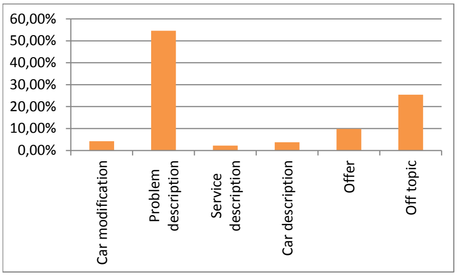


Fig. 1. Abstract topic analysis on 1,196 entries of the automotive forum benzworld.org. Nearly 50% of the posts are dealing with technical or conceptual problems in which the author needs some advice. 25% of the entries are not dealing with any car at all and can be disregarded for quality analysis.

3 Social Network Analysis

In the following, the workflow of the proposed recommendation system is discussed in detail. It is assumed that user generated content of weblogs, fora or other social networks is already downloaded by specialized web crawlers that only store relevant user generated content additional to author names and posting time [12]. Advertisement, navigation and other page structure elements are already filtered out.

3.1 Data Analysis

In contrast to product reviews, user generated content in social networks is not focused on quality or satisfaction reports nor is it supposed to be analyzed

[1] `http://www.ftc.gov/os/2009/10/091005endorsementguidesfnnotice.pdf`

automatically. While there is some structured information in product reviews (e.g. star ratings or pro and contra lists) the discussed user postings are completely unstructured. The text neither contains a special field that defines the discussed product features nor does it provide a defined field for the customer's satisfaction. It is for the algorithms to introduce this structure. This is done by data cleaning, natural language processing and information extraction algorithms. The schematic of the proposed workflow can be seen in figure 2. The recommendation system itself is designed to be independent of used data analysis algorithms. This is why different approaches are discussed and the used algorithms for the realized prototype (section 4) are presented in the following subsections.

Pre-processing. First of all, in a multilingual approach the language of the text has to be identified to load special language dependent resources and algorithms at downstream steps. This is done using a n-gram classification [13]. After tokenization using regular expressions, the content information is enriched with part-of-speech tagging applying the TreeTagger algorithm [14].

Contrary to texts typically used for research issues, user contributions in weblogs, fora and similar systems are of poor quality, containing a lot of abbreviations, misspellings, dialect words, community specific nicknames and syntax (e.g. *@nickname: ...*) so that some cleaning steps are necessary. Tokens which influence analysis negatively — such as stop words, abbreviations, (nick-) names, e-mail addresses and URLs which can be disregarded in downstream analysis steps — are annotated using simple word lists. Misspelled and dialect words are fixed in English comments [15,16]. In other languages this step is not done because tests have shown that a correction will cause more errors[2]. Finally redundant text doublets caused by quotations are annotated comparing word n-grams to detect similar text fragments across postings in a discussion thread.

Topic Detection. In recent years, different researchers have focused aspect-based opinion mining to capture reviewers' opinions toward different product aspects (e.g. [17,18,19,20]). Product feature extraction and categorization is surely the most difficult task in this process. Next to different statistical approaches using co-occurrences, association rules [17], PMI [18] and probability based algorithms [19] have been proposed to handle this difficult task. Synonym treatment is suggested by means of ontologies similar to Wordnet [17,18] or by using fuzzy string matches [5].

Contrary to the discussed approaches, the proposed recommendation system is not designed to focus on product features but on topics generally. The candidate list of potential topics has to be extracted of completely unstructured user comments. The usage of semistructured information [20] is not possible due to missing data structures. Different rule based approaches have shown an average precision between 79% and 89% on usually uniformly formulated phrases [17]. These performances cannot be assumed in unstructured content found in

[2] In German, for example, it is possible to combine different single words to a new one, which to our knowledge cannot be reliably handled by current available algorithms.

weblogs or fora because there are many different comment structures available. Schierle and Trabold proposed to realize a taxonomy based approach [21]: To label each comment with related topics, the terminology of a product domain is organized in a taxonomy providing the possibility to store multi-lingual synonyms for each concept. The special taxonomy structure makes it possible to disambiguate different concepts having the same word associated to them regarding part-of-speech and context. Using simple token matching algorithms, each user comment can be labeled with corresponding terminology terms. This approach ensures high precision which is very important for a recommendation system to gain customers' trust. Thanks to the large amount of available data it is possible to prefer precision while maintaining sufficient data points. The taxonomy structure itself summarizes different concepts to topics. In the following, these topics are called "product features" because it is not important if the analysis focuses on specific features or abstract topics.

To ensure adequate recall values and to cover different terms used by customers, this taxonomy can be extended using semi-supervised terminology extraction algorithms presented in previously mentioned publications. This approach causes of course more human effort. Low precision results, however, would undermine user trust because of wrong topic classification. In addition, off-topic discussions are excluded in the product analysis using this method.

Sentiment Analysis. Next to the topics the users are talking about, it is important to know in which mood they are talking. Recent work dealing with user reviews has used already classified pro and contra information provided by the users themselves [17,20]. Scaffidi et al. suggest to use the additional used star rating assuming that this average rating represents user satisfaction for mentioned product features [19]. This was already disproved by Kano et al. [22] who have recognized that product features satisfy a customer in varying degrees. Lexicon based approaches have been introduced to assign sentiment values to words causing emotions [23,24,25]. This simple technique is not adequate for most real world scenarios because the intended sentiment depends not only on single words but on the context. Different machine learning approaches have been proposed to take this into account (e.g. [26,27]). Dave et al. have shown that these methods perform well on whole reviews but not on sentences [27] so that it is not possible to extract topic related sentiment information.

The proposed analysis algorithm uses an approach similar to the algorithm used for topic detection: A sentiment lexicon is represented as a taxonomy, enriched with context information in which a given sentiment value is valid. Each concept appearance in user generated content is annotated with the corresponding sentiment value $\in [-1; 1]$. To take relational information into account, each sentiment word is assigned to the feature annotation located next to it.

3.2 User Driven SWOT Analysis

The proposed recommendation system is supposed to provide an abstract view to all user comments that are dealing with specific products and product features.

To provide customer related information about product specific strengths and weaknesses, opportunities and threats (SWOT), two different measures are calculated:

1. First a satisfaction index is calculated to estimate the overall quality estimation for each feature y_i and product x_j.
2. Additionally, a relevance index is calculated to provide information on how relevant the product feature is in customers' point of view.

Satisfaction Calculation. Intuitively, one might assume that satisfaction is the ratio of the number of positive to negative comments. But this presupposes to have a balanced lexicon and a balanced language. Both conditions are very unlikely. That is why the system defines *neutral* satisfaction for a feature y_i as ratio of positive $f^+(y_i)$ and negative $f^-(y_i)$ comments for all products. The satisfaction index $s(y_i, x_j)$ for a given product x_j is based on the number of positive posts concerning a product-feature combination $f^+(y_i, x_j)$ and the number of negative posts $f^-(y_i, x_j)$.

$$
\begin{aligned}
f_{\text{positive}} &= \tfrac{f^+(y_i,x_j)}{f^+(y_i)} \\
f_{\text{negative}} &= \tfrac{f^-(y_i,x_j)}{f^-(y_i)}
\end{aligned}
$$

$$
s(y_i, x_j) = \begin{cases} \dfrac{f_{\text{positive}}}{f_{\text{negative}}} - 1 & \text{if } f_{\text{positive}} \leq f_{\text{negative}} \\[2ex] 1 - \dfrac{f_{\text{negative}}}{f_{\text{positive}}} & \text{else} \end{cases} \tag{1}
$$

A potential customer can compare the satisfaction value of different products $x_1, \dots, x_n$ on feature level y_i: A higher satisfaction value $s \in [-1; 1]$ for product x_j and feature y_i implies higher satisfaction compared to other products with feature y_i.

Relevance Calculation. Different product features are discussed with varying frequency. A feature the customer has to deal with every day is more critical in case of misbehavior than a feature used more rarely. A user driven recommendation system has to take this issue into account and thus has to provide a relevance measure. Recent work has used absolute counts [17]. The analysis of user comments has shown however that it is important to be aware of different frequencies of occurrence for different products: On the one side product A is much more mentioned than product B because A is a mainstream product. On the other side a non-mainstream product C can be mentioned even more frequently because the manufacturer was able to motivate users of social networks to talk about its products (e.g. Apple).

The relevance of a given feature y_i for product x_j is the probability $p(y_i|x_j)$ that depends on the frequency f of both features relative to the product itself:

$$
p(y_i|x_j) = \frac{f(y_i, x_j)}{f(x_j)} \tag{2}
$$

The formula measures the probability that other users publish comments about feature y_i while dealing with product x_j. Thus it provides information about how relevant feature y_i is depending on product x_j. It is not necessary to care about different product frequencies any more.

4 Evaluation

4.1 Prototype Implementation

The proposed recommendation system has been realized as a prototype in the automotive domain. It tracks 20 automotive Internet fora and 103 weblogs. In sum, the analysis system has downloaded 13 million German and English user comments. The taxonomy for topic detection contains 2,081 automotive-related multilingual concepts (e.g. components, service terms) with 5,392 synonyms. These terms have been extracted from different automotive lexica in addition to cooccurrences found in real fora data. The sentiment lexicon was created based on [28] in addition to English translations. User comments are preprocessed using the OASIS Unstructured Information Management Architecture (UIMA) [29,30]. All analysis results are stored in a Lucene data index for fast data access so that incremental data updates are possible (fig. 2).

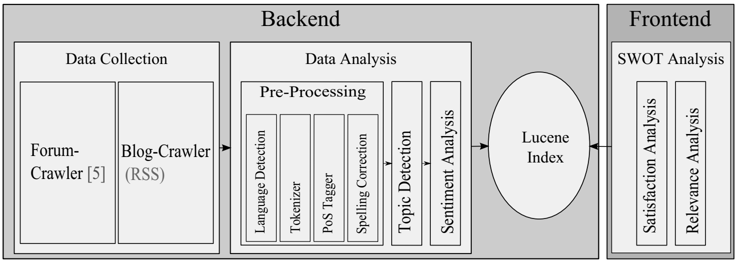
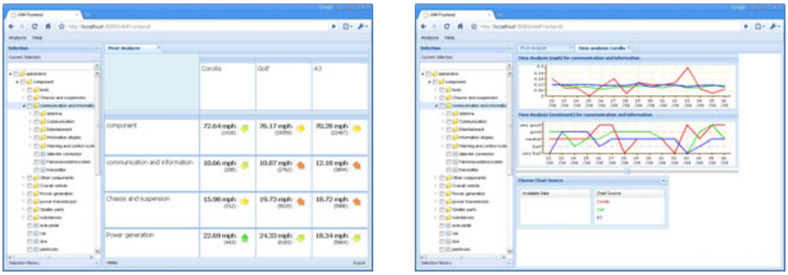


Fig. 2. Prototype architecture for the proposed recommendation system: The prototype tracks different Internet fora and weblogs and analyzes the downloaded data (section 3.1). The extracted information is stored in a Lucene data index for fast access and incremental updates, which is used by the frontend for the SWOT analysis (section 3.2).

The frontend, the customer interacts with, is a Rich Internet Application (RIA) based on GWT[3]. The user can drag products and product features to an analysis table in which the system automatically calculates all necessary frequencies using Lucene search queries. The recommendation system additionally classifies each satisfaction value to five different groups based on empirically determined borders. Each class is represented by one of five different arrows in order to give a quick quality impression (figure 3).

[3] http://code.google.com/webtoolkit/



Pivot Analysis

	Corolla	Golf	A3
component	72.64 mph (2418)	76.17 mph (13059)	70.28 mph (22467)
communication and information	10.66 mph (238)	10.87 mph (2762)	12.18 mph (3894)
Chassis and suspension	15.98 mph (312)	19.73 mph (5025)	18.72 mph (5986)
Power generation	22.69 mph (443)	24.33 mph (8183)	18.34 mph (5864)

(a) satisfaction analysis (b) time analysis

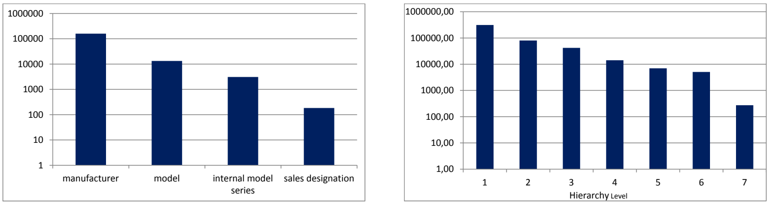
Fig. 3. Prototype use case: A potential customer wants to compare customer satisfaction for three different car models. After selecting features relevant to the user, the prototype calculates the satisfaction and the relevance index (a). Each product-feature combination can be analyzed over time (b).

Next to the abstract overview, the user can analyze the satisfaction and relevance indices over time to see quality changes during the last months and years. The user is able to list the corresponding comments to get linked to the original web pages.

4.2 Evaluation Results

In order to assess whether the proposed system performs a useful product advice, two different questions are analyzed. First, it has to be ensured, that there is enough data available to generate statistically relevant statements. Using the taxonomy structure, the average number of postings for manufacturers, products, product models and product features at different levels of detail can be quantified. The analysis shows that there is a large number of comments available for manufacturers and products (fig. 4 (a)). The sales designation instead can be found in an insufficient number. This is caused by the fact that sales designations in the automotive domain are specifying different product variations which in most cases are not subject of a problem description. Model names and internal model series are well mentioned. Product features on the other side are discussed in varying frequency: While feature comments are numerous for more abstract features (fig. 4 (b)) special ones are rarely mentioned. In this situation, the taxonomy approach does not fit. The user is neither mentioning a feature nor a synonym. To improve the system quality on detail level, it would be additionally necessary to link symptoms to all relevant product features, which on the other side would decrease data precision.

Ensuring the topics of analysis can be found in the proposed data source, the resulting product ratings have been compared to survey results of J.D. Power's "Initial Quality Study" IQS. This study is based on a 228-question battery designed to provide manufacturers with information to facilitate identifying problems. Using J.D. Power's quality index, it is possible to compare the system output to a well established quality measure. This is done by comparing 36



(a) product coverage

(b) topic coverage

Fig. 4. Coverage analysis: There is a large number of comments (y axis) available for manufacturers, models and internal model series. Sales designation instead is less useful due to small reliability (a). The discovery of product features depends on the hierarchical level (b). Very specific components are not mentioned in reliable number.

different cars of 10 different manufacturers in 2008. For each selected model, there are more than 400 user comments available so that each result is statistically reliable.

Due to different categorization methods, both measures are compared on the most abstract level. IQS distinguishes three different ratings: Initial Quality, Performance & Design and Predicted Reliability. Each one measures the quality relative to other cars in the market. The three parts are merged to one global car rating $\in [1;5]$. The proposed recommendation system distinguishes different features defined by the taxonomy structure. A comparable concept is the most abstract topic: *component*, which consists of all other product features.

The calculated correlation coefficient of both measures is 0.46. Thus, there is no perfect match between J.D. Power's IQS survey and the satisfaction index proposed in this work. Nevertheless there is some correlation. Due to different measure intervals, the significance of correlation can not be calculated directly. Therefore, the proposed satisfaction index $\in [-1;1]$ is rescaled to J.D. Power's measure interval $\in [1;5]$ using a simple linear regression, which does not affect correlation.

$$s_2(y_i, x_j) = 3.66 * s(y_i, x_j) + 3.31 \tag{3}$$

The correlation of both measures would be significant if the expected difference is 0. Using a t-test[4], H_0 ($\mu_0 = 0$) could not be rejected and thus, there is no evidence that there is a systematic difference. To minimize the Type II error, the t-test is calculated for other possible differences (table 1). The root mean squared error of both measures is 0.65.

There are some reasons for expecting small differences between J.D. Power's IQS and the proposed measure (e.g. ambiguities, irony, sarcasm, ...). The main reason is that J.D. Power uses a questionnaire while the proposed recommendation system only collects unrequested user feedback which not only contains information about hard problem facts but also soft facts (e.g. service quality). These customer expectations are implicitly available in the Internet data which

[4] Both measures are normally distributed, which is ensured using the Shapiro-Wilk Normality Test.

Table 1. t-test for significance calculation

H_0	t	two-tailed p-value	interpretation
$\mu_0 = 0$	-0.015	0.988	not rejectable
$\mu_0 = 0.25$	-1.833	0.073	rejectable for $\alpha = 10\%$
$\mu_0 = 0.5$	-3.650	0.001	rejectable for $\alpha = 1\%$
$\mu_0 = 1$	-7.284	$2.4 * 10^{-9}$	rejectable
$\mu_0 = 2$	-14.554	$2 * 10^{-19}$	rejectable

makes the data biased compared to J.D. Power's IQS. This fact changed the overall result, however, only moderately so that the proposed recommendation system tends to similar statements regarding the well established J.D. Power's Initial Quality Study.

5 Conclusion

Shopping systems can significantly increase the sales volume by providing product reviews created by other customers. Although it has been shown by different researchers that these reviews are not reliable in terms of product quality, potential customers trust in these comments. It is anticipated that this issue will decrease the long-term usage of the currently widely used recommendation system. In this work, an alternative data source is presented: social networks. Using different text mining techniques, user generated content of Internet fora, weblogs and other trustworthy social networks can be utilized to provide customers a mostly unbiased aspect-oriented satisfaction overview. Next to different abstraction levels, a potential customer can analyze quality changes over time. The applicability has been shown in the automotive domain.

The analysis method does not depend on the proposed natural language processing algorithms, so that the recommendation system can benefit from future research in the area of topic detection and sentiment analysis. Especially topic relevant algorithms could improve the recommendation system on product detail level.

References

1. Schafer, J.B., Konstan, J., Riedl, J.: Recommender systems in e-commerce. In: Proceedings of the 1st ACM Conference on Electronic Commerce, pp. 158–166 (1999)
2. Montaner, M., López, B., de la Rosa, J.L.: A taxonomy of recommender agents on the internet. Artificial Intelligence Review 19, 285–330 (2003)
3. David, S., Pinch, T.J.: Six degrees of reputation: The use and abuse of online review and recommendation systems. First Monday (July 2006); Special Issue on Commercial Applications of the Internet
4. Hu, N., Liu, L., Zhang, J.J.: Do online reviews affect product sales? the role of reviewer characteristics and temporal effects. Inf. Technol. and Management 9(3), 201–214 (2008)

5. Liu, J., Cao, Y., Lin, C.-Y., Huang, Y., Zhou, M.: Low-quality product review detection in opinion summarization. In: Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 334–342 (2007) (poster paper)
6. Danescu-Niculescu-Mizil, C., Kossinets, G., Kleinberg, J., Lee, L.: How opinions are received by online communities: a case study on amazon.com helpfulness votes. In: WWW '09: Proceedings of the 18th International Conference on World Wide Web, pp. 141–150. ACM, New York (2009)
7. Nielsen online, Buzzmetrics (May 2008), http://de.nielsen.com/products/documents/NielsenonlineBuzzMetrics200805%21.pdf
8. Ghose, A., Ipeirotis, P.G.: Designing novel review ranking systems: predicting the usefulness and impact of reviews. In: ICEC '07: Proceedings of the Ninth International Conference on Electronic Commerce, pp. 303–310. ACM, New York (2007)
9. Hu, N., Pavlou, P.A., Zhang, J.: Can online reviews reveal a product's true quality?: empirical findings and analytical modeling of online word-of-mouth communication. In: EC '06: Proceedings of the 7th ACM Conference on Electronic Commerce, pp. 324–330. ACM, New York (2006)
10. Kim, S.-M., Pantel, P., Chklovski, T., Pennacchiotti, M.: Automatically assessing review helpfulness. In: EMNLP '06: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pp. 423–430. Association for Computational Linguistics, Morristown (2006)
11. Universal McCann, Wave.3 - social media tracker (March 2008), http://www.universalmccann.com/
12. Bank, M., Mattes, M.: Automatic user comment detection in flat internet fora. In: DEXA Workshops, pp. 373–377. IEEE Computer Society, Los Alamitos (2009)
13. Cavnar, W.B., Trenkle, J.M.: N-gram-based text categorization. In: In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, pp. 161–175 (1994)
14. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proceedings of International Conference on New Methods in Language Processing (September 1994), http://www.ims.uni-stuttgart.de/ftp/pub/corpora/tree-tagger1.pdf
15. Schierle, M., Schulz, S., Ackermann, M.: From spelling correction to text cleaning - using context information. In: Data Analysis, Machine Learning and Applications. Studies in Classification, Data Analysis, and Knowledge Organization, pp. 397–404. Springer, Heidelberg (2008)
16. Smet, W.D., Moens, M.-F.: Generating a topic hierarchy from dialect texts. In: DEXA Workshops, pp. 249–253. IEEE Computer Society, Los Alamitos (2007)
17. Liu, B., Hu, M., Cheng, J.: Opinion observer: analyzing and comparing opinions on the web. In: WWW '05: Proceedings of the 14th International Conference on World Wide Web, pp. 342–351. ACM, New York (2005)
18. Popescu, A.-M., Etzioni, O.: Extracting product features and opinions from reviews. In: HLT '05: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 339–346. Association for Computational Linguistics, Morristown (2005)
19. Scaffidi, C., Bierhoff, K., Chang, E., Felker, M., Ng, H., Jin, C.: Red opal: product-feature scoring from reviews. In: EC '07: Proceedings of the 8th ACM Conference on Electronic Commerce, pp. 182–191. ACM, New York (2007)

20. Guo, H., Zhu, H., Guo, Z., Zhang, X., Su, Z.: Product feature categorization with multilevel latent semantic association. In: CIKM '09: Proceeding of the 18th ACM Conference on Information and Knowledge Management, pp. 1087–1096. ACM, New York (2009)
21. Schierle, M., Trabold, D.: Multilingual knowledge based concept recognition in textual data. In: Proceedings of the 32nd Annual Conference of the GfKl (2008)
22. Kano, N., Seraku, N., Takashi, F., Tsuji, S.: Attractive quality and must-be quality. The Journal of the Japanese Society for Quality Control 14(2), 39–48 (1984)
23. Wiebe, J.M., Bruce, R.F., O'Hara, T.P.: Development and use of a gold-standard data set for subjectivity classifications. In: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics, pp. 246–253. Association for Computational Linguistics, Morristown (1999)
24. Hatzivassiloglou, V., Wiebe, J.M.: Effects of adjective orientation and gradability on sentence subjectivity. In: Proceedings of the 18th Conference on Computational Linguistics, pp. 299–305. Association for Computational Linguistics, Morristown (2000)
25. Turney, P.D.: Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In: ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 417–424. Association for Computational Linguistics, Morristown (2002)
26. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: EMNLP '02: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, pp. 79–86. Association for Computational Linguistics, Morristown (2002)
27. Dave, K., Lawrence, S., Pennock, D.M.: Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In: WWW '03: Proceedings of the 12th International Conference on World Wide Web, pp. 519–528. ACM, New York (2003)
28. Remus, R., Quasthoff, U., Heyer, G.: SentiWS - a German-language Resource for Sentiment Analysis. In: Proceedings of LREC 2010 (2010)
29. Ferrucci, D., Lally, A.: UIMA: An architectural approach to unstructured information processing in the corporate research environment. Natural Language Engineering 10(3/4), 327–348 (2004)
30. Ferrucci, D.: Oasis unstructured information management architecture (uima), version 1.0. (2008), http://www.oasis-open.org/committees/download.php/28492/uima-spec-wd-05.pdf

Content-Based News Recommendation

Michal Kompan and Mária Bieliková

Institute of Informatics and Software Engineering, Faculty of Informatics and
Information Technologies, Slovak University of Technology,
Ilkovičova 3, 842 16 Bratislava 4, Slovakia
kompan05@student.fiit.stuba.sk, bielik@fiit.stuba.sk

Abstract. The information overloading is one of the serious problems nowadays. We can see it in various domains including business. Especially news represent area where information overload currently prevents effective information gathering on daily basis. This is more significant in connection to the web and news web-based portals, where the quality of the news portal is commonly measured mainly by the amount of news added to the site. Then the most renowned news portals add hundreds of new articles daily. The classical solution usually used to solve the information overload is a recommendation, especially personalized recommendation. In this paper we present an approach for fast content-based news recommendation based on cosine-similarity search and effective representation of the news. We experimented with proposed method in an environment of largest electronic Slovakia newspaper and present results of the experiments.

Keywords: news, recommendation, personalization, vector representation, user model, article similarity.

1 Introduction

There are plenty of news portals on the web. Renowned and influential portal contains hundreds of new articles from the whole world added daily. These articles cannot be easily accessed. For example users of the biggest Slovak news portal SME.SK spend daily approximately 16 min on the site, in usually two visits per day[1]. The amount of words on the websites has increased two times since year 2003. We can see this effect applied to links, pictures, tables, advertisements etc. More than 60% respondents participating in IDC research said, that they face up the information overloading in more than half of the time (see Fig. 1).

One of the quality criteria for a good news portal is time spending by reading considering the amount of useful information acquisition. It is extremely important to enquire new information as quick as possible. The importance of fresh news can be easily seen on various non-news portals, where various shorten top news can be found.

[1] Source www.aimmonitor.sk – Association of Internet Media.

F. Buccafurri and G. Semeraro (Eds.): EC-Web 2010, LNBIP 61, pp. 61–72, 2010.

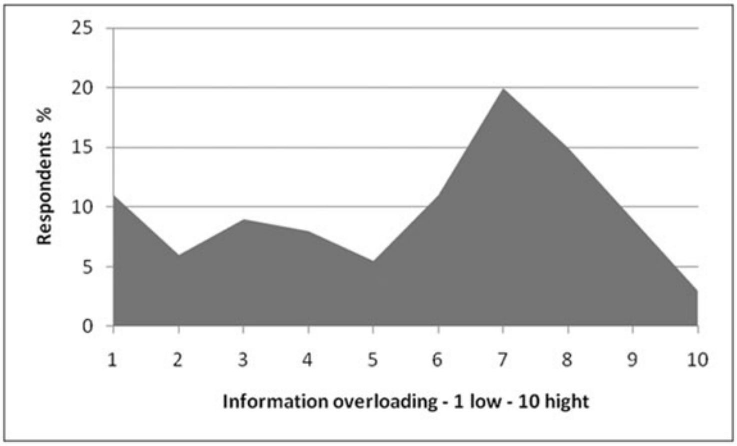


Fig. 1. Frequency of information overloading [IDC, autumn 2008, U.S., set of 500 respondents]

Considering much work already done in area of recommendation and personalization we should take into account dynamic nature of news and the volume of information flow. We propose a method for content-based news recommendation, which employs our devised effective article representation. This representation is important when similar articles are computed. Fast similarity estimation plays the critical role in the high changing domains as news portals are. It is necessary to process a new article as fast as possible and start to this article recommendation, because of the high information value degradation. Finally we use these similar articles to create recommended content based on the implicit user model.

Our content-based method for recommendation is based on three steps (see Fig. 2):

1. computing article similarity,
2. creating a user model and
3. the recommendation based on the first two steps.

In the article similarity step it is necessary to pre-process every article to reduce word space. The article is represented in an effective vector representation, which is used in the similarity computation. As a result of the article similarity step we obtain a list of similar articles for every article in the dataset.

A user model is created based on implicit feedback extracted from server logs by identification of visited and recommended articles for particular user (we compute it for unique cookie). Finally, the recommended content from both similar articles and the user model is created. We give more detailed description for every step in the following sections.

The paper is structured as follows. Section 2 describes state of work in the recommendation domain. In section 3 we provide overview of proposed vector structure representation. Section 4 describes our recommendation method. The evaluation of proposed method is described in section 5.

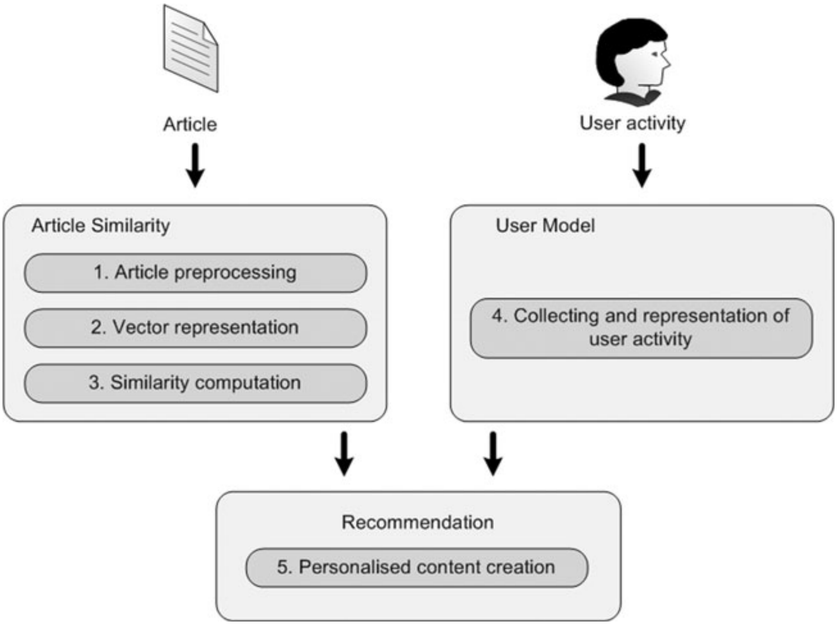


Fig. 2. Proposed news recommendation method

2 Related Work

The recommendation is one of the actual research topics nowadays. Two basic approaches of the recommendation exists [10]. Traditional collaborative filtering accounts a social element. Users are grouped based on their preferences, habits or the content ranking. The problem of personalized recommendation is reduced to the finding similar users and recommending such new items to the users, which were visited and high ranked by other similar users.

The second class of recommenders is based on the content-based filtering. The history of content-based filtering is connected with information retrieval and information research [1]. The main goal is to identify two similar items-create "cluster" of sites instead of users. It is necessary to map user profiles (user models) to specific site clusters. This type of filtering is successful in well structured domains like movies, news [11].

These two approaches are widely used and mixed together, which usually brings better results [9,3]. For example, we can find similar sites and then estimate user rank prediction for sites, which were not visited. Also the combination of various approaches for every type is possible e.g. Google News [5]. The main problem in the content-based filtering is effective and enough expressive representation of items (or articles). This is often done by means of text summarization [4], keywords extraction or by various categorization models [7]. These techniques are commonly used in recommending documents in English and cannot be easily applied in other languages. Keywords extraction and summarization brings better results as other methods but it is more time consuming. These methods cannot represent non-text documents without its actual modification.

There are several recommender systems in the news domain. The problem within this domain which is rather similar to other business domains is extremely large amount of dynamically changing data. This causes that the recommendation is not provided directly over the whole data set, when content-based recommendation is used [16,17]. OTS system [16] uses association rules to create "preference table" for every user. When there are a lot of new documents added daily, usual way is to compute recommendation lists off-line [17]. TRecom is a promising method for content-based recommendation, when uses binary-tree representation of similarity and user's preferences [18]. Brusilovsky [15] has shown that explicit filled and open user model in the news domain brings usually worse results. Some systems have involved user location into recommendation systems, where recommendation list is created depending on the user location [6].

3 Article Similarity Computation

For fast similarity estimation we propose effective vector article representation. This representation consists of six basic parts:

1. *Title.* It contains lemmatized words from article title (approx. 5 words – 150 000 Slovak article corpus). This should be good describing attribute in the most occurrences.
2. *TF of title words in the article content.* We use term frequency to compute article relevance. If the article name is an abstract and does not correspond to the article content, we can easily reveal this situation (using a threshold). Term frequency is computed as follows:

$$tf_i = \frac{n_i}{\sum_k n_k} \qquad (1)$$

   where tf_i is term frequency for term i (term from article title) and n_i is number of occurrences of term i in the document (article content) and $\sum_k n_k$ is the sum of numbers of all terms in document.
3. *Names and Places.* We extract names and places from article content. There exist several names or places extractors for English language. We use simple approach to detect these items. As name or place is marked word starting with an upper letter and there is no sentence end before (dot, question mark etc.). This method extracts most of names and places occurred in the article (precision = 0.934, recall = 0.863).
4. *Keywords.* We store 10 more relevant keywords. Several news portals define list of keywords for every article. These keywords are unfortunately on various abstract levels for various news portals. We introduced our own keywords list based in TF-IDF computation (150 000 Slovak news articles from news portal SME.SK). We adopt a part of speech tagging and removed any words except nouns and names[2].

[2] JULS dictionary – Slovak Academy of Sciences.

5. *Category.* It consists of "tree-based" category vector with weights. This vector is constructed based on specific news portal structure hierarchy (optional). This is useful, when not enough similar articles are found. The weight for every category is computed as:

```
n=1
For i=|Category| downto 0 do
 weight[i]=1/n
 n=n*2
end
```

6. *CLI – Coleman-Liau Index.* It provides information of understandability of the text. This vector part is not important for standard similarity computation, but it is important for the results rearrangement. Our hypothesis is that the user wants to read articles of similar level of writing style. This method is able to distinguish between two articles with similar title and different content ("Jaguar" – animal vs. car). CLI can be easily computed based on this formula [8]:

$$CLI = 5.89 \times \left(\frac{characters}{words}\right) - 29.5 \times \left(\frac{sentences}{words}\right) - 15.8 \qquad (2)$$

When using this article representation, we can store an article by the vector no longer than 30 items in most of occurrences. Example of proposed representation is given in Table 1.

Table 1. The example of vector article representation (in Slovak)

Vector part	**Weights**
Title	transplantácia_0.5 tvár_0.5
TF of title words in the content	transplantácia_0.0178571428571429 tvár_0.0714285714285714
Category	Sme.sk_0.5 PRESS_FOTO_1.0
Keywords	klinika_0.0357142857142857 povrch_0.0178571428571429 nos_0.0178571428571429 zub_0.0178571428571429 nerv_0.0178571428571429 svalstvo_0.0178571428571429 pacientka_0.0178571428571429 rozsah_0.0178571428571429
Names/Places	Cleveland_1
CLI	0.2543

For the purpose of similarity computation, we use cosine similarity [14], which is widely used in the information retrieval tasks. Our vector consists of 6 subvectors with weights so there is need to extend standard cosine similarity as:

$$similarity = \frac{\sum_{j=1}^{m} \sum_{i=1}^{n} a_{ji} b_{ji}}{\sqrt{\sum_{j=1}^{m} \sum_{i=1}^{n} a_{ji}^2} \sqrt{\sum_{j=1}^{m} \sum_{i=1}^{n} b_{ji}^2}} \quad (3)$$

where m is number of vector parts (6 in our method) and n is number of vector items. The similarity definition in recommendation systems is a difficult task. We can define similarity based on news content (like plagiarism task), or based on "topic" or "affair". This is extremely important when a recommendation list is created. Our method respects each of these types. We can easily redefine our similarity with simple changing the weights for vectors parts and adjust it for various recommender methods.

3.1 News Pre-processing

Text pre-processing holds an important role in the process of similarity search, because it can significantly reduce word space. This part of the process is high language dependent. We provided experiments in the Slovak language, which is one of the most complicated languages as it is flective language (declension of nouns, verbs etc.). The architecture of the system is flexible, so pre-processing for Slovak language can be easily replaced by other languages and their methods (e.g. Porter algorithm[3]). For the speed of next computations, plays pre-processing a critical role. There is need to maximum reduction of article words dimensions.

The first task is to remove stop-words. We used static list, which can be replaced by TF-IDF output [12]. This method can identify commonly repeated words over the dataset.

As the main part of the pre-processing of Slovak language articles we used lemmatizing of the text. There is problem with algorithmic solution for this process, which can be solved by using dictionary of lemmas. For the purpose of lemmatization we use dictionary of lemmas (600 000 records). The result we receive is lemmatized (basic form) bag of words for every article.

It is necessary to note that we remove any punctuation except sentences ends. We use dots as a fast name or place indicator – when we check if there is a dot before an uppercase letter, and if not, it is probably personal name, or place etc. Names and place extractors are standard tasks in information retrieval. As we mentioned above, this approach brought sufficient results and can be simply substituted by more sophisticated methods.

After keywords extraction we do not need the whole article content anymore. We can safely delete all words except the title words obtained in the article content. Then for every processed article we save following list of words:

- Lemmatized article Title
- Lemmatized words from Content (which were included in the Title)

[3] The Porter Stemming Algorithm page maintained by Martin Porter. www.tartarus.org/~martin/PorterStemmer

- 10 most relevant Keywords
- List of Names and Places

Pre-processing methods we described above can significantly reduce number of words stored for every article up to 80%.

4 Recommendation

The most important part of proposed method is the recommendation step (see Fig. 3). For recommendation creation we need two lists as an input:

1. The first list consists of 10 most similar articles for every article computed as we described above.
2. The second list is list of visited articles for every user (in our system based on cookie). In this list we need to distinguish between articles visited but not recommended to users and articles visited and recommended before which can be easily done by extending article URL with special attribute.

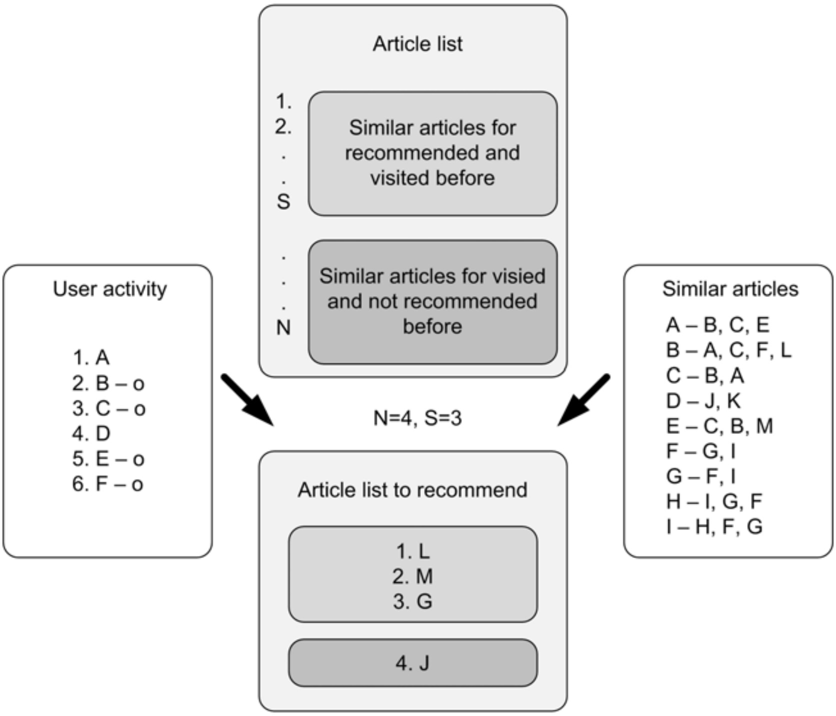


Fig. 3. Steps of the recommendation method

Firstly we define the number of articles to recommend (length of list to recommend). As we can see in Fig. 3 list of articles to recommend consists of two sub-lists:

- list of similar articles for visited and not recommended (S),
- similar articles for visited and before recommended (N-S).

The ratio of this list is dynamically computed as:

$$S = N\left(1 - \frac{Nr}{V}\right) \tag{4}$$

where S is number of similar articles for visited and not recommended articles, N is number of articles to recommend. Nr represents number of visited not recommended articles during the last session and V is number of visited articles together. For the proposed method, two sessions are distinguished as 1 hour break between the visits.

Recommendation is then computed for every part separately as follows:

```
foreach cookie do
 visited = get visited articles list
 visitedRec = get visited and recom. articles list
 foreach visited do
  if randomNum > random treshold
   listPart1 = get first non visited article from computed
               similarity list
  else
   listPart1 = get random non visited article
  end
 end
 foreach visitedRec do
  listPart2 = get first non visited article from computed
              similarity list
 end
 listToRecommend = listPart1[1..N] + +listPart2[1..M]
end
```

When there is not enough user activity (does not mean “cold start” we use random article assignment. In this manner we can flexible react to user’s most recent preferences. For the list of recommended and visited articles we also introduced a coincidence – where the user obtains a random article to the recommendation list.

Fig. 3 presents an example of recommended content creation. We have a list of user activities where “-o” attribute indicates whether the article was or was not recommended. Based on this we obtain list of visited articles and list of visited and before recommended articles. Then when we want to recommend 4 articles (*N=4*), we will obtain ratio 3:1 for sub-lists (similar articles for visited and recommended, similar articles for visited and not recommended before).

As we can see in our example, we have 4 visited and recommended articles *B, C, E, F*. We have found non visited article from the list of similar articles for every of these 4 articles. There is only one non visited similar article *L* for article *B*. This is repeated until the "before recommended" list is full. In the case when there do not exist non visited article in the similar list (article *C*), we skip this article, because the user saw all relevant articles for this "topic" already.

In our example there are 2 non recommended but visited articles, but there are no non-visited articles for *A* – method will skip this article and will recommend first non visited for article *D*. In this manner we obtain a full list of 4 articles to recommend.

Dynamical computation of the ratio between sublist allows us to adapt for actual user activity and preferences. If the user is not interested in recommended articles and he uses other portal navigation, the size of the first sub-list is decreasing while second part will increase respectively.

We store the "article age" for every recommended article. This number represents how long have been article recommended. If user does not visit this article for a defined time (number of recommendations) is this article deleted from the recommendation list as not interesting.

User activity list consists of pairs cookie – visited article. We use implicit user model representation, where there is no need to involve users into various forms completing or need of logging etc.

5 Experimental Results

We implemented proposed method within the news recommendation system in the research project SME-FIIT [2].We evaluated the similarity computation over 10 000 articles from the Slovak news portal SME.SK, which is equivalent to one week time period. For this window we are able to estimate the similarity in 2-3 seconds (2,6 MHz Pentium, 4Gb RAM, Ruby). The pre-processing takes approximately 20 seconds for the whole dataset. Then for the new article, when pre-processing is necessary, the whole computation process takes approximately 22s. When we need only re-estimate similarity with changed vectors parts weights is this process really fast as we mentioned above.

The accuracy of the similarity computation method was computed based on two datasets. The first one consists of 1 000 articles from news portal SME.SK. Every article from the dataset had assigned at least one similar article. These similar articles were obtained from the news portal, where there are mostly one or two similar articles quoted in the article footer. These similar articles are obviously chosen by the article author, which does not mean that there are not more similar articles.

The second dataset was the manually annotated dataset, which consists of 100 articles in 5 levels of similarity, so we obtained 10 000 article pairs with similarity level. Our method computed the list of similar articles for every article in the dataset. We compared these datasets to our method – the list of similar articles computed by our method and the list of similar obtained from one of two datasets with respect to order (more similar articles first). We calculated precision and recall and F-Score for every dataset and the method. Results were compared to standard text mining method TF-IDF (whole article content) as shown on Table 2.

Table 2. Similarity computation evaluation

Dataset	**SME.SK**		**Manually annotated**	
Method	Our method	TFIDF	Our method	TFIDF
Precision	0.165	0.091	0.700	0.511
Recall	0.202	0.117	0.816	0.587
F-score	0.182	0.102	0.753	0.546

The dataset SME.SK is created based on "similar article" data (none, one or two) in the articles footers. These similarities are assigned by article's authors intuitively and often this choice does not mean not the only possibility but also one of the best matching articles. This is reflected in the results as we obtained only 0.182 F-score. Providing manual check we found out that our method in most cases founded more similar (and relevant) articles as the authors assigned. This indicates that manual similarity articles list creation by the article authors can be improved by our method.

We also computed standard deviation based on similarity levels. We mapped cosine similarity range $< 0, 1 >$ to five similarity levels used in our manually annotated dataset. The worst standard deviation we obtained – 1.21 "similarity level" is an acceptable rate in the field of news recommendation.

6 Conclusion

In this paper we provided the overview of concise and high representative article vectors, which are used for similarity search and content-based recommendation in a large and dynamically changing datasets. A key future of our method is short article representing vector. In pursuance of these vectors can be computed similarity between articles (text or non-text content) in a fast way. Every article vector consists of 6 sub-vectors based on article part used for their construction. Every part has its own weight, which can be dynamically changed to rearrange similar articles list to enable fast personalization. Proposed approach as such is language independent so it can be easily adapted for other languages.

Weights were found by using evolution algorithms for every vector part, to obtain the best result. As an example, use of proposed representation brings 4 times better precision than use only article title, and at least 1.4 time better results as use only keywords. By considering the category part we improve precision only 1.15 time, but on the other hand it can be useful when "no similar" article is in the dataset.

Once the similarity is computed, further recommendation is created. User preferences are collected implicitly via server's logs. A recommended list consists of two sub lists, where the first one represents similar articles to the visited and already recommended. The second sublist is based on similar articles for visited but not recommended before. In this way we can easily adapt to user preferences. The ratio between these two sub-lists is dynamically computed.

Proposed vector representation is a promising method for the fast news similarity computation to allow a real time recommendation. We plan to make improvements on the precision and the recall, for example by using more sophisticated keywords extraction methods etc. and evaluate whole recommendation method by its implementation to existing news portal. Furthermore we expect significantly better results when combining our approach with TRecom [18] method or collaborative recommendation [13].

Acknowledgements

This work was partially supported by the grants VEGA 1/0508/09, KEGA 028-025STU-4/2010 and it is the partial result of the Research & Development Operational Programme for the project Support of Center of Excellence for Smart Technologies, Systems and Services, ITMS 26240120029, co-funded by the ERDF.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering 17(6), 734–749 (2005)
2. Barla, M., Kompan, M., Suchal, J., Vojtek, P., Zeleník, D., Bieliková, M.: News recommendation. In: Proc. of the 9th Znalosti, VSE Prague, pp. 171–174 (2010)
3. Bouras, C., Tsogkas, V.: Personalization Mechanism for Delivering News Articles on the User's Desktop. In: Proc. of the 4th int. Conf. on Internet and Web Applications and Services, ICIW, pp. 157–162. IEEE Computer Society, Washington (2009)
4. Dakka, W., Gravano, L.: Efficient summarization-aware search for online news articles. In: Proc. of the 7th ACM/IEEE-CS Joint Conf. on Digital Libraries, JCDL '07, pp. 63–72. ACM, New York (2007)
5. Das, A.S., Datar, M., Garg, A., Rajaram, S.: Google news personalization: scalable online collaborative filtering. In: Proc. of the 16th Int. Conf. on World Wide Web, WWW '07, pp. 271–280. ACM, New York (2007)
6. Chen, C., Hong, C., Chen, S.: Intelligent Location-Based Mobile News Service System with Automatic News Summarization. In: International Conference on Environmental Science and Information Application Technology, vol. 3, pp. 527–530. IEEE Computer Society, Washington (2009)
7. Kou, H., Gardarin, G.: Keywords Extraction, Document Similarity and Categorization. Tech.rep., PRiSM Laboratory of Versailles Univ., No.2002/22 (2009)
8. McCallum, D.R., Peterson, J.L.: Computer-based readability indexes. In: Proc. of the ACM '82 Conf., pp. 44–48. ACM, New York (1982)
9. Melville, P., Mooney, R.J., Nagarajan, R.: Content-boosted collaborative filtering for improved recommendation. In: Proc. of 18th National Conf. on Artificial Intelligence, Edmonton, Alberta, Canada, pp. 187–192. AAAI, Menlo Park (2002)

10. Mobasher, B., Anand, S.S.: Intelligent Techniques for Web Personalization. In: Mobasher, B., Anand, S.S. (eds.) ITWP 2003. LNCS (LNAI), vol. 3169, pp. 1–37. Springer, Heidelberg (2005)
11. Pazzani, M., Billsus, D.: Content-based recommendation systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 325–341. Springer, Heidelberg (2007)
12. Ramos, J.: Using TF-IDF to Determine Word Relevance in Document Queries. Tech. rep., Departament of Computer science. Rutgers University (2000)
13. Suchal, J., Návrat, P.: Full text search engine as scalable k-nearest neighbor recommendation system. In: Proc. of the Artificial Intelligence in Theory and Practice 2010. World Computer Congress. Springer, Boston (2010)
14. Tata, S., Patel, J.M.: Estimating the Selectivity of tf-idf based Cosine Similarity Predicates. SIGMOD Record 36, 7–12 (2007)
15. Wongchokprasitti, C., Brusilovsky, P.: Newsme: A case study for adaptive news systems with open user model. In: Proc. of The Third Int. Conf. on Autonomic and Autonomous Systems, ICAS 2007, pp. 69–75. IEEE Press, Los Alamitos (2007)
16. Wu, Y., Chen, Y., Chen, A.L.: Enabling Personalized Recommendation on the Web Based on User Interests and Behaviors. In: Proc. of the 11th Int. Workshop on Research Issues in Data Engineering, RIDE, p. 17. IEEE Computer Society, Washington (2001)
17. Yoneya, T., Mamitsuka, H.: Pure: a pubmed article recommendation system based on content-based filtering. Genome informatics. In: International Conference on Genome Informatics, vol. 18, pp. 267–276. Imperial College Press, London (2007)
18. Zeleník, D., Bieliková, M.: Dynamics in hierarchical classification of news. In: Proc. of the 4th Work. on Intel. and Knowledge Oriented Technologies. Equilibria, pp. 83–87 (2009)

Towards a Lawfully Secure and Privacy Preserving Video Surveillance System

Aniello Castiglione[1,⋆], Marco Cepparulo[1]
Alfredo De Santis[1], and Francesco Palmieri[1,2]

[1] Dipartimento di Informatica ed Applicazioni "Renato M. Capocelli"
Università degli Studi di Salerno, I-84084 Fisciano (SA), Italy
Tel.: +39089969594
castiglione@acm.org, mcepparulo@gmail.com, ads@dia.unisa.it
[2] Università degli Studi di Napoli "Federico II"
I-80126 Napoli, Italy
francesco.palmieri@unina.it

Abstract. Privacy protection and content confidentiality in video surveillance are new challenging security issues where the basic requirements are still under discussion. They are truly interdisciplinary topics pertaining several sectors of the information society (Finance, Homeland Security, Healthcare, etc) that require inputs from legal experts, technologists, privacy advocates and general public. This work presents a novel video surveillance system that provides content confidentiality and distributed trust by using a hybrid cryptosystem based on a threshold multi-party key-sharing scheme. Due to the flexibility of the underlying video content access-control scheme, such approach can handle the problem of users loosing their private shares as well as dynamically adding new users that can participate to content reconstruction. The system can be efficiently implemented on low-cost special purpose devices.

Keywords: Secure Video Surveillance, Secret Sharing, Key-Escrow, Video-Escrow, Privacy-Preserving Video Recording, Secure CCTV, Secure Video Recording, Crypto Camera, Homeland Security.

1 Introduction

Video surveillance systems are now widely deployed in many strategic places such as airports, banks, public transportations or busy city centers, where they assume a strategic role for a variety of critical tasks like personal safety, traffic control, resource planning and law enforcement. However, although people usually appreciate the sense of increased security brought by such technologies, the proliferation of video cameras used for surveillance purposes has introduced severe concerns about the privacy and trustiness of the captured data. Also, the introduction of wireless cameras while appealing, since it makes the deployment and relocation of the devices very easy and cost-effective, represents another

[⋆] Corresponding author.

F. Buccafurri and G. Semeraro (Eds.): EC-Web 2010, LNBIP 61, pp. 73–84, 2010.

great risk factor for the security of the transmitted video streams. Visual data may be intercepted by illicit parties for a use that is not the originally intended one, or may be maliciously manipulated to hide some evidence and/or introduce some fake one. For example, in a health-care scenario interception of images by outsiders compromises patient privacy rights, as well as in a banking, industrial or military environment where the fraudulent replacement of a camera with a tricky device or the manipulation/tampering of its output can be used to hide an hostile or illegal activity.

Therefore, in any modern video surveillance system there is a need to introduce a reliable way to protect, with lawful enforcements, the produced images and videos starting from their source devices. In order to avoid physical attacks, electronic means should be employed in addition to more robust ad hoc architectural features. Traditionally, the use of encryption technologies is required [1] to provide confidentiality in specific applications and to deny third party access to their content. However, it has been noted that applying robust asymmetric cryptosystems on visual data to real-time applications further exacerbates the processing power requirements (see [2], [3], [4] and [5]) and introduces several impairment factors such as packet size expansion, ciphering latency and jitter, adversely conditioning the overall video quality.

One way to avoid these problems is to use hybrid cryptography, with asymmetrical key-agreement procedures based on X.509 certificates for initial end-to-end authentication between the communicating parties and fast symmetric data stream encryption, needed for satisfactory performance. Such a hybrid approach relies on the use of an optimal combination of both asymmetric and symmetric algorithms, by taking advantage of their strengths and peculiarities to achieve an acceptable level of security. It is also necessary for a few privileged parties, e.g., police or government agencies, to access the data stored in the Surveillance Server. However, it is not desirable that to a single party is granted full access right to the data since there is a danger of a malicious power party misusing her/his privileges and/or compromising system security.

In order to safely distribute and refresh encryption keys and periodically check integrity of the cameras, a key-distribution solution based on the threshold secret sharing by Shamir [6] could be chosen. Starting from these ideas and concepts, this paper presents an innovative network-based video surveillance solution that meets all the above legitimate privacy and security needs ensuring that the recorded video material will be only available to a subset of mutually trusting authorities under exactly defined policies, agreements and circumstances. The proposed solution is independent from both the used image/video compression and encoding algorithms, thus allowing the use of standard video encoders/decoders together with industrial strength smart cameras that can efficiently output encrypted video. To demonstrate the validity and effectiveness of the proposed architecture, the authors analyse not only the privacy protection capabilities and coding efficiency features, but also its resistance against both brute-force and error concealment attacks, resulting in the undeniable evidence of a really secure and robust video surveillance framework. Strong authentication of video sources, integrity protection of the stored data and privacy preserving capabilities respect to a subset of cooperating authorities lower than a predetermined threshold, guarantee the enforcement of common lawful requirements in video surveillance scenarios.

2 The Architectural Scheme

The system is composed of several surveillance cameras connected through the network to a Surveillance Server, which controls all the activities of the cameras, and to a data collection and archival server (DVR) which record the video streams sent by the cameras (see Fig. 1). Video cameras capture either single pictures or video streams and are able to transmit them to the DVR Server in real-time on an Ethernet connection by using common IP-based transport protocols.

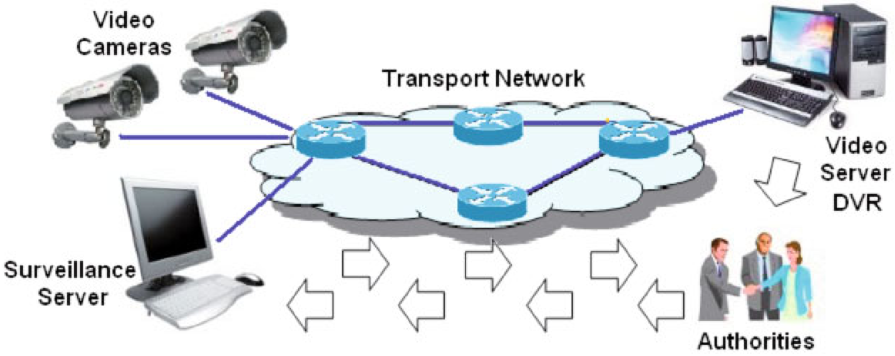


Fig. 1. The overall architecture

To ensure lawful evidence and reliability of all the captured video data, it is important to demonstrate that each video stream stored on the DVR has not been altered, before or during the transmission, and that its capturing source are properly computed and transmitted. In doing this the system need to enforce, in advance, mutual authentication between any camera and the servers to ensure strong identity trust between the communicating parties and hence to certify the origin of any service/control transaction or transmitted/received information. To ensure timing consistency, all the system components should be synchronised on a common time reference (e.g. using the NTP protocol [7]).

In a non-authenticated environment, external elements can insert themselves in the data path between cameras and servers and then collect and disrupt/corrupt the information through eavesdropping or Man-In-The-Middle attacks. Furthermore, since the authors intentions were to guarantee strong security and privacy protection, it is important to reliably encrypt all the recorded material together with the key-exchange/notification information before transmitting/storing them.

The order according to which the encryption and video encoding activities have to be combined needs particular attention for the success of the whole security framework. Indeed, the use of encryption on a compressed video channel is definitely not straightforward. The typical encoders implement a compression/decompression process heavily based on the assumption that the input signal will be a sequence of visual image frames. Except for a few rare cases, the difference between two successive video frames is small (many times the background is unchanging or panning in a predictable manner).

For example, the MPEG-4 and H.264 compressed encoding standards take advantage of this by outputting the difference between frames within specific objects called P-frames. Complete frames, called I-frames, are only produced periodically. The period within the I-frames defines the GOP (Group Of Pictures) length. Since the difference between frames is usually small, MPEG-4 does a good job in minimizing the video data close to its entropy value. Clearly, if the video signal were encrypted before arriving to the encoding block, it would not satisfy the expected characteristics because it would be randomized by the encryption process. Hence, it would fail to go through the encoding process with sufficient accuracy. After the encryption, the video data is sent to the DVR Server where it is stored to be successively accessed by cooperating agencies.

3 The Encryption Framework

The two most important security requirements of the proposed video surveillance system, namely trust on video capture devices and content confidentiality, seem to be practically in contrast with each other. In fact, the techniques based on digital signatures, certificates and PKI infrastructures needed to ensure strong authentication between the cameras and the servers, are absolutely unusable also for the purpose of ensuring confidentiality due to their performance and quality impacts on the encoded video streams. Consequently, the framework uses a practical secure solution to achieve both lawful identity enforcement and real-time performance in video encryption employing hybrid cryptography, with asymmetrical key-agreement procedures based on X.509 certificates for initial end-to-end authentication between each camera and both servers (DVR and Surveillance), together with fast symmetric data stream encryption needed for reasonable performances.

Finally, in order to enforce the cooperation of several users in the decryption process, the corresponding decryption secret is shared among a group of users/agencies. Hence each user, for decrypting video material, needs his share of the whole encryption key, share which is stored on the Surveillance Server in encrypted format. Note that the full encryption key is never stored in the system or present during intermediate results of the decryption process.

Keys that are used for encrypting videos must be chosen intervalwise and periodically changed to minimise risks of access to videos through cryptoanalysis techniques. Since the encryption keys are periodically generated by each camera, all the encrypted shares generated by such keys must be transmitted to the Surveillance Server on each key change.

3.1 The Symmetric Encryption Solution

The most undemanding solution is to simply encrypt the entire data stream with a general-purpose symmetric key algorithm such as AES. This is often referred to as the *naive* solution. The advantage of this method is that not only makes implementation easy, but it allows code modularity enabling straightforward changes in encryption algorithms, in key-distribution framework and even in the video codec. However, a clear disadvantage is the computational overhead of the cryptographic operations. In particular, in a system that uses hardware for decoding, a software decryption solution could become a bottleneck. An acceptable alternative is using *selective lightweight encryption*, where, only parts of video content are encrypted, thereby reducing the computational burden at the cost of some acceptable security tradeoffs. This method is generally suitable for video surveillance systems because the full content of the video is not critical. For example the Video Encryption Algorithm (VEA), developed by Qiao and Nahrstedt [8], while applying a standard symmetric encryption algorithm to the fully encoded video stream, is based on the statistical properties of MPEG for reducing the amount of data that is actually encrypted and relies on the Shannon's principle of *selective permute-then-encrypt* to keep the security at an acceptable level. In the VEA scheme, a chunk of the I-frame is divided in two halves. Both the halves are XORed and stored in one half. The other half is encrypted with a symmetric encryption algorithm such as DES or AES. This can yield an almost 50% gain in performance over the naive solution.

The encryption solution proposed in this paper is based on the VEA scheme with some additional features introduced to avoid byte modifications in the video ciphertext and to enforce the order of video chunks chaining.
The overall scheme can be described by the following steps:

1. each I-frame chunk, described as $a_1, a_2, a_3, \cdots, a_{2n-1}, a_{2n}$ is partitioned in two byte streams $a_2, a_4, \cdots, a_{2n}$ and $a_1, a_3, \cdots, a_{2n-1}$ respectively associated to the even and odd bytes;
2. the above byte streams are XORed bitwise resulting in the stream $c_1, c_2, \cdots, c_{n-1}, c_n = a_2, a_4, \cdots, a_{2n} \oplus a_1, a_3, \cdots, a_{2n-1}$;
3. a proper ciphering function $Enc(.)$ is chosen to encrypt the even stream $a_2, a_4, \cdots, a_{2n}$ so that the resulting ciphertext stream has the form $c_1, c_2, \cdots,$ c_n, $Enc(a_2, a_4, \cdots, a_{2n})$. The decryption mechanism at the other end of the communication channel consists in applying the deciphering function $Dec(.)$ with the appropriate key to the second half of the ciphertext to obtain the first half of the original sequence, and XOR this result with the first half of the ciphertext to obtain the other half of the original sequence. Clearly, if $a_2, a_4, \cdots, a_{2n}$ has no repeated patterns, then the overall ciphertext secrecy depends on function $Enc(.)$ since $a_2, a_4, \cdots, a_{2n}$ is a one-time pad;
4. finally, since it is necessary to operate with an unvarying transform on fixed-length groups of bits (the two halves blocks), the whole encryption framework is structured according to a block ciphering scheme. Specifically, a Propagating Cipher Block Chaining (PCBC) is applied to cause small changes in

the ciphertext to propagate indefinitely when decrypting, and hence to ensure that any manipulation of the ciphertext would damage all subsequent ciphertext. To further enforce the ciphertext integrity, a blockwise HMAC [9] may be added.

The encryption function *Enc*(.) has been implemented according to the AES symmetric encryption scheme with a configurable m-bits secret (at least 128 bit) calculated obeying to a key-sharing algorithm.

An alternative method for the key-management and for the encryption of the media would have been the one adopted by the modern digital broadcasting televisions, which use several Conditional Access (CA) systems. The variuos CAs, by way of summary, scramble the video data by using very small keys called "Control Words" (CW) periodically generated (for example every 10 seconds in the latest systems like Videoguard or Nagravision) and sent to the customers. Such CWs are decrypted by using a key which is updated less frequently (usually on a monthly basis) and stored on the smartcard of the customer. The Decrypted Control Words (DCW) are used to decrypt the stream only by the customers who own an official smartcard of the PayTV broadcasting the video content. The PayTV scenario is remarkably different from the one of the video surveillance because the CWs are used only for the real-time decryption and destroyed after their use. On the contrary, in a secure video surveillance system, the encryption keys must be preserved in order to decode the video data subsequently. Moreover, in a PayTV system the scrambling is secure if and only if the decryption keys (i.e. the DCW) are updated repeatedly (i.e. every 10 seconds). A secure video surveillance system makes use of a strong cryptosystem (like DES or AES) and consequently is more secure than the scrambling used by the PayTVs.

3.2 End-to-End Authentication and Session Setup

The cameras are responsible of the frames acquisition (in the YUV format), which are subsequently encoded in video frames (in the MPEG-1 format) and encrypted on-the-fly by using the above symmetric encryption algorithm. The encryption key K is generated on-board by each camera in a pseudo-random way and is never stored on the device. The main task of each camera is the generation of the session keys. The session keys are directly derived from the encryption key K by using a secret sharing scheme. In other words the session keys are pieces of the main key K that has been split among several authorities. Each authority owns a couple of private/public keys. Each camera owns the public keys of the authorities and, after obtaining the session keys (starting from the random encryption key K), encrypts each session keys by using the public keys of the various authorities. For the sake of supporting end-to-end authentication, the DVR and the Surveillance servers have a couple of private/public keys. Such public keys are present on each camera and are used to support mutual authentication between the servers and the cameras and thus to create a secure authenticated channel by means of the TLS/SSL protocol. The shares are never stored on the cameras. Such shares are handed over to the Surveillance Server by using the secure authenticated channel and are stored always in encrypted format. Only the authority that owns the corresponding private key of the public key responsible of the encryption would be able to decrypt that part of the encryption key K. The shares are vital for future decryption of the video. The Surveillance Server, besides storing the shares in encrypted format, is responsible for the authentication of the cameras during the start-up phase or in case of a new camera joining the architecture. Another prominent task of the Surveillance Server is to distribute the proper shares to the authorities in case of decryption of a video.

After the above phases, that substantially perform the setup of the system, the cameras would be able to send the encrypted data to the DVR server, assured that the video cannot be viewed unless the participation of a number of authorities suitable for the Shamir scheme. For the authentication of the cameras with the Surveillance Server there are two different scenarios with respect to the security of private information:

- each camera holds a smartcard, PIN-protected, where there is the secret information used for the authentication with the Surveillance and DVR servers, as well as the public keys of the authorities and of both servers. In such a way, no secret information is stored on the cameras thus avoiding the risk of accessing them in case of theft or abuse. The advantage of this method is the high level of security at the expense of usability because, each time the entire system (or just a single camera) is started (or plugged into the architecture), it is necessary to perform the setup and provide the smartcard with the PIN in order to unblock the private information needed for the authentication;
- a secret information (useful to start a challenge-response authentication protocol) may be derived from a particular unique data which is normally on the camera, like for example its serial number or a generic string present on its firmware. In such a case, the setup phase is completely automatic and no human intervention will be required. The advantage of this method is obviously the usability. Consequently, the overall security of the system may be threatened in case of reverse-engineering of the camera firmware, or simply if a user would be able to understand which is the “secret” used for the camera authentication. In such a case, an attacker could impersonate a legitimate camera and try to compromise the whole architecture. It is important to note that, even in such less secure scenario, usually the cameras are positioned in a place that is not so easily accessible and sometimes cameras are, in turn, monitored by each other camera(s), so a tentative of reaching a camera for manipulation can be easily discovered and prosecuted.

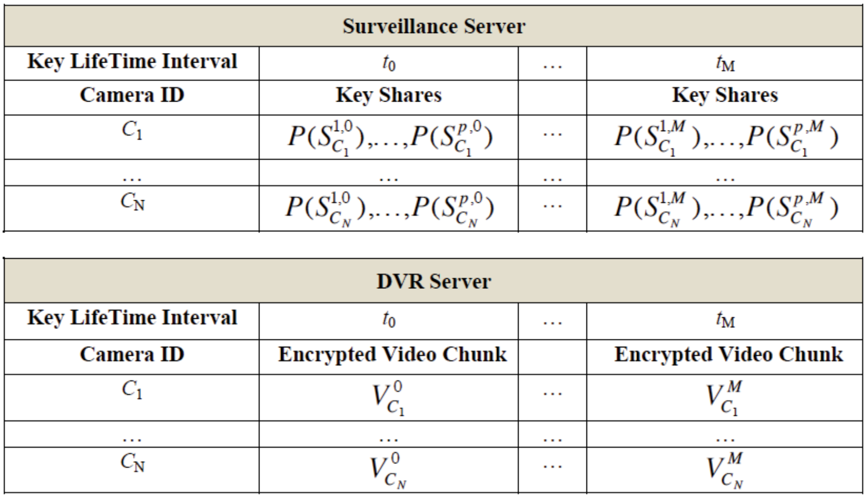
The second approach, the one using a secret information on-board the camera, can be considered nearly secure as the first one if the cameras make use of some Trusted Computing technology. In such a case, the secret information is protected by the secure architecture provided by the trusted device. For the implementation of the prototype, the authors have chosen to consider the simplest case, i.e. the case in which the secret information is stored on the camera. In particular, such secret information corresponds to the serial number of the device. Such a simplification has been done only for developing reason and can be easily replaced by a more secure means, like a smartcard or a Trusted Computing device. This generalization does not undermine the overall security of the architecture.

After the initial setup phase, the surveillance system is ready to be used. Several keys thus control all its cryptographic operations:

1. the master key K_{C_i} which is randomly and independently generated by each camera C_i (clearly they are different from camera to camera);
2. the p shares $S^1_{C_i}, \cdots, S^p_{C_i}$ which are generated by using the secret sharing scheme for each K_{C_i} key;
3. the public/private key pairs of each involved authority;
4. the "protected" encryption keys $P(S_{C_i})$ which are encrypted using the public key of the authorities prior of sending them to the Surveillance Server and that are used for the real-time encryption.

For the public/private key at point 3, the authors assume that each authority already owns such keys. Due to the use of standards X.509 certificates, the system allows an authority to store such keys wherever their security policies require (for example on a PIN-protected smartcard). During the normal day-by-day operations, each camera will encrypt the video using the master key K_{C_i} which is changed on a time basis (for example every 5 minutes) and destroyed after its expiration. The shares $S^1_{C_i}, \cdots, S^p_{C_i}$, associated with the p involved authorities, are then encrypted with the public key of the corresponding authority and sent, through a secure TLS/SSL session, to the Surveillance Server where they will be stored on specific files to be accessed from the requiring authorities afterwards.

The whole data structures used in the presented architecture, together with their localisation on the specific system components, are shown in Figure 2. For



Surveillance Server			
Key LifeTime Interval	t_0	…	t_M
Camera ID	**Key Shares**		**Key Shares**
C_1	$P(S_{C_1}^{1,0}),\ldots,P(S_{C_1}^{p,0})$	…	$P(S_{C_1}^{1,M}),\ldots,P(S_{C_1}^{p,M})$
…	…	…	…
C_N	$P(S_{C_N}^{1,0}),\ldots,P(S_{C_N}^{p,0})$	…	$P(S_{C_N}^{1,M}),\ldots,P(S_{C_N}^{p,M})$

DVR Server			
Key LifeTime Interval	t_0	…	t_M
Camera ID	**Encrypted Video Chunk**		**Encrypted Video Chunk**
C_1	$V_{C_1}^0$	…	$V_{C_1}^M$
…	…	…	…
C_N	$V_{C_N}^0$	…	$V_{C_N}^M$

Fig. 2. Data allocation taxonomy

each camera C_i $(1 \leq i \leq N)$ in a specific time interval t_j $(0 \leq j \leq M)$ there are p shares $S_{C_i}^{1,j}, \cdots, S_{C_i}^{p,j}$ $(1 \leq k \leq p)$ associated to the cyphering secret K_{C_i} stored on the Surveillance Server in encrypted format. Analogously, for each camera C_i and time interval t_j, there will be a corresponding video chunk (stored on the DVR Server) $V_{C_i}^{j}$ encrypted with the secret K_{C_i}.

4 Implementation Details

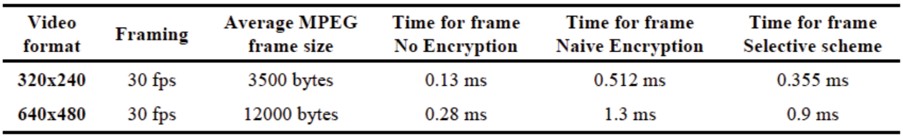
A simple proof-of-concept video surveillance system has been developed to test the effectiveness of the proposed security framework, with an emphasis on the use of currently available commercial off-the-shelf (COTS) devices and open source components, in order to avoid licensing costs and legalities. The use of COTS components for building the prototype hardware platform let the system to be easily implemented on an industrial production scale, allowing it to increase in performance and decrease in cost together with the evolution of the involved COTS technologies. The system, implemented entirely in ANSI C on standard Linux-based devices (with kernel 2.6.19), supports USB and wireless/wireline IP cameras and implements the scheme in which the cameras perform the initial setup in an automatic way by using the serial number of the camera. Heterogeneous clients can access the DVR Server in order to decrypt the recorded videos. For instance, policemen or security officers, with the correct combination of shares required by the implemented security policy, can interface with the DVR Server by using laptops or PDAs equipped with smartcard readers, and examine the video material captured by one or more cameras. The prototype implementation makes use of the Video for Linux (V4L2) API [10] [11] to interface the camera devices used for video captures. Such a library is in charge with the management of the various video devices that a Linux box can handle, leaving the programmers free to focus on non-hardware specific programming tasks. For the GUI, the authors adopted the GTK+ library [12], a highly usable toolkit that provides the developers (using various programming languages such as C/C++, Python, etc.) with lot of features very useful when working with GUIs.

The DALÌ library [13], developed at the Cornell University, which handles the structural elements of the MPEG compression, has been used for MPEG video manipulation, and precisely for the creation of the MPEG-1 file starting from a series of images in YUV format. The OpenSSL [14] and SSSS (Shamir Secret Sharing Scheme) [15] libraries have been used for all the encryption and key generation tasks. OpenSSL has been adopted even for the symmetric algorithms (DES and AES) which are in charge of securing the real-time video data. The prototype application requires that the operators download their encrypted key shares from the Surveillance Server and decrypt them by using the associated private keys. After the decryption of the needed shares, the master key is recovered and the involved operator is able to process the encrypted video data.

5 Coding Performance

One of the fundamental parameter describing the performance of the proposed encryption framework is coding efficiency. Indeed, it is important that coding performance is not excessively affected by the encryption activity. For this purpose, the case in which no encryption is applied (i.e., corresponding to the original MPEG-1 encoding), together with the cases where encryption is performed using the naive AES approach (all the packets in the stream are encrypted with a 128 bits key) and the proposed selective encryption scheme, have been analysed comparing their average frame encoding time.

Table 1. Coding efficiency results



Video format	Framing	Average MPEG frame size	Time for frame No Encryption	Time for frame Naive Encryption	Time for frame Selective scheme
320x240	30 fps	3500 bytes	0.13 ms	0.512 ms	0.355 ms
640x480	30 fps	12000 bytes	0.28 ms	1.3 ms	0.9 ms

In detail, for a 10 minutes video (both 320x240 and 640x480 at 30 fps) with no moving objects processed on an Intel Centrino 1.6GHz Dual-Core CPU, the proposed approach, differently from the naive one, presents a minimum impact of coding efficiency, as shown in Table 1.

6 Security Analysis

This section aims at giving arguments to substantiate the security of the proposed framework. All the building blocks of the framework communicate through TLS/SSL secure sessions guaranteeing strong authentication between the communicating parties. This assures that all the communications share a common and consolidated level of security guaranteed by the state-of-the-art public-key encryption technologies implemented in TLS/SSL. The shares are created inside each camera and encrypted before the transmission so that they never leave the camera in clear-text format. The Surveillance Server, upon receiving the shares, stores them in encrypted format. When a decryption request arrives, the secret key is reconstructed by decrypting a set of shares with cardinality equal to the value of threshold k associated with the used secret scheme. All the above operations are performed in memory without storing any information on semi-permanent storage space. Hence, the only Achilles heel is the short permanence in memory of such data. All the key-management operations are performed by using well-known cryptographic libraries relying on a good pseudo-random number generator. Appropriately chosen key lengths (AES 128 bits for symmetric operations and RSA 1024 bits for asymmetric ones) ensure that exhaustive searching in the key space or factoring will be infeasible, protecting all the cryptographic operations from brute-force attacks.

Each master key, used for encrypting video material, is generated randomly and independently by the cameras which may be assumed to be trusted entities if all their activities are implemented by using trust enforcement technologies such as smartcards or Trusted Computing (TPM) hardware. At the same time, even the involved processing facilities (DVR Server, Surveillance Server) should be secured by using analogous technologies and appropriate security hardening policies. Since the encryption keys are periodically refreshed during the system activity, it is obvious that by choosing a sufficiently short key lifetime, the proposed scheme can be considered enough secure against *known-plaintext* attacks.

Furthermore, the adopted video encryption scheme, based on AES and one-time pad, is not affected by *replacement* attacks [16]. Whereas AES is known to be vulnerable to some *side-channel* ciphertext-only attacks such as the *timing attack* [17] [18], which may occur in any implementation that does not run in fixed time, there are some workarounds (introducing delays to the faster operations or avoiding the use of S-boxes) which make AES a bit slower but ensure that it remains provably secure to *ciphertext-only* attacks.

Finally, the use of Shamir secret sharing scheme protect the framework from coalition attacks with less then k cooperating compromised parties.

7 Error Tolerance

Secret sharing is the fundamental factor helping to provide error tolerance. In the scheme, k shares of the secret (used for the stream encryption) will be distributed among p (with $p \geq k$) available ones. Due to the flexibility of the (k, p) threshold scheme, the proposed framework can support both the addition of new agencies that can participate to the process of reconstructing video material, and manage key-shares losses or theft, since lose or change of one key can be tolerated. However, such advanced features introduce an additional, but tolerable, computational overhead. If more than k keys are distributed, when a key share is lost, the remaining k keys can be used to reconstruct the original video content according to the Shamir secret sharing scheme. If an encrypted content gets modified the PCBC scheme (and eventually the HMAC) will ensure the detection of an error or an integrity violation.

8 Conclusions

The presented architecture demonstrates to be an efficient, flexible and standard-based solution for a really secure video surveillance system, supporting the privacy and reliability needs of modern mission-critical business-oriented applications. A media-aware hybrid encryption scheme has been implemented to efficiently enforce lawful evidence and confidentiality of the video streams in an open and insecure networked environment.

This should be very useful in easing the spreading and deployment of video surveillance infrastructures because it overcomes most of the privacy concerns which often adversely affect the retention of sensible data. The authors believe that such cross-domain application could bring new light on existing problems, or reveal new issues due to the interaction of different technologies adopted within interdisciplinary fields.

References

1. Stinson, D.R.: Cryptography: Theory and Practice, 1st edn. Chapman & Hall, New York (1995)
2. Tang, L.: Methods for encrypting and decrypting MPEG video data efficiently. In: Proceedings of the 4th ACM International Conference on Multimedia, Boston, Mass, USA, November 1996, pp. 219–229 (1996)
3. Agi, I., Long, L.: An Empirical Study of Secure MPEG Video Transmissions. In: Proceedings of the Internet Society Symposium on Network and Distributed System Security, San Diego, CA, February 1996, pp. 137–144 (1996)
4. Qiao, L., Nahrstedt, K.: Comparison of MPEG Encryption Algorithms. Computers and Graphics 22(4), 437–448 (1998)
5. Shi, C., Bhargava, B.K.: A Fast MPEG Video Encryption Algorithm. In: Proceedings of the 6th ACM International Conference on Multimedia, Bristol, England, September 1998, pp. 81–88 (1998)
6. Shamir, A.: How to Share a Secret. Communications of the ACM 22, 612–613 (1979)
7. IETF NTP WG: The Network Time Protocol (March 2010), http://datatracker.ietf.org/wg/ntp/
8. Qiao, L., Nahrstedt, K.: A New Algorithm for MPEG Video Encryption. In: Proceedings of The First International Conference on Imaging Science, Systems, and Technology (CISST'97), Las Vegas, Nevada, July 1997, pp. 21–29 (1997)
9. Menezes, J., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press, Boca Raton (1997)
10. Schimek, M.H.: Video for Linux Two API Specification (2008), http://www.linuxtv.org/downloads/video4linux/API/V4L2_API/
11. de Goede, H.: Video4Linux v4l-utils and libv4l (March 2010), http://freshmeat.net/projects/libv4l/
12. The GTK+ Team, The GTK+ Project (September 2009), http://www.gtk.org/
13. Dalì, A.: Multimedia Software Library (June 1999), http://www.cs.cornell.edu/dali/
14. The OpenSSL Core and Development Team: The OpenSSL Project (March 2010), http://www.openssl.org/
15. Poettering, B.: Shamir's Secret Sharing Scheme (September 2006), http://point-at-infinity.org/ssss/
16. Podesser, M., Schmidt, H., Uhl, A.: Selective Bitplane Encryption for Secure Transmission of Image Data in Mobile Environments. In: CD-ROM Proceedings of the 5th IEEE Nordic Signal Processing Symposium (NORSIG 2002), Tromso-Trondheim, Norway (2002)
17. Bernstein, D.J.: Cache-Timing Attacks on AES (2005), http://cr.yp.to/antiforgery/cachetiming-20050414.pdf
18. Bonneau, J., Mironov, I.: Cache-Collision Timing Attacks Against AES (2005), http://research.microsoft.com/users/mironov/papers/aestiming.pdf

Reputation as Aggregated Opinions

John Debenham[1] and Carles Sierra[2]

[1] QCIS, University of Technology, Sydney, Australia
debenham@it.uts.edu.au
[2] Institut d'Investigació en Intel·ligència Artificial - IIIA,
Spanish Scientific Research Council, CSIC
08193 Bellaterra, Catalonia, Spain
sierra@iiia.csic.es

Abstract. A model of reputation is presented in which agents share and aggregate their opinions, and observe the way in which their opinions effect the opinions of others. A method is proposed that supports the deliberative process of combining opinions into a group's reputation. The reliability of agents as opinion givers are measured in terms of the extent to which their opinions differ from that of the group reputation. These reliability measures are used to form an *a priori* reputation estimate given the individual opinions of a set of independent agents.

1 Introduction

This paper describes a reputation model that is inspired by information theory and that is based on the *information-based agency* explained elsewhere [1]. *Reputation* is the opinion (more technically, a social evaluation) of a group about something [2]. So a group's reputation about a thing will be related in some way to the opinions that the individual group members hold towards that thing [3]. An opinion is an assessment, judgement or evaluation of something, and are represented in this paper as probability distributions on a suitable ontology called the *evaluation space* E.

An opinion is an evaluation of an *aspect* of a thing [4]. An aspect is the "point of view" that an agent has when forming his opinion. An opinion is evaluated in context. The *context* is the set of all things that the thing is being, explicitly or implicitly, evaluated with or against. The set of valuations of all things in the context calibrates the valuation space [5]. For example, "this is the best paper in the conference". The context can be vague: "of all the presents you could have given me, this is the best". If agents are to discuss opinions then they must have some understanding of each other's context.

Summarising the above, an *opinion* is an agent's evaluation of a particular aspect of a thing in context [6]. A representation of an opinion will contain: the thing, its aspect, its context, and a distribution on E representing the evaluation of the thing.

In this paper we explore the case of opinions being formed through a social evaluation process. Each agent in a group of agents first forms an individual

F. Buccafurri and G. Semeraro (Eds.): EC-Web 2010, LNBIP 61, pp. 85–96, 2010.

opinion on some thing. Second these individual opinions are shared with rest of the group. A group discussion follows as a result of which each agent states a revised opinion. Following that there is another discussion during which the group attempts to formulate a shared reputation for the thing. The model that we describe is based on three observations only for each participating agent: their initial individual opinion, their revised opinion, and the group's reputation if one is agreed upon. This social evaluation process was suggested by a process used to evaluate submissions to conferences.

2 The Multiagent System

We assume that a multiagent system $\{\alpha, \beta_1, \ldots, \beta_o, \xi, \theta_1, \ldots, \theta_t\}$, contains an agent α that interacts with negotiating agents, β_i, information providing agents, θ_j, and an *institutional agent*, ξ, that represents the institution where we assume the interactions happen [7]. Institutions give a normative context to interactions that simplify matters (e.g an agent can't make an offer, have it accepted, and then renege on it). The institutional agent ξ may form opinions on the actors and activities in the institution and may publish reputation estimates on behalf of the institution. The agent ξ also fulfils a vital role to compensate for any lack of sensory ability in the other agents by promptly and accurately reporting observations as events occur; for an example, without such reporting an agent may have no way of knowing whether it is a fine day or not. When we consider the system from the point of view of a particular agent we will use agent α, and that is α's only significance.

Our agents are information-based [8], everything in their world is uncertain. To deal with this uncertainty, the world model, $\mathcal{M}^t$, consists of random variables each representing a point of interest in the world. Distributions are then derived for these variables on the basis of information received. Additionally, information-based agents [8] are endowed with machinery for valuing the information that they have, and that they receive. They were inspired by the observation that "everything an agent says gives away information". They model how much they know about other agents, and how much they believe other agents know about them. By classifying private information into functional classes, and by drawing on the structure of the ontology, they develop a map of the 'intimacy' [9] of their relationships with other agents.

3 Forming Opinions

This section describes how an information-based agent forms opinions [10]. Section 4 will describe how the opinions of the agents in a group may be distilled into a reputation.

An opinion is a valuation by an agent of an aspect of a thing taken in context. Formally, $O_i(z, a, C)$ represents the result of the valuation by agent β_i of aspect a of thing z in context C. For example, the valuation by agent "Carles" of the "scientific quality" aspect of the thing "John's paper" in the context of "the

AAMAS conference submissions". The context C is often subjectively chosen by the agent, and is not part of the opinion($\cdot$) primitive, although context may be the subject of associated argumentation.

As noted above, to preserve consistency and generality we assume that all opinions are expressed as probability distributions over some suitable E. If an agent expresses an opinion as $\mathbb{P}(X = e_i)$ we treat this as the distribution with minimum relative entropy with respect to the prior subject to the constraint $\mathbb{P}(X = e_i)$ — in case there is no known prior we use the maximum entropy, uniform distribution. For example, if $E = (\text{fine}, \text{cloudy}, \text{wet}, \text{storm})$ then the opinion "I am 70% certain that tomorrow will be fine" will be represented as $(0.7, 0.1, 0.1, 0.1)$ for a uniform prior.

The distributions in an agent's world model $\mathcal{M}^t$ represent the agent's opinions about the value of the corresponding random variable over some valuation space. Opinions may be derived from opinions. For example, to form an opinion on "tomorrow's suitability for a picnic" and agent may introduce random variables for: tomorrow's mid-day temperature, tomorrow's mid-day cloud cover, and tomorrow's mid-day wind strength, construct distributions for them using on-the-fly weather forecast information, and then derive an opinion about the picnic somehow from these three distributions.

In Section 3.1 we describe how the distributions in the world model are updated as real-time information becomes available; in that section we also estimate the reliability of each information source by subsequently validating the information received from it.

3.1 Updating Opinions with Real-Time Information

In the absence of in-coming messages the distributions in $\mathcal{M}^t$ should gradually decay towards some zero-information state. In many cases there is background knowledge about the world — for example, a distribution of the daily maximum temperature in Barcelona in May — such a distribution is called a *decay-limit distribution*. If the background knowledge is incomplete then one possibility is to assume that the decay limit distribution has maximum entropy whilst being consistent with the available data. Given a distribution, $\mathbb{P}(X_i)$, and a decay limit distribution $\mathbb{D}(X_i)$, $\mathbb{P}(X_i)$ decays by:

$$\mathbb{P}^{t+1}(X_i) = \varDelta_i(\mathbb{D}(X_i), \mathbb{P}^t(X_i)) \tag{1}$$

where $\varDelta_i$ is the *decay function* for the X_i with $\lim_{t\to\infty} \mathbb{P}^t(X_i) = \mathbb{D}(X_i)$. For example, $\varDelta_i$ could be linear: $\mathbb{P}^{t+1}(X_i) = (1 - \nu_i) \times \mathbb{D}(X_i) + \nu_i \times \mathbb{P}^t(X_i)$, where $\nu_i < 1$ is the decay rate for the i'th distribution. Either the decay function or the decay limit distribution could also be a function of time: $\varDelta_i^t$ and $\mathbb{D}^t(X_i)$.

The following procedure updates $\mathcal{M}^t$. Suppose that α receives a message μ from agent β at time t.[1] Suppose that this message states that something is so

[1] This message is not necessarily a message from the language in section 2. We refer with μ to any *inform* message with propositional content that can be processed by the agent.

with probability v, and suppose that α attaches an epistemic belief $\mathbb{R}^t(\alpha, \beta, \mu)$ to μ — this probability reflects α's level of personal *caution*. Each of α's active plans, s, contains constructors for a set of distributions $\{X_i\} \in \mathcal{M}^t$ together with associated *update functions*, $J_s(\cdot)$, such that $J_s^{X_i}(\mu)$ is a set of linear constraints on the posterior distribution for X_i. Denote the prior distribution $\mathbb{P}^t(X_i)$ by $\boldsymbol{p}$, and let $\boldsymbol{p}_{(\mu)}$ be the distribution with minimum relative entropy[2] with respect to $\boldsymbol{p}$: $\boldsymbol{p}_{(\mu)} = \arg\min_{\boldsymbol{r}} \sum_j r_j \log \frac{r_j}{p_j}$ that satisfies the constraints $J_s^{X_i}(\mu)$. Then let $\boldsymbol{q}_{(\mu)}$ be the distribution:

$$\boldsymbol{q}_{(\mu)} = \mathbb{R}^t(\alpha, \beta, \mu) \times \boldsymbol{p}_{(\mu)} + (1 - \mathbb{R}^t(\alpha, \beta, \mu)) \times \boldsymbol{p} \tag{2}$$

and then let:

$$\mathbb{P}^t(X_{i(\mu)}) = \begin{cases} \boldsymbol{q}_{(\mu)} & \text{if } \boldsymbol{q}_{(\mu)} \text{ is more interesting than } \boldsymbol{p} \\ \boldsymbol{p} & \text{otherwise} \end{cases} \tag{3}$$

A general measure of whether $\boldsymbol{q}_{(\mu)}$ is more interesting than $\boldsymbol{p}$ is: $\mathbb{K}(\boldsymbol{q}_{(\mu)} \| \mathbb{D}(X_i)) > \mathbb{K}(\boldsymbol{p} \| \mathbb{D}(X_i))$, where $\mathbb{K}(\boldsymbol{x} \| \boldsymbol{y}) = \sum_j x_j \ln \frac{x_j}{y_j}$ is the Kullback-Leibler divergence between two probability distributions $\boldsymbol{x}$ and $\boldsymbol{y}$.

Finally merging Eqn. 3 and Eqn. 1 we obtain the method for updating a distribution X_i on receipt of a message μ:

$$\mathbb{P}^{t+1}(X_i) = \varDelta_i(\mathbb{D}(X_i), \mathbb{P}^t(X_{i(\mu)})) \tag{4}$$

This procedure deals with integrity decay, and with two probabilities: first, the probability v in the message μ, and second the belief $\mathbb{R}^t(\alpha, \beta, \mu)$ that α attached to μ.

Reliability of the Information Source. An empirical estimate of $\mathbb{R}^t(\alpha, \beta, \mu)$ may be obtained by measuring the 'difference' between commitment and verification [13]. Suppose that μ is received from agent β at time u and is verified by ξ as μ' at some later time t.[3] Denote the prior $\mathbb{P}^u(X_i)$ by $\boldsymbol{p}$. Let $\boldsymbol{p}_{(\mu)}$ be the posterior minimum relative entropy distribution subject to the constraints $J_s^{X_i}(\mu)$, and let $\boldsymbol{p}_{(\mu')}$ be that distribution subject to $J_s^{X_i}(\mu')$. We now estimate what $\mathbb{R}^u(\alpha, \beta, \mu)$ should have been in the light of knowing *now*, at time t, that μ should have been μ'.

The idea of Eqn. 2, is that $\mathbb{R}^t(\alpha, \beta, \mu)$ should be such that, *on average* across $\mathcal{M}^t$, $\boldsymbol{q}_{(\mu)}$ will predict $\boldsymbol{p}_{(\mu')}$ — no matter whether or not μ was used to update

[2] Given a probability distribution $\boldsymbol{q}$, the *minimum relative entropy distribution* $\boldsymbol{p} = (p_1, \ldots, p_I)$ subject to a set of J linear constraints $\boldsymbol{g} = \{g_j(\boldsymbol{p}) = \boldsymbol{a_j} \cdot \boldsymbol{p} - c_j = 0\}, j = 1, \ldots, J$ (that must include the constraint $\sum_i p_i - 1 = 0$) is: $\boldsymbol{p} = \arg\min_{\boldsymbol{r}} \sum_j r_j \log \frac{r_j}{q_j}$. This may be calculated by introducing Lagrange multipliers $\boldsymbol{\lambda}$: $L(\boldsymbol{p}, \boldsymbol{\lambda}) = \sum_j p_j \log \frac{p_j}{q_j} + \boldsymbol{\lambda} \cdot \boldsymbol{g}$. Minimising L, $\{\frac{\partial L}{\partial \lambda_j} = g_j(\boldsymbol{p}) = 0\}, j = 1, \ldots, J$ is the set of given constraints $\boldsymbol{g}$, and a solution to $\frac{\partial L}{\partial p_i} = 0, i = 1, \ldots, I$ leads eventually to $\boldsymbol{p}$. Entropy-based inference is a form of Bayesian inference that is convenient when the data is sparse [11] and encapsulates common-sense reasoning [12].

[3] This could be later communicated as $\text{inform}(\gamma, \alpha, \text{experience}(\gamma, \beta, \mu, \mu'), t)$.

the distribution for X_i, as determined by the condition in Eqn. 3 at time u. The *observed reliability* for μ and distribution X_i, $\mathbb{R}^t_{X_i}(\alpha,\beta,\mu)|\mu'$, on the basis of the verification of μ with μ', is the value of k that minimises the Kullback-Leibler divergence:

$$\mathbb{R}^t_{X_i}(\alpha,\beta,\mu)|\mu' = \arg\min_k \mathbb{K}(k \cdot \boldsymbol{p}_{(\mu)} + (1-k)\cdot \boldsymbol{p} \parallel \boldsymbol{p}_{(\mu')})$$

The predicted *information* in the enactment of μ with respect to X_i is:

$$\mathbb{I}^t_{X_i}(\alpha,\beta,\mu) = \mathbb{H}^t(X_i) - \mathbb{H}^t(X_{i(\mu)}) \tag{5}$$

that is the reduction in uncertainty in X_i where $\mathbb{H}(\cdot)$ is Shannon entropy. Eqn. 5 takes account of the value of $\mathbb{R}^t(\alpha,\beta,\mu)$.

If $\mathbf{X}(\mu)$ is the set of distributions that μ affects, then the *observed reliability* of β on the basis of the verification of μ with μ' is:

$$\mathbb{R}^t(\alpha,\beta,\mu)|\mu' = \frac{1}{|\mathbf{X}(\mu)|}\sum_i \mathbb{R}^t_{X_i}(\alpha,\beta,\mu)|\mu' \tag{6}$$

If $\mathbf{X}(\mu)$ are independent the predicted *information* in μ is:

$$\mathbb{I}^t(\alpha,\beta,\mu) = \sum_{X_i \in \mathbf{X}(\mu)} \mathbb{I}^t_{X_i}(\alpha,\beta,\mu) \tag{7}$$

Suppose α sends message μ to β where μ is α's private information, then assuming that β's reasoning apparatus mirrors α's, α can estimate $\mathbb{I}^t(\beta,\alpha,\mu)$.

For each formula φ at time t when μ has been verified with μ', the *observed reliability* that α has for agent β in φ is:

$$\mathbb{R}^{t+1}(\alpha,\beta,\varphi) = (1-\nu)\times\mathbb{R}^t(\alpha,\beta,\varphi) + \nu\times\mathbb{R}^t(\alpha,\beta,\mu)|\mu' \times \text{Sim}(\varphi,\mu)$$

where Sim measures the semantic distance between two sections of the ontology, and ν is the learning rate. Over time, α notes the context of the various μ received from β, and over the various contexts calculates the relative frequency, $\mathbb{P}^t(\mu)$. This leads to an overall expectation of the *reliability* that agent α has for agent β:

$$\mathbb{R}^t(\alpha,\beta) = \sum_\mu \mathbb{P}^t(\mu)\times\mathbb{R}^t(\alpha,\beta,\mu)$$

3.2 Verifiable Opinions

An opinion is *verifiable* if within a "reasonable amount of time" it ceases to be an opinion and becomes an observable fact; for example, the opinion "tomorrow's maximum temperature will be over 30°" is verifiable, whereas the opinion "the Earth will exist in 100,000 years time" is not verifiable in any practical sense, and "Brahms' symphonies are ghastly" will never be verifiable.

The articulation by β of a verifiable opinion carries with it the intrinsic commitment that it will in due time become an observable true fact. α will be interested in any variation between β's commitment, φ, and what is actually observed (as advised by the institution agent ξ), as the fact, φ'. We denote the relationship between opinion and fact, $\mathbb{P}^t(\text{Observe}(\varphi')|\text{Commit}(\varphi))$ simply as $\mathbb{P}^t(\varphi'|\varphi) \in \mathcal{M}^t$.

In the absence of in-coming messages the conditional probabilities, $\mathbb{P}^t(\varphi'|\varphi)$, should tend to ignorance as represented by the *decay limit distribution* and Eqn. 1. We now show how Eqn. 4 may be used to revise $\mathbb{P}^t(\varphi'|\varphi)$ as observations are made. Let the set of possible factual outcomes be $\Phi = \{\varphi_1, \varphi_2, \ldots, \varphi_m\}$ with prior distribution $\boldsymbol{p} = \mathbb{P}^t(\varphi'|\varphi)$. Suppose that message μ is received from ξ that verifies or refutes a previously stated verifiable opinion expressed by β, we estimate the posterior $\boldsymbol{p}_{(\mu)} = (p_{(\mu)i})_{i=1}^m = \mathbb{P}^{t+1}(\varphi'|\varphi)$.

First, if $\mu = (\varphi_k, \varphi)$ is observed then α may use this observation to estimate $p_{(\varphi_k)k}$ as some value d at time $t+1$. We estimate the distribution $\boldsymbol{p}_{(\varphi_k)}$ by applying the principle of minimum relative entropy as in Eqn. 4 with prior $\boldsymbol{p}$, and the posterior $\boldsymbol{p}_{(\varphi_k)} = (p_{(\varphi_k)j})_{j=1}^m$ satisfying the single constraint: $J^{(\varphi'|\varphi)}(\varphi_k) = \{p_{(\varphi_k)k} = d\}$.

Second, we consider the effect that the verification ϕ' of another simple, verifiable opinion ϕ of β has on $\boldsymbol{p}$. This is achieved by appealing to the structure of the ontology using a semantic distance function $\text{Sim}(\cdot)$. Given the observation $\mu = (\phi', \phi)$, define the vector $\boldsymbol{t}$ by:

$$t_i = \mathbb{P}^t(\varphi_i|\varphi) + (1- \mid \text{Sim}(\phi', \phi) - \text{Sim}(\varphi_i, \varphi) \mid) \cdot \text{Sim}(\varphi', \phi)$$

for $i = 1, \ldots, m$. $\boldsymbol{t}$ is not a probability distribution. The multiplying factor $\text{Sim}(\varphi', \phi)$ limits the variation of probability to those formulae whose ontological context is not too far away from the observation. The posterior $\boldsymbol{p}_{(\phi', \phi)}$ is defined to be the normalisation of $\boldsymbol{t}$.

In this section we have shown how an information-based agent models the accuracy of an agent's opinions when they are verifiable. The model produced is predictive in the sense that when an opinion is stated it gives a distribution of expectation over the space of factual outcomes.

3.3 Unverifiable Opinions

If an opinion can not be verified then one way in which it may be evaluated is to compare it with the corresponding individual opinions, or group reputation, of a group of agents. The focus of this paper is on reputation; that is, a social evaluation conducted by a group. We deal with unverifiable opinions using a social evaluation framework that is abstracted from any particular case. The idea is that a group G of n agents independently form a prior opinion, O_i on the same thing. Each agent has a prior confidence value, c_i, that estimates how close its prior opinion, O_i, is expected to be to the reputation, or common opinion, of the group, R_G — precisely c_i measures how effective the agent is at influencing the opinions of other agents, it does *not* measure how good its opinion is in any absolute sense as the opinion is assumed to be unverifiable. The agents then make their prior opinions public to the other agents and an argumentative

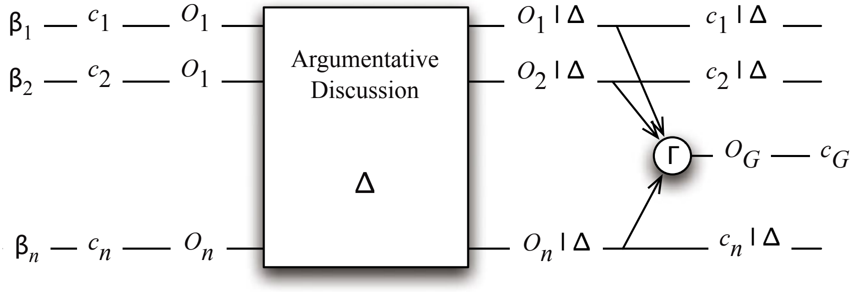


Fig. 1. The social evaluation framework in which a group G of n agents β_1,...,β_n table their private opinions $O_1, \ldots, O_n$, have an open, argumentative discussion Δ (see Section 3.3), and then revise their opinions $O_1|\Delta, \ldots, O_n|\Delta$. This is followed by another argumentative discussion Γ (see Section 4) during which the agents consider whether revised opinions can be distilled into a common reputation R_G. The symbols c_i and c_G are confidence values as explained below.

discussion, Δ, takes place during which the agents may choose to revise their opinions, $O_i|\Delta$. When the revised opinions are published a second argumentative discussion, Γ, takes place during which the agents attempt to distil their opinions into a group reputation, R_G. The confidence estimates, c_i are then revised by noting the differences between O_i, $O_i|\Delta$ and R_G, to give posterior values, $c_i|\Delta$. The processes in Figure 1 are summarised as:

$$\Delta : f(\{(O_i, c_i\}) = \{O_i|\Delta\}$$

$$\Gamma : g(\{(O_i|\Delta, c_i\}) = (R_G, d_G)$$

$$\{\Delta, \Gamma\} : h(\{(O_i, c_i, O_i|\Delta\}, R_G) = \{c_i|\Delta\}$$

The function $f(\cdot)$ is the product of the discussion Δ — we simply observe the outcome. Function $g(\cdot)$ is described in Section 4, and $h(\cdot)$ in Section 5.

4 Combining Opinions and Forming Reputation

A reputation is a social evaluation by a group. When the group is a set of autonomous agents the only sense in which an opinion can exist is as a common opinion throughout the group. The objective of the argumentative process Γ in Figure 1 is to determine a common view if one exists. The following procedure first determines whether a common view exists, and second it offers three views of what that common view could be. The three different views vary with differing degrees of statistical dependence between the agents.

The process of distilling opinions into a reputation can not simply be computed. For example, consider two agents who are reviewing the same conference paper and are in total agreement about the result "a 'strong accept' with confidence 0.8" where the reliability of each agent is 90%. What should their combined opinion, or in this case 'paper reputation', be? As their individual reliability is 90% perhaps the common view is "a 'strong accept' with confidence 0.72". Alternatively because they both agree, and may have quite different reasons supporting their views, perhaps the common view should be "a 'strong accept' with some confidence greater than 0.8".

The work described in the remainder of this section and in Section 5 is expressed in terms of two agents; it extends naturally to n agents. The procedure is based on three methods that are detailed below.

Dependent Method. To form a *combined opinion* of two opinions, X_1 and X_2, construct the joint distribution $W = (X_1, X_2, Z)$ and impose the constraints:

$$\left(\sum_i \mathbb{P}(W = w_i) \mid X_k = x_j\right) = \mathbb{P}(X_k = x_j),\ k = 1, 2$$

$$\left(\sum_i \mathbb{P}(W = w_i) \mid X_k = Z\right) = c_k,\ k = 1, 2$$

let W be the distribution of maximum entropy that satisfies these constraints. Then the combined opinion $\text{Dep}(X_1, X_2)$ is $\mathbb{P}(Z = z)$. If the data is inconsistent then the value is undefined — this is a test of whether the data is consistent. If the data is inconsistent then this indicates that there is no shared opinion. Being based on a maximum entropy calculation the posterior is a conservative combination of the given opinions — it is "maximally noncommittal" to that which is not known. To calculate this dependent, combined opinion when the prior is known, calculate the minimum relative entropy distribution with respect to that prior using the same constraints as described.

Υ Method. Let's define $\mathbb{P}(\alpha, d)$ as the probability that an opinion O_α expressed by α (i.e, a probability distribution) is at distance d of the true distribution (or at distance d of a group opinion). That is, the probability that a certain distribution Q is the right one is defined as $P(Q \text{ is right}) = \mathbb{P}(\alpha, DIST(O_\alpha, Q))$ for an appropriate distance measure $DIST$.[4] These distributions can be obtained by datamining past group opinion formation processes.

Given a group G, we look for the group opinion, R_G such that the certainty on that group opinion being the right one is maximised. That is,

$$R_G = \max_Q \Upsilon(\{\mathbb{P}(\alpha, DIST(O_\alpha, Q))\}_{\alpha \in G})$$

Where Υ is the uninorm operator [15]. In case there are several such group opinions we prefer the one with maximum entropy. And then,

$$d_G = \Upsilon(\{\mathbb{P}(\alpha, DIST(O_\alpha, R_G))\}_{\alpha \in G})$$

For the values in Table 1, we discreetise the $\mathbb{P}(\alpha, d)$ in the intervals between the points in the following list: $[0, 0.035, 0.3, 0.5, 0.8, 1]$.

[4] Kullback-Leibler divergence, or the earth movers distance [14] could be used.

Independent Method. Given a prior distribution $\mathbb{P}(W = x_j)$, a pair of opinions, $\mathbb{P}(X_i = x_j)$ $i = 1, 2$, with their respective certainties c_i, assuming that the agents' opinions are statistically independent, let $w_{i,j} = c_i \times \mathbb{P}(X_i = x_j)$, $i = 1, 2$, and let $v_j = \frac{\prod_i w_{i,j}}{\prod_i w_{i,j} + \prod_i (1 - w_{i,j})}$ then the combined opinion $\mathrm{Ind}(X_1, X_2)$ is: $v_j + (1 - \sum_k v_k) \times \mathbb{P}(W = x_j)$, with strength $\sum_k v_k$. This method assumes that the priors are independent (unlikely in practice) and has the property that the probabilities in two similar distributions are amplified.

The overall procedure plays the role of a mediator [16]. If the 'Dependent Method' does not return a value then the data is inconsistent, and the agents should either have further discussion or "agree to disagree". Otherwise calculate the three values $\mathrm{Dep}(\cdot)$, $\varUpsilon(\cdot)$ and $\mathrm{Ind}(\cdot)$. Propose $\varUpsilon(\cdot)$ to the agents, and if they accept it then that is their common opinion. Otherwise propose that their common opinion lies somewhere between $\mathrm{Dep}(\cdot)$ and $\mathrm{Ind}(\cdot)$ and leave it to them to determine it.

Table 1 contains some sample values for the three methods. In Case 3 the two opinions are identical with maximal value of 0.8 and strengths of 0.8 and 0.9. The $\mathrm{Dep}(X_1, X_2)$ method is conservative and gives 0.77 because of the strength values. The $\varUpsilon(X_1, X_2)$ method balances the strength uncertainty with the fact that their are two shared views to give 0.8. The $\mathrm{Ind}(X_1, X_2)$ method is bold and gives 0.85 because two agents share the same view; the boldness of the $\mathrm{Ind}(X_1, X_2)$ method is balanced by its comparatively low strength values.

Table 1. Three cases of sample values for the three methods for combining opinions. In each case the opinions are X_1 and X_2 and the strength of the distributions is denoted by "Str". The right hand column contains the discreetised $\mathbb{P}(\alpha, d)$ values described in the '$\varUpsilon$ Method'. All calculations were performed with a uniform prior.

#1	X_1	0.1000	0.5000	0.2000	0.1000	0.1000	Str = 0.9	$P = \langle 0.9, 0.05, 0.03, 0.01, 0.01 \rangle$
	X_2	0.0500	0.8000	0.0500	0.0500	0.0500	Str = 0.7	$P = \langle 0.7, 0.2, 0.05, 0.03, 0.02 \rangle$
	Dep	0.0919	0.5590	0.1653	0.0919	0.0919	$c_G \approx 1$	
	$\varUpsilon$	0.0700	0.7000	0.1700	0.0700	0.0700	$c_G = 0.95$	
	Ind	0.0978	0.6044	0.1022	0.0978	0.0978	$c_G = 0.53$	
#2	X_1	0.1000	0.6000	0.1000	0.1000	0.1000	Str = 0.8	$P = \langle 0.8, 0.1, 0.04, 0.01, 0.01 \rangle$
	X_2	0.0500	0.8000	0.0500	0.0500	0.0500	Str = 0.9	$P = \langle 0.9, 0.06, 0.03, 0.01, 0.01 \rangle$
	Dep	0.0683	0.7266	0.0683	0.0683	0.0683	$c_G \approx 1$	
	$\varUpsilon$	0.08	0.63	0.08	0.08	0.08	$c_G = 0.97$	
	Ind	0.0601	0.7596	0.0601	0.0601	0.0601	$c_G = 0.72$	
#3	X_1	0.0500	0.8000	0.0500	0.0500	0.0500	Str = 0.8	$P = \langle 0.8, 0.1, 0.04, 0.01, 0.01 \rangle$
	X_2	0.0500	0.8000	0.0500	0.0500	0.0500	Str = 0.9	$P = \langle 0.9, 0.06, 0.03, 0.01, 0.01 \rangle$
	Dep	0.0573	0.7707	0.0573	0.0573	0.0573	$c_G \approx 1$	
	$\varUpsilon$	0.05	0.8	0.05	0.05	0.05	$c_G = 0.97$	
	Ind	0.0363	0.8548	0.0363	0.0363	0.0363	$c_G = 0.83$	

5 Reputation of the Agents

In the previous section we described how a mediator could assist agents to agree on a common opinion, or reputation, of some thing being evaluated [17]. Additionally, the institution ξ builds a view of the reputation of the individual agents who perform the evaluations by observing the process illustrated in Figure 1. In particular, ξ observes the development of the c_i values (described below), the distances between initial opinion O_i and considered opinion $O_i|\varDelta$, and the distances between both opinions and the group reputation R_G when it exists.

Given two opinions X_1 and X_2 the *strength* of X_1 on X_2 is defined as: $\mathbb{P}(X_1 = X_2)$. If X_1 and X_2 are both defined over the same valuation space $E = \{e_i\}_{i=1}^n$ then: $\mathbb{P}(X_1 = X_2) = \sum_i P(W = w_i) \mid X_1 = X_2$, where $W = (X_1, X_2)$ is the joint distribution. That is, we sum along the diagonal of the joint distribution. We estimate the diagonal w_i values using the dependent estimate: $\mathbb{P}(X_1 = e_i) \wedge \mathbb{P}(X_1 = e_i) = \min_j \mathbb{P}(X_j = e_i)$, and hence: $\text{Str}(X_1, X_2) = \mathbb{P}(X_1 = X_2) = \sum_i \min_j \mathbb{P}(X_j = e_i)$. A measure of the distance between X_1 and X_2 is then: $\text{Dist}(X_1, X_2) = 1 - \text{Str}(X_1, X_2)$. This definition of strength is consistent with the 'Dependent Method' in Section 4 that is the basis of the reputation mediation procedure. Other definitions include the Kullback-Leibler divergence, $\text{Dist}(X_1, X_2) = \mathbb{K}(X_1 || X_2)$, and the earth movers distance [14].

Each time a reputation R_G is formed, the c_i values are updated using: $c_i|\varDelta = \mu \times \text{Dist}(O_i, R_G) + (1 - \mu) \times c_i$, where μ is the learning rate. These c_i values are the product of successive social evaluation processes, and so they are reputation estimates.

The measures described above do not take the structure of the evaluation space E into account. Four additional measures are:

A generic distance measure. $\text{Dist}(X, Y) = \mathbb{K}(X' || Y')$ where (X', Y') is a permutation of (X, Y) the satisfies $X' < Y'$, and the order is defined by: $R_G < O_i|\varDelta < O_i$. I.e. the earliest occurring distribution "goes in the second argument". This complication with ordering is necessary because $\mathbb{K}$ is not symmetric; it attempts to exploit the sense of relative entropy. An alternative is to use the symmetric form as it was originally proposed: $\frac{1}{2}(\mathbb{K}(X, Y) + \mathbb{K}(Y, X))$

A distance measure when the prior, Z, is known. This builds on the generic measure, and captures the idea that the distance between a pair of unexpected distributions is greater than the difference between a pair of similar, expected distributions. We measure of how expected X is by: $\mathbb{K}(X, Z)$, and normalise it by: $max_I \mathbb{K}(I, Z)$ to get: $e(X) = \frac{\mathbb{K}(X,Z)}{max_I \mathbb{K}(I,Z)}$. Then this measure is the arithmetic product of the previous generic measure with: $\frac{e(X)+e(Y)}{2}$.

A semantic distance measure. Suppose there is a difference measure $\text{Diff}(\cdot, \cdot)$ defined between concepts in the ontology. Then the distance between two opinions X and Y over valuation space E (represented as distributions p_i and q_i respectively) is: $\text{Dist}(X, Y) = \sum_{ij} p_i \times q_j \times \text{Diff}(e_i, e_j)$ where e_i are the categories in E.

A distance measure when E is ordered and the prior is known. If the valuation space E has a natural order, and if there is a known prior then define $\text{Diff}(e_i, e_j)$ to be the proportion of the population that is expected to lie between e_i and e_j. Then define $\text{Dist}(X, Y) = \sum_{ij} p_i \times q_j \times \text{Diff}(e_i, e_j)$. For example, in conference reviewing, if the expectation is that 40% of reviews are 'weak accept' and 20% are 'accept' then Diff('weak accept', 'accept')$= \frac{40}{2} + \frac{20}{2}$; i.e. taking the mid points of the intervals.

The measures described for $\text{Dist}(X, Y)$ are now used to enable ξ to attribute various reputations to agents. These reputation measures all assume that the agents have been involved in a number of successive social evaluation rounds.

Inexorable. If agent β_i is such that: $\text{Dist}(O_i, O_i|\Delta) \ll \text{Dist}(O_i, O_j|\Delta), \forall j \neq i$ consistently holds then β_i is *inexorable.*

Predetermination. If: $\text{Dist}(O_i, R_G) \ll \text{Dist}(O_j, R_G), \forall j \neq i$ consistently, then β_i is a good '*predeterminer*'. Such an agent will have a high c_i value.

Persuasiveness. If β_i is such that: $\text{Dist}(O_i, O_j|\Delta) \ll \text{Dist}(O_j, O_j|\Delta), \forall j \neq i$ consistently then β_i is *persuasive.*

Compliance. If β_i is such that: $O_i|\Delta \approx \arg\min_X \sum_{j\neq i} \text{Dist}(O_j|\Delta, X)$, then β_i is *compliant.*

Dogmatic. If β_i is such that: $O_i = O_i|\Delta$ consistently then β_i is *dogmatic.* A dogmatic agent is highly inexorable.

Adherence. If β_i is such that $O_i|\Delta = O_j$ where $j = \arg\max_{k,k\neq i} c_k$ consistently then β_i is *adherent* (in this round adherent to agent β_j).

6 Discussion

Reputation measures are becoming a cornerstone of many applications over the web. This is the case in recommender systems or in trading mediation sites. In these applications there is a need to assess, for instance, how much should we trust the recommendation coming from an unknown source, or how reliable a trading partner is. This paper has proposed a number of methods to ground the social building of reputation measures. The methods are based on information theory and permit to combine opinions when there is a high level of independence in the formation of the individual opinions. The method permits the computation of reputation values as aggregation of individual opinions, and also detects when agreement is not feasible. This impossibility may be used to trigger further discussions among the members of the group or to introduce changes in the composition of the group to permit agreements.

The use of social network analysis measures permits to define heuristics on how to combine opinions when there is no complete independence in the opinions expressed by the agents. There are a number of different relationships that may be used to guess dependency. For instance, in the context of scientific publications, co-authorship or affiliation, meaning that authors have written papers together or belong to the same laboratory may indicate a significant exchange of information between them and therefore a certain level of dependency. The aggregation of values by function h can then use these measures to diminish the joint influence of dependent opinions into the reputation. This is to be explored in future extensions of the information based reputation model.

References

1. Sierra, C., Debenham, J.: Information-based agency. In: Proceedings of Twentieth International Joint Conference on Artificial Intelligence IJCAI-07, Hyderabad, India, January 2007, pp. 1513–1518 (2007)
2. Friedkin, N., Johnsen, E.: Social influence networks and opinion change. Advances in Group Processes 16, 1–29 (1999)
3. Surowiecki, J.: The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations. Random House, New York (2004)
4. Sabater, J., Sierra, C.: Review on computational trust and reputation models. Artificial Intelligence Review 24(1), 33–60 (2005)
5. Viljanen, L.: Towards an ontology of trust. In: Katsikas, S., Løpez, J., Pernum, G. (eds.) TrustBus 2005. LNCS, vol. 3592, pp. 175–184. Springer, Heidelberg (2005)
6. Huynh, T., Jennings, N., Shadbolt, N.: An integrated trust and reputation model for open multi-agent systems. Autonomous Agents and Multi-Agent Systems 13(2), 119–154 (2006)
7. Arcos, J.L., Esteva, M., Noriega, P., Rodríguez, J.A., Sierra, C.: Environment engineering for multiagent systems. Journal on Engineering Applications of Artificial Intelligence 18 (2005)
8. Sierra, C., Debenham, J.: Information-based agency. In: Veloso, M. (ed.) Twentieth International Joint Conference on AI (January 2007) (in press)
9. Sierra, C., Debenham, J.: The LOGIC Negotiation Model. In: Proceedings Sixth International Conference on Autonomous Agents and Multi Agent Systems AAMAS-2007, Honolulu, Hawai'i, May 2007, pp. 1026–1033 (2007)
10. Klimek, P., Lambiotte, R., Thurner, S.: Opinion formation in laggard societies. EPL (Europhysics Letters) 82(2) (2008)
11. Cheeseman, P., Stutz, J.: On The Relationship between Bayesian and Maximum Entropy Inference. In: Bayesian Inference and Maximum Entropy Methods in Science and Engineering. American Institute of Physics, Melville, NY, USA, pp. 445–461 (2004)
12. Paris, J.: Common sense and maximum entropy. Synthese 117(1), 75–93 (1999)
13. Kuter, U., Golbeck, J.: Sunny: A new algorithm for trust inference in social networks using probabilistic confidence models. In: Proceedings of the 22nd national conference on Artificial intelligence (AAAI'07), pp. 1377–1382. AAAI Press, Menlo Park (2007)
14. Peleg, S., Werman, M., Rom, H.: A unified approach to the change of resolution: Space and gray-level. IEEE Transactions on Pattern Analysis and Machine Intelligence 11(7), 739–742 (1989)
15. Yager, R.R.: On the determination of strength of belief for decision support under uncertainty — part i: generating strengths of belief. Fuzzy Sets and Systems 142(1), 117–128 (2004)
16. Pinyol, I., Sabater-Mir, J., Cuní, G.: How to talk about reputation using a common ontology: From definition to implementation, pp. 90–101 (2007)
17. Jelasity, M., Montresor, A., Babaoglu, O.: Gossip-based aggregation in large dynamic networks. ACM Transactions on Computer Systems 23(3), 219–252 (2005)

The Open Metaverse Currency (OMC) – A Micropayment Framework for Open 3D Virtual Worlds

Frank Kappe and Michael Steurer

Institute for Information Systems and Computer Media
Graz University of Technology, Austria
{frank.kappe,michael.steurer}@iicm.tugraz.at

Abstract. Virtual worlds have become very popular recently, in particular 3D virtual worlds with a built-in virtual currency that enables providers and users to monetize their creations. Virtual Worlds based on the relatively new open-source and free-to-install "OpenSimulator" software currently lack such a virtual currency framework, in particular one that can be used across the boundaries of such virtual worlds (also known as grids). In this paper we present the design of such a micropayment system for this class of virtual worlds. The functionality expected by the users of such a 3D environment is somewhat different from what is usually expected from Web payment systems. A particular challenge is the fact that parts of the software that our system is to be integrated with cannot be trusted, as they are run by unknown parties, and can easily be modified (as it is open-source software). Transaction values in such a virtual environment are usually very small, so the cost of a transaction is also of concern. The system has already been implemented and is used by a number of such virtual worlds.

Keywords: 3D Virtual Worlds, Micropayment, Virtual Currency, OpenSimulator, Hypergrid, Open Metaverse.

1 Introduction

Virtual worlds are a special class of social software, where users are represented by avatars in a virtual environment. The avatars interact with each other in real time using chat or voice chat. In contrast to most other social networks, the users behind the avatars typically remain anonymous. Unlike Massively Multiplayer Online Games (MMOGs), which use similar technology, virtual worlds are not a game, with predefined goals, winners, and losers, but are more open-ended, with a focus on user interaction and creativity.

Virtual worlds have become very popular and are still growing [1]. A recent study by virtual world research firm Kzero [2] estimates over 800 million users in over 300 different virtual worlds. However, these virtual worlds are very different in nature. The majority targets children and teenagers as customers, but there are others where the average user is over 30 years old. Many of the virtual worlds are only two-dimensional in nature, running in a web browser.

F. Buccafurri and G. Semeraro (Eds.): EC-Web 2010, LNBIP 61, pp. 97–106, 2010.

Yesha Sivan defines as “Real Virtual Worlds” [3] virtual worlds which share the “3D3C” properties, *i.e.* three-dimensional, with an emphasis on Community, Creation, and Commerce. An example of such a “3D3C Real Virtual World” is Second Life [4].

Real virtual worlds allow users to create virtual goods (*e.g.* clothing, buildings, furniture, vehicles, weapons, animations, scripted objects), and sell them to other users who lack the skills or time to create them for themselves [5]. The global market for virtual goods is estimated to be approximately EUR 1,5 billion a year [6] with Second Life, Tencent, IMVU, Gaia Online, and Habbo Hotel among the leading players. In the virtual economy of Second Life alone, about 2 million US$ worth of virtual goods and services are traded between users on an average day [7]. According to statistical data published by Linden Lab [8], the company that runs Second Life, about 69.000 users made a profit in Second Life in February 2010. In 2009, there existed a few Second Life entrepreneurs, whose profits exceed 1 million US$ per year [9].

However, there is a significant obstacle for further growth: The current virtual worlds are walled gardens, completely isolated from each other. The situation can be compared to the pre-Web Internet, with companies like Compuserve, America Online, Prodigy, The Source, and others competing for users and publishers. The incompatibility required publishers to take bets on where to invest their money, which limited the growth of this industry. Finally, the Web came along, with open standards and open-source implementations, and removed this barrier to entry for both publishers and users.

2 Introducing the Open Metaverse

The term “Metaverse” has been coined in 1992 by Neal Stephenson in his science fiction novel “Snowcrash” [10], where he described an immersive 3D virtual world. In fact, this book has strongly influenced the design of virtual worlds such as Active Worlds and Second Life, which can therefore be regarded as an implementation of this vision. We will use the term “Open Metaverse” to describe virtual worlds like Second Life, but implemented using open-source software such as OpenSimulator (also known as OpenSim), Sun’s Project Wonderland, and OpenCroquet. In addition to these “de facto” standards, a few “official” standardization initiatives try to standardize communication protocols and file formats of virtual worlds [11].

The vision of the open metaverse is to do for the 3D Internet what Apache and Mozilla did for the 2D Web. In fact, some limited interoperability (teleporting) between Second Life and OpenSim has already been demonstrated, and we already see some OpenSim-based virtual worlds appear [12].

However, the situation with 3D worlds is more complicated than in the 2D Web: Even if all virtual worlds would use the same protocol, viewer, and 3D object representation, it still would not be possible to *e.g.* buy a virtual good in one world, and use or sell it in another. Today’s virtual worlds are like islands, each with their own currency, user profiles, permissions system, and asset repositories.

However, this may soon change. The Hypergrid [13] is an extension to the OpenSim protocol which enables teleporting between OpenSim-based virtual worlds (which are called grids). You can even take virtual assets (*e.g.* clothing) with you (more precisely, your avatar) as you roam the OpenSim Hypergrid.

A key component for making the Open Metaverse a "real virtual world" according to Sivan's definition is the "commerce" aspect. This in turn requires a payment system, which is not yet part of the OpenSim framework.

3 Requirements

Designing and implementing a payment system for the OpenSim environment poses some interesting requirements (and challenges), both technical and non-technical. We can divide these issues roughly into the three groups functionality, trust, and cost.

3.1 Functionality

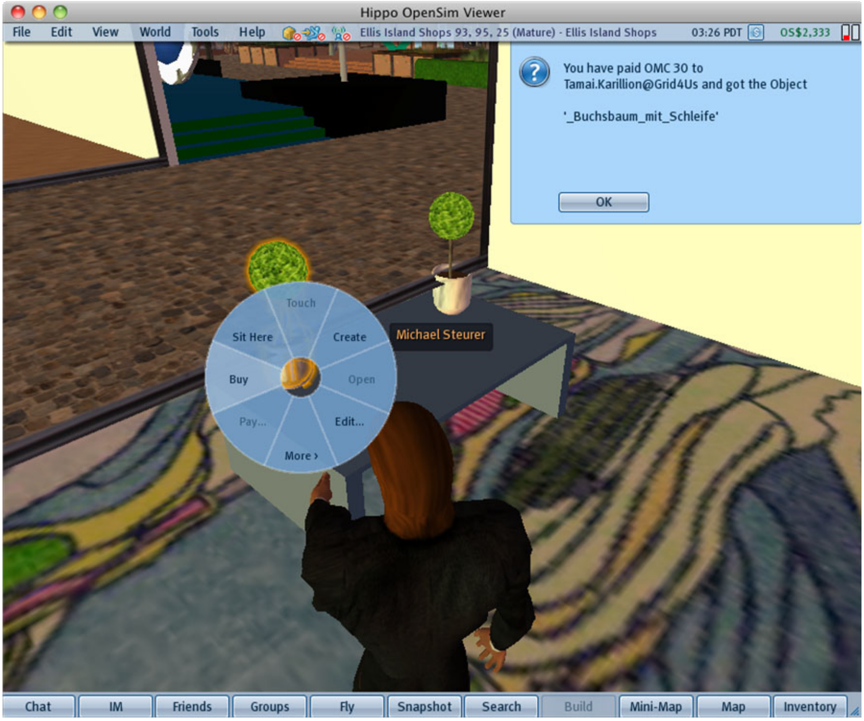
The functional requirements of a payment system for 3D virtual worlds are in some aspects quite different from the "traditional" payment systems found in the 2D Web or even the real world. In the OpenSim case, the expectation of the users is to find the same level of functionality available in the virtual world "Second Life", which is arguably the most successful implementation of a micropayment system, with 50 million US$ worth of transactions per month [14]. Also, as we will describe in more detail in the "System Architecture" section, the client software used is compatible with the Second-life client (which is open source). Therefore it makes sense to model the micropayment system for OpenSim after what is available in Second Life.

Unlike earlier implementations [15], we did not want to use in-world objects representing virtual wallets or head-up displays, because of the negative impact on simulator performance and the necessity to allow script permissions to everybody. Specifically, our micropayment system has to support the following use cases:

1. User A pays user B, while both users are online in the same simulator and see each other. In the user interface, this can be done by user A right-clicking on the avatar of user B, and selecting "pay" from a pie-chart menu.
2. Like use case 1, but user B is not online (or out of sight). Still, user A can bring up B's profile and select "pay" in the profile window.
3. User A likes to pay an object in the virtual word owned by user B. This can be achieved in the user interface by right-clicking on the object, and then selecting "pay" similar to use case 1. The owner of the object (B) will receive the money, but in addition an event (the money-event) is raised in the script running in the object, so that the object can react to it.
4. User A wants to buy an object in the virtual word owned by user B. Again, this is done by user A right-clicking the object and then selecting "buy" in the pie menu (see Figure 1). User B has to set the object's properties to "sellable" and set a price beforehand. User B will get the money; in addition the object (or a copy of it) will be delivered to the inventory of user A.
5. User A wants to buy land currently owned by user B. User A initiates the process by selecting "buy land" form the land properties page. User B has set the land properties to "sellable" and set a price beforehand. Once the money has been paid, the OpenSim database will be updated to reflect the transfer of ownership.

6. An object owned by user B pays (virtual) money to user A. Note that user B will typically not be on-line when this happens (nor does user A have to be online). Therefore, the owner (B) will have to specifically give debit permissions to the object beforehand, to access the user's account and send money on the user's behalf without further confirmation.

These are the six cases necessary for a smooth in-world user experience similar to that of Second Life. Additional use cases were included to support buying objects and the like from more traditional user interfaces like a Web shop, but these are less interesting and beyond the scope of this paper.



You have paid OMC 30 to Tamai.Karillion@Grid4Us and got the Object

'_Buchsbaum_mit_Schleife'

Fig. 1. "Object Buy" dialog with the notification on the payment made

3.2 Security and Trust

Obviously, security is a key concern of any payment system. In a single-provider virtual world (like Second Life), the provider will usually not only run the virtual world as such, but also be providing the micropayment system, as well as means to buy the virtual currency (an exchange), and take the role of the central bank responsible for keeping the value of the currency stable. Therefore, the user will have to trust only the provider of the virtual world.

In contrast, in the OpenSim environment, the people running the individual servers (so-called simulators) are many, and users should not need to trust them. Therefore, it makes sense to separate the functionality of the payment system provider from that of the provider of the virtual world, so that the payment authorization process effectively bypasses the unsecure OpenSim servers. Users will of course still have to trust the payment system provider, but not necessarily the provider of the specific simulator they are currently in, as the (virtual) money never leaves the realm of the payment provider.

For the user experience, this will mean that – unlike in Second Life – for the use cases 1 – 5 the payment will have to be confirmed at the external website of the payment system provider, where users will have to have an account to participate in the virtual commerce. This is somewhat more inconvenient than a complete in-world experience, but is the price to be paid for security. An alternative would be to embed the payment functionality in a modified viewer, and enforce its use, which would probably be seen as even more inconvenient.

Use case six is a special security concern. The owner of the object paying (virtual) money on the owner's behalf will have to not only confirm the debit permissions in the virtual world, but in addition on the payment providers website. This is necessary to restrict any provider of a simulator to fake these permissions. Still, the script running inside the object paying out can be read and modified by the administrator of the simulator, which is why the users using this functionality must trust the simulator provider, or preferably run the simulator themselves.

3.3 Transaction Costs – Virtual Currency

The value of a user-to-user transaction in a virtual world is typically very small, *i.e.* less than a Dollar or Euro. It is therefore important to keep the transaction costs for user-to-user transactions close to zero. This requirement rules out the classic forms of payments like credit cards, debit cards, or PayPal.

The solution that most virtual worlds have adopted is to use a virtual currency, with all user-to-user transactions taking place in the virtual currency within the realm of the provider of the virtual world, usually without any transaction fee. Fees are only charged when the virtual currency is bought or sold in exchange for real-world currency. This typically happens in larger amounts than the individual user-to user transactions and consequently this exchange fee is less of an issue.

In addition to the reduction in transaction cost, using a virtual currency offers the users the benefit that the goods can be offered worldwide, but without the hassle of constantly adjusting the prices in real-world currencies as a result of moving exchange rates. For the provider, using a virtual currency can reduce the liability of the provider in case things go wrong, *e.g.* Linden Lab states in the Terms of Service of Second Life:

"Regardless of terminology used, Linden Dollars represent a limited license right governed solely under the terms of this Agreement, and are not redeemable for any sum of money or monetary value from Linden Lab at any time. You agree that Linden Lab has the absolute right to manage, regulate, control, modify and/or eliminate such Currency as it sees fit in its sole discretion, in any general or specific case, and that Linden Lab will have no liability to you based on its exercise of such right." [16]

For these reasons, we have chosen to implement our micropayment system by introducing a new virtual currency, the Open Metaverse Currency (OMC), with the Open Metaverse Cent (OM¢) as currency unit. To be fully functional, this requires that in addition to the micropayment system itself, there is a framework including an exchange that lets users buy and sell OMC for real-world currency, and a kind of central bank that controls the money supply. In this paper, however, we concentrate on the aspects of the micropayment system.

4 Architecture

In this section we show the design and the architecture of the entire payment mechanism but focus on a general view instead of detailed implementation issues.

4.1 Modified Event Handling

In Section 1.1 we have already described use cases of user interactions that require a transfer of money. All these mechanisms are already implemented in the client software, further referred to as OpenSim Viewer or in the server software, further referred to as Simulator. The Simulator has a modular design and the modules detect events sent by the OpenSim Viewer with event listeners and act on the received parameters with event handlers. Existing server implementations are not aware of money and, for instance, immediately deliver objects with the event handler in case of a “Buy Object” event initiated by the client and detected by the event listener.

To add money capabilities to the Simulator we have to split the event listener from the event handler and add the entire money transfer process in between. The methods in the event handler will only be executed if the actual payment succeeded. Figure 2 depicts this extension with an additional step in between the event handler and the event listener.

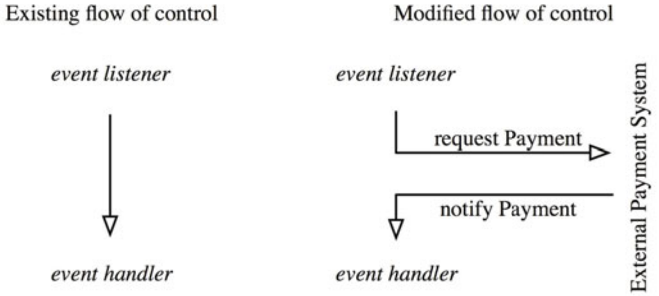


Fig. 2. Split the event listener from the event handler in the simulator to do the payment

The structure of a grid comprises numerous different simulators responsible for all listen-events, event handlers, and everything in between. The source code of the simulators is publicly available and allows to run the service with potentially maliciously modified sources. This implies that every simulator owner could abuse a user's money account. To get rid of this problem the module that implements the event listener and event handler for money transfer does not contain any *control logic* but redirects all money related events to an external and trusted web service referred to as “Money Gateway”. It puts all simulators under one umbrella and connects Simulators with the payment provider.

If the simulator detects a money-related event it sends a request to this centralized instance for further processing and waits for a notification to continue with the event handler. The money gateway employs the payment provider for the actual money transfer and therefore every user needs an account there. Users provide the payment sensitive information, *e.g.* login credentials, only to the payment provider and therefore bypass the untrusted simulator and the money gateway.

4.2 Bypassing Untrusted Software for Security

In the previous section we have stated that the actual payment process is hidden from the Simulator due to security concerns. To do so, a different and secure channel is needed because the client software suffers from the same problems as the simulator does – it cannot be trusted. To get rid of this constraint we establish the secure connection between the user and the payment provider with a common (and trusted) web browser. Secure HTTP connections (HTTPs) provide sufficient authentication mechanisms and cannot be wiretapped.

In case of a money related event the money gateway processes the simulator's request and redirects the user to the website of the payment provider in an external web browser. Most payment providers offer special programming interfaces to create payment requests that only need to be confirmed by a user. To do so, the user needs an account with the payment provider, provides credentials for the authentication, and finally confirms the payment created by the money gateway. On success the money is transferred and the payment provider informs the money gateway about the successful payment. The money gateway redirects this notification to the simulator and prompts the interrupted event to be continued.

Figure 3 depicts a block diagram of the involved parties for a money transfer. As mentioned in Section 1.2 we cannot trust the OpenSim client viewer and simulator, and therefore need the bypass to directly communicate using a trusted web browser. Figure 4 shows an example of a message exchange for handling a money event. A user requests a money event as mentioned in Section 3.1 which is detected by the Simulator's event listener. It extracts all required parameters needed for the actual transaction and transmits them to the gateway. To give an example, a simple transaction between two avatars involves parameters to identify avatars, the amount of money, and the current location where the transaction takes place. All these parameters are stored in the gateway's database and the user is requested to confirm the outstanding payment. After this confirmation the gateway queries the database for the stored parameters and sends them to the Simulator. The Simulator processes this notification according to the extracted parameters.

The benefit of this approach is a complete separation between the requests and the notifications because all parameters are stored in the gateway's database and sent to the Simulator upon a payment confirmation. Further, the Simulator does not store any parameters but just passes them to the gateway and process them in case of a notification.

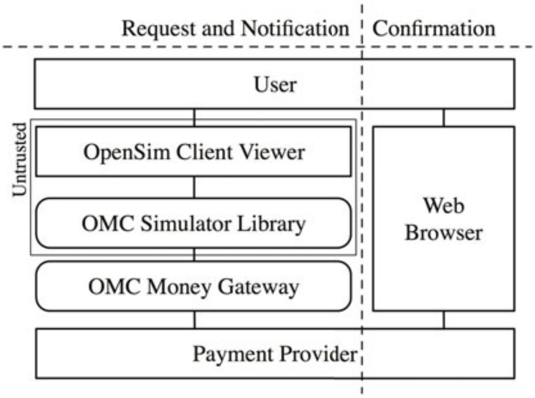


Fig. 3. General architecture of a secure and trustworthy payment system in OpenSim based environments

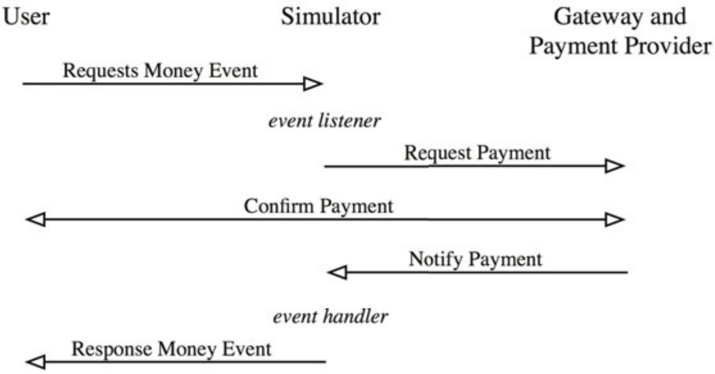


Fig. 4. Requests, confirmation, and notifications for a successful and secure money transfer

4.3 Design Limitations

We have already described the requirement of a secure channel for the payment confirmation due to the untrusted client viewer and simulator. The early stage of the entire simulator development raises another critical problem related to the security of the notification messages. All notifications are unencrypted XML-RPC and could origin from the money gateway, but also from anywhere else. Currently, the only available way to exchange information over an encrypted and authenticated channel is a secured HTTP request from the simulator to the money gateway.

Instead of directly transmitting information after a successful payment to the Simulator, the money gateway just sends a random 128-bit identifier by unencrypted XML-RPC. This identifier can be employed to fetch the necessary parameters from the gateway needed for further processing via secured HTTP link. It is not possible to repeat this process by sending an XML-Request with the same 128-bit identifier twice because the gateway would refuse the request for the according parameters. However, we consider the term “notification” as a synonym for the entire information transfer from the money gateway to the simulator regardless of the needed workaround due to unencrypted XML-RPC.

The simulator needs to store all received parameters from the Viewer to be able to continue the execution on a notification of a successful payment. Unfortunately, the listen-event and the “Confirm Payment” message from Figure 4 are not necessarily temporally close. As described in Section 1.4 we have extracted the entire control logic from the simulator module to make the layer as thin as possible. We send all parameters received by the event listener to the money gateway but do not store it on the simulator. To continue with the processing on a successful payment all necessary parameters are sent with the notification message. The additional overhead of the notification message due to the sent parameters is acceptable because the stateless design makes the entire system more reliable and easier to monitor.

5 Conclusions

The complete micropayment framework has been implemented and put into actual use, with the Austrian company VirWoX [17] as external payment system provider and money exchange. Obviously, it is crucial for the acceptance of the new currency and the associated payment system that the provider enjoys the trust of the users. VirWoX has earned a reputation as the leading independent exchange for trading Linden dollars (the virtual currency used in Second Life) in exchange for real-life currency. In business since beginning of 2008, until the time of writing (June 2010) about 4 billion Linden dollars (worth about 12 million Euros) have been exchanged there. Therefore we believe that OpenSim users will trust the new currency.

At the time of writing, about 600 simulators in 10 different grids are connected to the system (see Figure 5). 250 users have registered an account, and 260,000 OM¢ are in circulation.

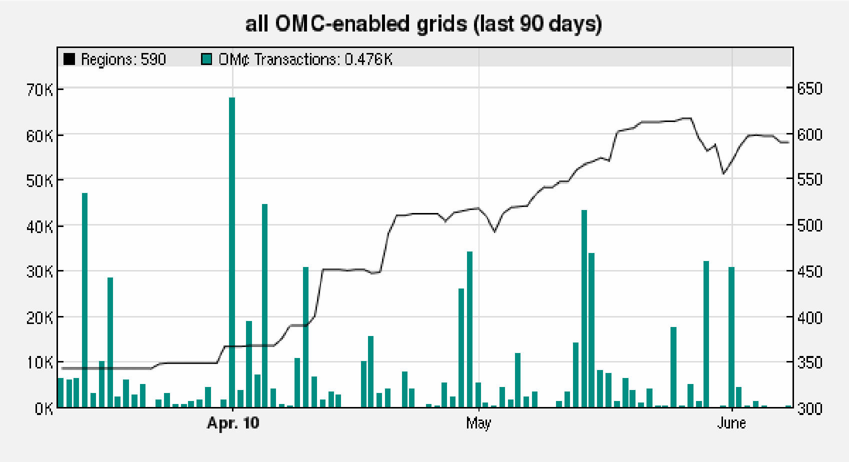


Fig. 5. Active regions versus transaction volume

As there are not many items to buy yet, it is not surprising that the number of transactions is still relatively low. In the first 100 days of operation, about 900,000 OM¢ have been transferred between 211 different users, in about 1,400 transactions [18]. However, we are confident that the transaction volume will grow as the user numbers of OpenSim based worlds rise.

References

1. Gartner Inc.: 80 Percent of Active Internet Users Will Have A "Second Life" in the Virtual World by the End of 2011 (2007), http://www.gartner.com/it/page.jsp?id=503861
2. Mitham, N.: Virtual Goods: Good for Business? Journal of Virtual Worlds Research 2(4) (2010), https://journals.tdl.org/jvwr/article/view/864/629
3. Sivan, Y.: 3D3C Real Virtual Worlds Defined: The Immense Potential of Merging 3D, Community, Creation, and Commerce (2008)
4. Rymaszewski, M., Au, W.J., Wallace, M., Winters, C., Ondrejka, C., Batstone-Cunningham, B., Rosedale, P.: Second Life: The Official Guide. SYBEX Inc. (2006)
5. Chesney, T., Noke, H.: Virtual World Commerce: An Exploratory Study. SSRN eLibrary (2008), http://ssrn.com/paper=1286036
6. ENISA: Virtual Worlds - Real Money. European Network and Information Security Agency (2008), http://www.enisa.europa.eu/media/press-releases/2008-prs/virtual-worlds-real-money
7. Linden Lab.: 2009 End of Year Second Life Economy Wrap up (2009), https://blogs.secondlife.com/community/features/blog/2010/01/19/2009-end-of-year-second-life-economy-wrap-up-including-q4-economy-in-detail
8. Linden Lab.: Economic Statistics, http://secondlife.com/statistics/economy-data.php
9. Wagner, J.: Top Second Life Entrepreneur Cashing Out US$1.7 Million Yearly; Furnishing, Events Management Among Top Earners, http://nwn.blogs.com/nwn/2009/03/million.html
10. Stephenson, N.: Snow Crash. Bantam Books (1992)
11. Jakobs, K.: Real Standards for Virtual Worlds, Why and How? Journal of Virtual Worlds Reserach 2(3) (2009), https://journals.tdl.org/jvwr/article/view/717/517
12. Childers, B.: Run your own virtual reality with OpenSim. Linux Journal 2009 6 (2009)
13. Fishwick, P.A.: A Introduction to OpenSimulator and virtual environment agent-based M&S applications. In: Proceedings of the 2009 Winter Simulation Conference (2009)
14. Linden Labs.: 1 Billion Hours, 1 Billion Dollars Served: Second Life Celebrates Major Milestones for Virtual Worlds (2009), http://lindenlab.com/pressroom/releases/22_09_09
15. Schrank, D., Gütl, C., Kappe, F.: Design and Implementation of a Payment System for Virtual Worlds. In: WASET 2010, Paris, France (to appear, 2010)
16. Linden Labs.: Second Life: Terms of Service, http://secondlife.com/corporate/tos.php
17. VirWoX: The Virtual World Exchange, http://www.virwox.com
18. VirWoX: Economy Data for the OMC, http://www.virwox.com/open-metaverse-economy.php

T-REX: A Hybrid Agent Trust Model Based on Witness Reputation and Personal Experience

Kalliopi Kravari, Christos Malliarakis, and Nick Bassiliades

Dept. of Informatics, Aristotle University of Thessaloniki, GR-54124 Thessaloniki, Greece
{kkravari,malliara,nbassili}@csd.auth.gr

Abstract. Semantic Web will transform the way people satisfy their requests letting them delegate complex actions to intelligent agents, which will act on behalf of their users into real-life applications, under uncertain and risky situations. Thus, trust has already been recognized as a key issue in Multi-Agent Systems. Current computational trust models are usually built either on an agent's direct experience or reports provided by others. In order to combine the advantages and overcome the drawbacks of these approaches, namely interaction trust and witness reputation, this paper presents a hybrid trust model that combines them in a dynamic and flexible manner. The main advantage of our approach is that it provides a reliable and flexible model with low bandwidth and storage cost. Moreover, we present the integration of this model in JADE, a multi-agent framework and provide an evaluation and an e-Commerce scenario that illustrate the usability of the proposed model.

Keywords: Trust, Reputation, Multi-Agent Systems.

1 Introduction

Intelligent agents (*IAs*) are considered the most prominent means towards realizing the Semantic Web (SW) vision [1]. In the future, via the gradual integration of *multi-agent systems* (*MAS*) with SW technology, complex tasks will be efficiently performed by agents with less or no human intervention. Nevertheless, a critical issue is now raised: how can an agent trust an unknown partner in an open and thus risky environment?

To this end, a number of researchers have proposed, in different perspectives, models and metrics of trust, focusing on estimating the degree of trust that can be invested in a certain agent [2]. Current computational trust models are usually built either on *interaction trust* or *witness reputation*, namely an agent's direct experience or reports provided by others, respectively.

However, both approaches have limitations; for instance when an agent enters an environment for the first time, it has no history of interactions with the other agents in the environment and thus there is no available information. Thus, if the trust estimation is based only on direct experience, it would require a long time to reach a sufficient amount of interactions that could lead to a satisfying estimation.

On the other hand, models based only on witness reports could not guarantee reliable estimation as self-interested agents could be unwilling or unable to sacrifice their

F. Buccafurri and G. Semeraro (Eds.): EC-Web 2010, LNBIP 61, pp. 107–118, 2010.

resources in order to provide reports. In addition, in the distributed and open environments, such as MAS, there is no centralized authority to observe and control the environment. As a result, the witness reports could be difficult to locate. This is usually overcome by using some form of centralized mechanism to collect all the reports [3]. Hence, models based only on one or the other approach typically cannot guarantee stable and reliable estimations.

In order to overcome the above drawbacks, in this paper we propose T-REX, a novel hybrid trust model that combines both *interaction trust* and *witness reputation* in a dynamic and flexible manner, letting the final user to proportion their active participation in the final estimation. The novelty of our approach is that T-REX could vary from a completely witness-based system to a system totally based on personal experience, according to what is more important, more appropriate or more convenient for the end user.

In addition, it provides a stable and reliable estimation mechanism based on certain parameters such as the information correctness and completeness and the agent's response time. Moreover, the administration responsibility for this model is delegated to a special agent, called *Trustor*, overcoming the difficulty to locate witness reports. Hence, the main advantage of our approach is that it provides a reliable and flexible model with low bandwidth and storage cost. Finally, an evaluation and an e-Commerce scenario are presented that illustrate the usability of the approach.

The rest of the paper is organized as follows. In Section 2, we present our reputation model. Section 3 presents the implementation of the model and its integration into JADE, the popular multi-agent framework. In Sections 4 and 5, the model's evaluation and an e-Commerce scenario are presented, respectively, that illustrate the usability of the approach. Section 6 discusses related work, and Section 7 concludes with final remarks and directions for future work.

2 T-REX

As mentioned, the proposed model, T-REX, is actually a reputation–based centralized system that combines both interaction trust and witness reputation. Its administration tasks are up to a special agent, called *Trustor*, who is responsible for collecting, storing and keeping the reports safe and available. Moreover, whenever an agent requests the reputation value of another agent, the Trustor has to calculate and return the agent's final reputation value according to the model's metric (see subsection 2.1). Notice that Trustor is considered certified by the protocol and, thus, reliable.

2.1 Rating Mechanism

Consider an agent A establishing an interaction with an agent B; agent A can evaluate the other agent's performance and thus affect its reputation. We call the evaluating agent (A) *truster* and the evaluated agent (B) *trustee*. Hence, after each interaction in the environment the *truster* has to evaluate the abilities of *trustee* and report its ratings to *Trustor* in terms of *correctness*, *completeness* and *response time*. The Tustor stores the information of these ratings in the form:

$$\overrightarrow{R_{ab}(t)} = \langle Corr^{ab}(t) | Resp^{ab}(t) | Comp^{ab}(t) \rangle = \left\langle r_1^{ab}(t) \middle| r_2^{ab}(t) \middle| r_3^{ab}(t) \right\rangle \qquad (1)$$

where *a*, *b*, *Corr*, *Resp* and *Comp* stands for truster, trustee, correctness, response time and completeness, respectively. In order to ease formulae presentation, we map *Corr*, *Resp*, and *Comp*, to indices 1, 2, 3, respectively and, thus ratings are represented as r_x. Finally, *t* is a time stamp used by Trustor in order to organize its case records. Note that ratings with a higher time stamp are considered more important as they refer to more recent evaluations. Thus, our approach overcomes the main problem faced in dynamic environments, such as MAS, where agents may change their objectives at any time. As a result, a reliable agent (in t_0) may be transformed into a mercenary and malicious agent (in $t_{1,}$ where $t_1 > t_0$).

In T-REX, ratings vary from 0.1 (terrible) to 10 (perfect); $r_x \in [0.1, 10] \mid x \in \{1,2,3\}$. However, Trustor in order to cross out extremely positive or extremely negative values computes logarithmically each rating. Hence, each rating is normalized (r∈[-1, 1] | -1≡terrible, 1≡perfect) and stored in Trustor's repository. Thus, the final reputation value ranges from -1 to +1, where -1, +1, 0 stand for absolutely negative, absolutely positive and neutral (also used for newcomers), respectively, which means that an agent's reputation could be either negative or positive.

In addition, whenever an agent requests the reputation value of another agent, it can determine (by using weights) how important the above normalized ratings, are for him/her (formula 2). For instance, an agent may be considering more important the response time (e.g. a 50% on the final value) rather than the correctness (35%) or the completeness (15%). Formula 2 calculates the weighted normalized ratings:

$$r_{ab}(t) = \left| \overrightarrow{R_{ab}(t)} \right| = \frac{\sum_{x=1}^{3} \left[p_x \cdot \log\left(r_x^{ab}(t) \right) \right]}{\sum_{x=1}^{3} p_x} \qquad (2)$$

where *a*, *b*, r_x, p_x stand for truster, trustee, ratings and weights (for each corresponding rating), respectively.

Hence, the final reputation value (*TR*) of an agent, at a specific time *t*, is based on the weighted sum of the relevant reports (normalized ratings) stored in Trustor's repository and is calculated according to formula 3.

$$TR_{ab}(t) = \frac{\pi_p}{\pi_p + \pi_o} \cdot \frac{\sum_{\forall t_i < t} [r_{ab}(t_i) \cdot t_i]}{\sum_{\forall t_i < t} t_i} + \frac{\pi_o}{\pi_p + \pi_o} \cdot \sum_{\forall j \neq a, j \neq b} \frac{\sum_{\forall t_i < t} [r_{jb}(t_i) \cdot t_i]}{\sum_{\forall t_i < t} t_i} \qquad (3)$$

There are two important aspects in this formula, considering an agent *a* requests the reputation of another agent *b*. The first one is that the normalized ratings are divided to two groups, one referring to transactions between *a* and *b* (r_{ab}) and one referring to transactions between *b* and each one (*j*) of the rest of the agents (r_{jb}), separating ratings to personal experience and witness reputation, accordingly. The second is that the user, through his/her agent *a*, is able to set what we call the "social trust weights" (π_p, π_o). These weights specify the balance between personal experience (π_p). and witness reputation (π_o), which is actually an opinion provided by strangers. Thus, *TR* value is calculated according to what is more important for the end user, its own experience or the witnesses' opinion. Finally, note that time, as already mentioned, is important and thus it affects the final value, as more recent ratings "weigh" more. This is achieved by multiplying the rating at time t_i, with t_i itself.

2.2 Model's Advantages

T-REX is a dynamic and flexible model, letting the end user not only to determine the importance of each rating criterion but also to even transform the model from an absolutely witness system to an absolutely subjective system based only on past personal experience (as discussed in subsection 2.1). Thus, it can be applied to a variety of applications, such as bargains and negotiations among agents, auctions and e-commerce transactions.

Furthermore, it provides a reliable system, as it is based on a centralized certified mechanism provided by Trustor, the model's special agent. Centralized systems are usually faced with skepticism, as in open MAS agents represent different owners, who may question the trustworthiness of a central authority. However, T-REX overcomes that by providing a certified authority, implemented in JADE, a reliable and widely used multi-agent framework. Hence, Trustor is always honest, calculating and returning the correct *TR* value. Opposed to that, consider a distributed approach where trustees' collect and store ratings referred to them, such as [8][10]. A malicious trustee with low reputation value would provide only the positive ratings in order to con its partners.

In addition, T-REX provides a mechanism with low bandwidth and storage cost. On one hand trustees do not have to transmit or store anything and, on the other hand, trusters just report their rating in only one communication step, e.g. an ACL (Agent Communication Language) message, which has extremely low cost, and, furthermore, need not worry storing it for future reference. Thus, the total trust computational and storage costs are assigned to Trustor. Practically, as discussed in section 3, the Trustor's final computational complexity is just O(1), however its storage complexity is O(n*n), which is inevitable.

3 Implementation

This section presents the integration of the T-REX model into a multi-agent framework. We have chosen JADE [4], the popular MAS Framework. Moreover, in order to reduce the model's computational complexity, the final formula (3) was revised.

3.1 Reducing Computational Complexity

The run-time computational complexity of formula (3) is high, since there is a double sum at the second part of the formula; the inner sum ranges over each preceding time step, whereas the outer sum ranges over all agents, except a and b. Thus, the run-time computational complexity of formula (3) is O(n*t), where n is the number of agents acting in the system and t is the total amount of transactions. Both of these numbers increase rapidly in an open, dynamic and evolving multi-agent environment.

Our goal is to reduce the computational cost at run time. In order to achieve this, Trustor needs to aggregate and store multiple values, in addition to the raw ratings. So, the Trustor does not store the normalized ratings (formula 2) but multiplies them by t_i, their time stamp. In addition, Trustor summarizes the ratings referred to the transactions between two specific agents. Thus, firstly, we replaced the r_{ab} and r_{jb} values in the final formula (3) by using formula 2:

$$TR_{ab}(t) = \frac{\pi_p}{\pi_p+\pi_o} \cdot \frac{\sum_{\forall t_i<t} \left[\frac{\sum_{x=1}^{3} \left[p_x \cdot \log\left(r_x^{ab}(t)\right) \right]}{\sum_{x=1}^{3} p_x} \cdot t_i \right]}{\sum_{\forall t_i<t} t_i} + \frac{\pi_o}{\pi_p+\pi_o} \cdot \sum_{\forall j\neq a, j\neq b} \frac{\sum_{\forall t_i<t} \left[\frac{\sum_{x=1}^{3} \left[p_x \cdot \log\left(r_x^{jb}(t)\right) \right]}{\sum_{x=1}^{3} p_x} \cdot t_i \right]}{\sum_{\forall t_i<t} t_i}$$

$$= \frac{\pi_p}{\pi_p+\pi_o} \cdot \frac{\sum_{x=1}^{3} \left[p_x \cdot \sum_{\forall t_i<t} \log\left(r_x^{ab}(t)\right) \cdot t_i \right]}{\sum_{x=1}^{3} p_x \cdot \sum_{\forall t_i<t} t_i} + \frac{\pi_o}{\pi_p+\pi_o} \cdot \sum_{\forall j\neq a, j\neq b} \frac{\sum_{x=1}^{3} \left[p_x \cdot \sum_{\forall t_i<t} \log\left(r_x^{jb}(t)\right) \cdot t_i \right]}{\sum_{x=1}^{3} p_x \cdot \sum_{\forall t_i<t} t_i} \quad (4)$$

The sum $\sum_{\forall t_i<t} \log\left(r_x^{qy}(t)\right) \cdot t_i$, where $q \in \{a, j\}$ and $y \equiv b$, is not necessary to be calculated at rum time. It can be calculated incrementally, whenever a new report is arrived. We can represent the above sum as $\sigma_x^{qy}(t)$. Similarly, the sum $\sum_{\forall t_i<t} t_i$ can be calculated incrementally (we represent it as $T(t)$). Hence, the formula can be written now in the following form:

$$TR_{ab}(t) = \frac{\pi_p}{\pi_p+\pi_o} \cdot \frac{\sum_{x=1}^{3} \left[p_x \cdot \sigma_x^{ab}(t) \right]}{\sum_{x=1}^{3} p_x \cdot T(t)} + \frac{\pi_o}{\pi_p+\pi_o} \cdot \sum_{\forall j\neq a, j\neq b} \frac{\sum_{x=1}^{3} \left[p_x \cdot \sigma_x^{jb}(t) \right]}{\sum_{x=1}^{3} p_x \cdot T(t)} \quad (5)$$

At this point, the computational complexity is reduced to O(n) compared to O(n*t) in formula 3, since there is only one sum ranging over the number of agents. However, we have a loss in the storage complexity (memory space), inevitably. Storing all these sums leads to O(n^2) complexity, since the quantities $\sigma_x^{qy}(t)$ need to be calculated and stored for each pair q-y of agents, which is actually tolerable. Moving one step further, we notice that variables x and j in formula 5 are independent, thus the sums can swap:

$$TR_{ab}(t) = \frac{\pi_p}{\pi_p+\pi_o} \cdot \frac{\sum_{x=1}^{3} \left[p_x \cdot \sigma_x^{ab}(t) \right]}{\sum_{x=1}^{3} p_x \cdot T(t)} + \frac{\pi_o}{\pi_p+\pi_o} \cdot \frac{\sum_{x=1}^{3} \left[\mathrm{p}_x \cdot \sum_{\forall j\neq a, j\neq b} \sigma_x^{jb}(t) \right]}{\sum_{x=1}^{3} p_x \cdot T(t)} \quad (6)$$

Additionally, calculating and storing the quantities $\sum_{\forall j\neq a, j\neq b} \sigma_x^{jb}(t)$ incrementally, let's call them $S_x^{ab}(t)$, the final revised formula becomes:

$$TR_{ab}(t) = \frac{\pi_p}{\pi_p+\pi_o} \cdot \frac{\sum_{x=1}^{3} \left[p_x \cdot \sigma_x^{ab}(t) \right]}{\sum_{x=1}^{3} p_x \cdot T(t)} + \frac{\pi_o}{\pi_p+\pi_o} \cdot \frac{\sum_{x=1}^{3} \left[p_x \cdot S_x^{ab}(t) \right]}{\sum_{x=1}^{3} p_x \cdot T(t)} \quad (7)$$

The computational complexity is reduced to O(1), since there are no sums depending on the problem size in the formula. On the other hand, the storage complexity remains O(n*n), as the Trustor needs to additionally store some more sums $S_x^{ab}(t)$ for each pair of agents.

3.2 Trustor

Trustor was entirely implemented in Java and deployed in JADE. Practically, it communicates with the rest of the agents via ACL messages with specific communication acts. For instance, a new request (requesting an agent's reputation value – TR) is considered valid only if its communication act is *REQUEST*.

More specifically, whenever an agent wants to reports its ratings regarding another agent, it has to send an ACL message with an *INFORM* communication act to Trustor. The content of this message also has to be in a specific form, containing firstly the agents involved followed by the three ratings, one for each criterion:

$$(\textit{the\_truster's\_name the\_trustee's\_name } r_{corr}\ r_{resp}\ r_{comp})$$

As soon as, Trustor receives a valid report calculates all the necessary sums according to the formulas presented in the previous section and stores the information in its own repository. Later, an agent may ask for someone else's reputation value (via a *REQUEST* ACL message). Once again, the content has to be in a specific form:

$$(\textit{the\_requester's\_name the\_trustee's\_name } p_{corr}\ p_{resp}\ p_{comp}\ \pi_p\ \pi_o),$$

where $p_{corr},$ $p_{resp},$ p_{comp} are the given weights that define the importance of each criterion and $\pi_p,$ π_o are the social trust weights that define the importance of agent's own experience vs. rumor (discussed in the previous section). Then, Trustor calculates the *TR* value and returns it via an ACL message with an *INFORM* communication act.

4 Evaluation

In order to evaluate our model, we use a testbed designed in [5] with slight changes, adopted from [8]. Below, a description of the testbed is given and the next section presents the methodology and the experimental settings for our experiments.

4.1 The Testbed

The testbed environment for evaluating T-REX is a multi-agent system consisting of agents providing services and agents using these services in an on-line community. We assume that the performance of a provider (and effectively its trustworthiness) is independent from the service that is provided (e.g. selling software services or banking services). In order to reduce the complexity of the testbed's environment, it is assumed that there is only one type of service in the testbed. As a result, it is assumed that all the providers offer the same service. Nevertheless, the performance of the providers, such as the quality of the service, differs and determines the utility that a consumer gains from each interaction (called UG≡utility gain).

Each agent interaction is a round in the simulations of the testbed. Events that take place in the same round are considered simultaneous. The round number is used as the time value for events. In the T-REX model, at each round, if a consumer agent needs to use the service it can contact the centralized third-party agent (Trustor) in order to be informed about the reputations of the provider agents.

The consumer agent will select one provider to use its service, the one with the highest value of reputation. The selection process relies on the trust model to decide which provider is likely to be the most reliable. Consumer agents without the ability to choose a trust model will randomly select a provider from the list.

Firstly, the consumer agent selects a provider, then uses the service of the selected provider and gains some utility from the interaction (UG). The value of UG varies from −10 to 10 and depends on the level of performance of the provider in that interaction. A provider agent can serve many users at a time. After an interaction, the consumer agent rates the service of the provider based on the level of performance and the quality of the service it received. It records the rating for future trust evaluations and also informs the provider about the rating it gave. The provider agent (for decentralized models, such as Certified Reputation [8]) or the centralized third-party agent (as in T-REX and SPORAS [3]) record the rating as evidence about its performance to be presented to potential consumers. It is assumed that all agents exchange their information honestly in this testbed. This means an agent (as a witness or as a referee) provides its true ratings as they are without any modification.

4.2 Experimental Methodology

The testbed in each experiment is populated with provider and consumer agents. Each consumer agent is equipped with a particular trust model, which helps it select a provider when it needs to use a service. The only difference among consumer agents is the trust models that they use, so the utility gained by each agent through simulations will reflect the performance of its trust model in selecting reliable providers for interactions. As a result, the testbed records the UG of each interaction with each trust model used. In order to obtain an accurate result for performance comparisons between trust models, each one will be employed by a large number of consumer agents.

The experimental variables, that were adopted from [8] and used in all experiments, are presented in Table 1. The provider agents used are 100, 10 of which are good providers, 40 are ordinary, 5 are intermittent and 45 are bad providers.

Table 1. Experimental Variables

Number of simulation: 500 / Number of providers: 100	
Good providers	10
Ordinary providers	40
Intermittent providers	5
Bad providers	45

Fig. 1 shows particularly that the performance of the "No Trust" group, formed by selecting providers randomly without any trust evaluation, is, as expected, consistently the lowest. T-REX, being a centralized service, is able to gather ratings about all interactions in the system. This allows agents using it to achieve high performance right from the first interactions (the average UG per interaction of T-REX is 5,5).

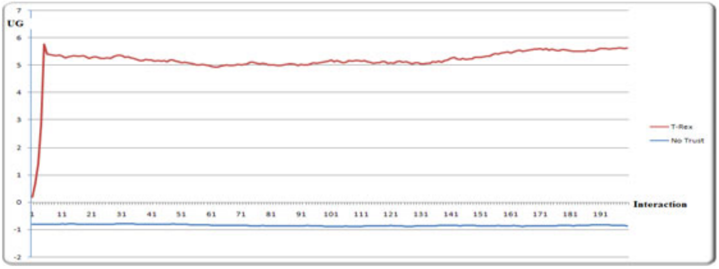


Fig. 1. Performance of T-REX and No Trust

Experiments with the same variables, presented in [8], have shown that the SPORAS [3] and Certified Reputation (CR) [8] are beneficial to consumer agents, helping them obtain significantly high UG. This means that the tested trust models can learn about the provider's population and allow their agents to select profitable providers for interactions. In contrast, since each provider only shows a small number of ratings to agents using CR, they spend the first few interactions learning about their environment.

Comparably, agents using T-REX maintain a slightly higher stable performance than the ones using SPORAS and CR (the average UG per interaction of SPORAS and CR are 4.65 and 5.48, respectively while for T-REX it is 5,5). This is because T-REX is a centralized model and is able to gather much more information than decentralized models and uses reputation in a fully dynamic and flexible way.

5 Use Case

This section presents a software-trading use-case paradigm. The scenario aims at demonstrating the overall functionality of the trust model and more specifically the usability of the agent Trustor in the Multi-Agent System and its ability to easily rate each agent in the system in a totally dynamic and flexible way.

5.1 Scenario Overview

A MAS is formed by two independent groups, represented by intelligent agents: (a) the first group consists of Customer agents, that are potential customers who wish to buy a specific software based on their preferences, and (b) the second group consists of Seller agents, that are potential sellers who wish to sell their software. Finally, (c) Trustor is the independent, certified, third-party agent-based service that provides the described T-REX mechanism.

Initially, the customer finds a seller, by asking the third-party agent Trustor for the level of the sellers' effectiveness in their previous transactions in the environment (step 1). This is actually achieved by providing its performance criteria weights via a *REQUEST* ACL message. Trustor uses the provided weights, calculates the final reputation value for each of the requested sellers and, then, returns it via an *INFORM* ACL message (step 2). The Customer selects the most trusted seller and transacts with it, in order to get the proper software (step 3). Then, the seller replies sending back the requested software (Step 4) and the customer rates its response time and efficiency and reports its rating to Trustor via an *INFORM* ACL message (step 5).

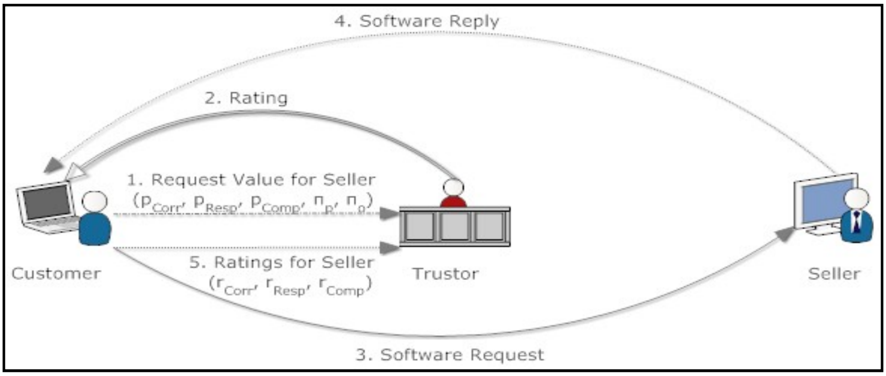

4. Software Reply
2. Rating
1. Request Value for Seller
$(p_{Corr}, p_{Resp}, p_{Comp}, n_R, n_p)$
Customer
5. Ratings for Seller
$(r_{Corr}, r_{Resp}, r_{Comp})$
Trustor
Seller
3. Software Request

Fig. 2. Transaction Steps

According to these transaction steps, depicted in figure 2, we have developed four sellers with different efficiency characteristics and three customers. Each customer asks for specific software from each of the four sellers and after each transaction the customer sends an ACL *INFORM* message to the Trustor with its ratings according to the correctness of the information received, how fast the information was received and its level of completeness. Then, the Trustor stores these ratings to its repository. Practically, the overall process is depicted in figure 3, taken from JADE's sniffer agent. This instance presents the transactions performed by the first three agents and the efforts of Trustor to create its case history.

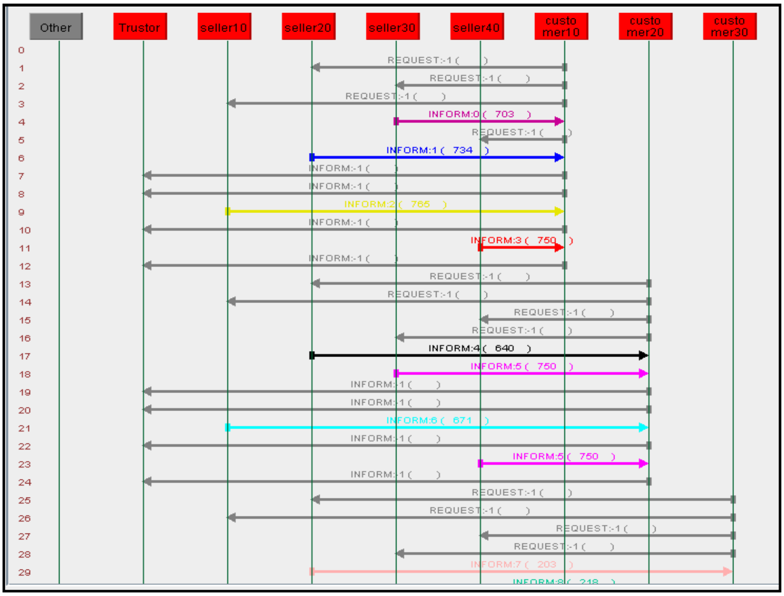


Fig. 3. The overall process – sniffer's snap-shot

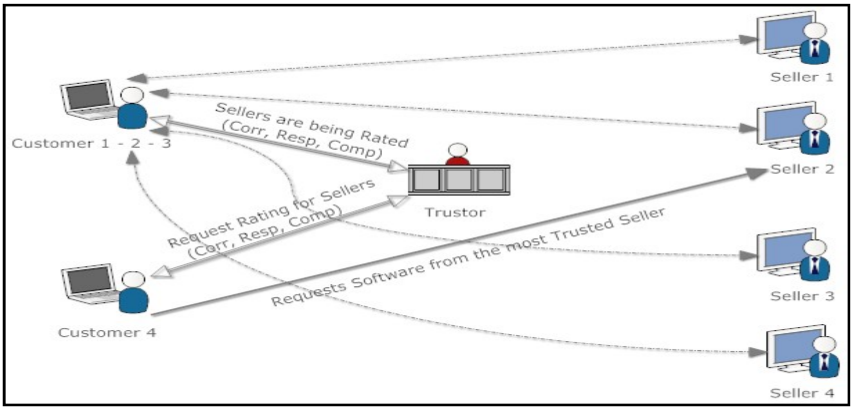

Seller 1
Customer 1 - 2 - 3
Sellers are being Rated (Corr, Resp, Comp)
Seller 2
Trustor
Request Rating for Sellers (Corr, Resp, Comp)
Requests Software from the most Trusted Seller
Seller 3
Customer 4
Seller 4

Fig. 4. Scenario Steps

Later, when a new customer agent enters the multi-agent system and requests from Trustor the reputation values for each seller acting in the environment (providing its personal performance criteria weights), the Trustor responds via an ACL *INFORM* message containing the necessary information. This process is depicted in the figure 4. In this case, the most reliable agent is Seller 2. Note that Trustor has already observed a number of transactions among the agents acting in this environment and thus has made a sufficient case history (repository records). This allows supporting either a witness system or a totally personal-based system.

To sum up, Trustor provides the implemented T-REX model, guaranteeing the transactions among agents. Finally, note that the reputation value of an agent is not fixed but dynamic, depending on the weights provided by another interested agent. Hence, a good agent according to an agent's preferences may be bad for another.

6 Related Work

Trust and reputation are key ingredients to most multi-agent systems and as a result many different metrics have been proposed [6][7]. SPORAS [3] is one of the most notable of these models. In this model, each agent rates its partner after an interaction and reports its ratings to the centralized SPORAS repository. The received ratings are then used to update the global reputation values of the rated agents. In SPORAS new agents start with a minimum value of reputation and they build it up during their activity on the MAS. After each transaction the values of reputation are updated according to the feedback provided by other agents that are involved. Both in T-REX and SPORAS ratings are discounted over time so that the most recent ratings are more weighted in an agent's reputation evaluation. Moreover, both models use a learning formula for the updating process so that the reputation value can closely reflect an agent's performance, at any time. However, SPORAS has limitations and as a result it is not as dynamic and flexible as T-REX. Furthermore, in SPORAS newcomers are not supposed to be reliable and the other agents do not trust them easily. On the other hand, our approach overcame the problem by evaluating new agents with a neutral rating value.

Jurca and Faltings [9] is a reputation model, where agents are incentivized to report truthfully about their interactions' results. There are a set of broker agents that are buying and aggregating reports from other agents and then they are selling the needed information to them. Although these broker agents are distributed in the system, each of them collects the reputation values centrally as T-REX. However, reputation values are limited to the values 0 and 1 (0 for cheating agents and 1 for reliable), which can be a problem if an agent is more reliable than another, but the other is not a cheating agent. Conversely, in the T-REX model there are continuous values in the range of [-1, 1], where -1 is the worst value for the most unreliable agents and 1 is the best for the most trustworthy agents. Moreover, despite T-REX, Jurca and Faltings model is a static parametric model and these parameters are not adjusted dynamically to adapt environment changes.

In the REGRET model [10], each agent evaluates the reputation of others after every interaction and records its ratings in a local database. Each rating is weighed according to how recent it is. Similarly to T-REX, in REGRET more recent ratings are more weighed than others that are less recent. However, REGRET is a completely decentralized reputation model. This model cannot deal with situations where the agent is changing behavior in the MAS, and is being transformed from a reliable, trustworthy agent to an unreliable one. On the other hand, T-REX overcame this problem as discussed above.

Certified Reputation [8] is a decentralized reputation model involving each agent keeping a set of references given to it from other agents. In this model each agent is asked to give certified ratings of its performance after every transaction. The agent then chooses the highest rating and stores them as references. Any other agent can then ask for the stored references and calculate the agent's certified reputation. This model overcame the problem of initial reliability in a similar way with T-REX. Most previous trust metrics did not perform reliably until a large enough number of interactions had built up, but certified reputation has been proved to perform reliably from the beginning of a simulation. However, this model, opposed to our approach, is designed to determine the access rights of agents, rather than to determine the expected performance of them.

7 Conclusions and Future Work

This paper presented T-REX, a hybrid agent trust model based on both witness reputation and personal experience. It is a dynamic and flexible model that overcomes the limitations of the known approaches, based solely either on witness reputation or personal experience. Furthermore, it provides a reliable centralized mechanism that can be adopted in any multi-agent system in the semantic web. In addition, among others, T-REX has low bandwidth and run-time computational cost and even low client storage cost. This paper also provided an evaluation of the model's performance and a comparison to two other popular models, SPORAS and Certified Reputation. Finally, a use case was presented that illustrated the functionality of the model. As for future directions, it would be interesting to verify our model's performance compared to more reputation models from the literature and use it in real-world e-commerce applications, combining it also with Semantic Web metadata for trust [11][12].

References

1. Hendler, J.: Agents and the Semantic Web. IEEE Intelligent Systems 16(2), 30–37 (2001)
2. Ramchurn, S.D., Huynh, D., Jennings, N.R.: Trust in multi-agent systems. Knowledge Engineering Review 19(1), 1–25 (2004)
3. Zacharia, G., Maes, P.: Trust management through reputation mechanisms. Applied Artificial Intelligence 14(9), 881–908 (2000)
4. Java Agent Development Framework; JADE, `http://jade.tilab.com`
5. Huynh, D., Jennings, N.R., Shadbolt, N.R.: An integrated trust and reputation model for open multi-agent systems. Journal of AAMAS (2006)
6. Grandison, T., Sloman, M.: A survey of trust in internet applications. IEEE Comm. Surveys & tutorials 3(4) (2000)
7. Maximilien, E.M., Singh, M.P.: Reputation and endorsement for web services. ACM SIGEcon Exchanges 3(1), 24–31 (2002)
8. Huynh, D., Jennings, N.R., Shadbolt, N.R.: Certified reputation: How an agent can trust a stranger. In: AAMAS '06: Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems, Hokkaido, Japan (2006)
9. Jurca, R., Faltings, B.: Towards incentive-compatible reputation management. In: Falcone, R., Barber, S.K., Korba, L., Singh, M.P. (eds.) AAMAS 2002. LNCS (LNAI), vol. 2631, pp. 138–147. Springer, Heidelberg (2003)
10. Sabater, J., Sierra, C.: REGRET: A reputation model for gregarious societies. In: Fourth Workshop on Deception Fraud and Trust in Agent Societies, Canada, pp. 61–70 (2001)
11. Ceravolo, P., Damiani, E., Viviani, M.: Adding a Trust Layer to Semantic Web Metadata. In: Soft Computing for Information Retrieval on the Web, vol. 197, pp. 87–104. Springer, Heidelberg (2006)
12. Ceravolo, P., Damiani, E., Viviani, M.: Bottom-Up Extraction and Trust-Based Refinement of Ontology Metadata. IEEE Transactions on Knowledge and Data Engineering 19(2), 149–163 (2007)

QoS Contract Formation and Evolution

Vasilios Andrikopoulos[1], Mariagrazia Fugini[2], Mike P. Papazoglou[1],
Michael Parkin[1], Barbara Pernici[2], and Seyed Hossein Siadat[2]

[1] European Research Institute in Service Science (ERISS),
Tilburg University, The Netherlands
[2] Politecnico di Milano, Italy

Abstract. This paper is concerned with the issues of QoS contract formation between service providers and consumers and how either party can evolve independently from each other without violating the agreed contract. We show how a QoS contract can be generated using a subtyping relation on the quality dimensions and value ranges. For that purpose, we use Allen's Interval Algebra (AIA). We also define both strict and relaxed constraints for different dimensions in order to deal with what constitutes acceptable change to different quality dimensions. In particular, we define assertion compatibility as a sufficient condition for ensuring the compatibility of provider and consumer with respect to an existing contract.

Keywords: QoS, service contracts, service evolution, quality dimensions, type theory.

1 Introduction

A Service Level Agreement (or SLA) is a mutually-agreed contract between a service provider and consumer that can describe how, amongst other things, the non-functional properties of a service should be supplied, monitored and charged. Typical non-functional properties include availability, throughput and response time, and they are often collectively referred to as *quality dimensions* to indicate they may apply to any aspect of the service, from the system infrastructure level to the business application layer.[1]

An agreement on the quality dimensions and their values (i.e., an SLA or contract) goes through several stages, from the initial service advertisement or query to the final evaluation of the service's performance against the agreed SLA. Fig. 1 shows a typical SLA lifecycle. This paper is concerned with the QoS contract formation stage of the SLA's lifecycle, shown in detail in Fig. 1. This abstract contract formation process from [6] shows how this stage of the SLA's lifecycle is made up of three parts: the 'matchmaking' phase filters the providers of services according to the consumer's requirements from which services are selected and (in the final phase) the SLA is configured with the quality dimensions that should be monitored and enforced.

[1] Quality dimensions have also been called *SLA terms* [2].

F. Buccafurri and G. Semeraro (Eds.): EC-Web 2010, LNBIP 61, pp. 119–130, 2010.

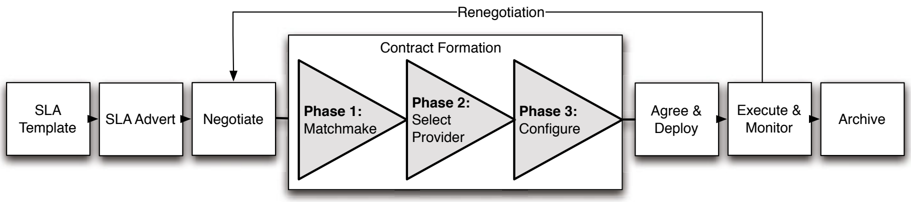


Fig. 1. An SLA Lifecycle [4] showing the QoS Contract Formation Process [6]

Much work exists on the negotiation of SLAs and how they can be monitored; however there is a gap in current research in how a contract is *formed* from the service provider's advertised capabilities and the service consumer's desired service capabilities. We define *QoS contract formation* as the process of determining if the desired service quality capabilities can be met from the advertised capabilities and this paper provides a formal model for the QoS contract formation phase of the SLA lifecycle.

As also shown in Fig. 1, the renegotiation of an agreement can occur if either party changes its circumstances during the execution of the task associated with the SLA. This requires the contract formation process to be repeated, which could be time consuming if the provider and consumer do not re-agree. For *shallow* [3] (i.e. non-breaking to the consumers) changes in the SLA, we propose a method of allowing both the provider and consumer to evolve without requiring negotiation.

Underlying this work are the following basic assumptions: *a)* the contract is being formed between two parties, the service provider and the service consumer and, *b)* the set of quality dimensions, their semantics and domains or ranges are understood and agreed (explicitly or implicitly) by both parties in the agreement. An explicit agreement on the quality dimensions can be achieved using a *meta-negotiation* [5] step that allows both parties to negotiate about what can be agreed, whilst an implicit agreement is in place when the service is part of a service network that shares a common, well-known model for quality dimensions such as the WSLA specification [7]. In the paper we focus on defining how providers capabilities and consumer requests are allowed to vary without the need of changing the contract.

Table 1. Working example: Service Provider Capabilities & Consumer Requirements

ACME Provides:	Availability Response Time Price	Between 80–95% of the time. Between 2–5 seconds. €1/invocation.
	Availability Response Time Price	Between 70–95% of the time. Between 3–5 seconds. €0.80/invocation.
Consumer Requires:	Availability Response Time Price	Between 80–90% of the time. Between 3–5 seconds. Up to (and including) €1/invocation.

To illustrate our approach we use an example that runs throughout the paper, as summarized in Table 1 which shows two SLA adverts from a service provider called ACME. The first describes a service profile that has high availability and low response time but a high cost per invocation. The second describes a service with lower availability and response time but with a lower cost-per-use. The consumer's requirements are also shown. As can be seen, the consumer requires high availability but low response time. The challenge in this example is how to find an acceptable agreement to satisfy both parties.

We start by briefly reviewing related work (Section 2). In Section 3 we show how to formalize the SLA adverts and requirements and in Section 4 how an SLA is formed based on them; in Section 5 we show how either party can evolve while respecting an existing SLA before discussing an initial validation of the approach (Section 6) and concluding the paper in Section 7.

2 Related Work

The current standard for agreeing SLAs between Web services [2] describes a "protocol and agreement template structure to facilitate the discovery of compatible agreement parties" and creating an SLA but does not specify a method of matching the service provider's capabilities with the customer requirements nor how to form the agreement. As a result, there is much previous work on service selection techniques policies based on, for example, reputation-systems [8], semantic annotations [11] and multi-attribute utility theory [10]. The systems built to use these policies often use an SLA language to describe their services and contracts, such as [12] and [7]. However, in the work we have reviewed, service selection policies and languages fail to define what the process of QoS contract formation actually entails. This paper proposes a formal model for QoS contract formation in Section 4.

Much of the work on service selection and SLA negotiation, agreement and management does not consider the evolution of the involved parties as this paper proposes in Section 5. In current models and techniques if either the service provider or consumer wish to change their side of the contract (i.e., what is being provided or what is required from the service) the current contract must be renegotiated which brings additional overheads to both the service providers and consumers. The goal of this paper is to define cases in which the evolution of providers and/or consumers does not require a renegotiation of the contract.

3 Quality Dimensions and Service Descriptions

We define set $\mathcal{D}$ to contain the quality dimensions (such as availability, execution time, price or throughput) identified and agreed *a priori* by the service provider and consumer. Each quality dimension has a domain and range; e.g., availability is a probability usually expressed as a percentage in the range 0-100% and execution time is in the domain of real numbers in the range $0..+\infty$. In this paper, we only consider *ordinal* quality dimensions, i.e., quality dimensions whose values can be ordered. An ordered quality dimension d can be considered *monotonic* (denoted by d^+) or *antitonic* (d^-); monotonicity indicates that values closer to the upper bound of the range are considered "better", whilst with antitonic dimensions values closer to the lower bound are considered better. A *parameter* m associates a quality dimension to a value range in the following way:

Definition 1 (Parameter). *Parameter* $m \in \mathcal{M}$ *where* $\mathcal{M} : \mathcal{D} \times \mathcal{V}$. $\mathcal{D}$ *is the set of quality dimensions, whilst* $\mathcal{V}$ *is the set of ranges for all quality dimensions* $\mathcal{D}$. *A dimension and its range are therefore the tuple* $m := (d, v), d \in \mathcal{D}, v \in \mathcal{V}$.

Using this definition we can define multiple value ranges for the same quality dimension. For example, if we consider the single monotonic quality dimension of `Availability` (which is monotonic since higher values of availability are considered "better") then $d_1^+ = \texttt{Availability}$. If the range of availability is between 80–95% then $v_1 = [.8, .95]$ and $m_{p1,1} = (d_1, v_1)$, whereas if it is in the range 70–95% then we can define $v_2 = [.7, .95]$ and $m_{p1,2} = (d_1, v_2)$, etc.

Table 2 has three parts: the first is the set of parameters denoting the intention of the service provider to provide a group of particular quality dimensions within the advertised ranges. To signify that the service provider requires the service's consumer to provide dimensions for parameters — as in the case of `Price` (d_3^-) — we use the *complementary* operator, $\overline{m} = (d, v)$, to show that dimension d is required to be within the range of v[2]. The pricing requirement of the provider's advertisement can then be formalized as $\overline{m_{p3,1}} = (d_3, [1, 1])$ and $\overline{m_{p3,2}} = (d_3, [.8, .8])$.

We can reverse this logic and apply it to the consumer's requirements to formalize the consumer service descriptions, as shown in the second part of Table 2: the consumer's required availability is between 80–90% in the example and can be expressed as $\overline{m_{c1,1}} = (d_1, [.8, .9])$; the required response time is between 3–5 seconds and can be written as $\overline{m_{c2,1}} = (d_2, [3, 5])$; the intention to pay up to €1/invocation is written as $m_{c3,1} = (d_3, [0, 1])$.

Combining parameters into *assertions* allows providers and consumers to express statements of intention or expectation that couple parameters. For example, we can combine parameters $m_{p1,1}$, $m_{p2,1}$ and $\overline{m_{p3,1}}$ into assertion $q_{p1} = (m_{p1,1}, m_{p2,1}, \overline{m_{p3,1}})$, which is a conjunction of dimensions d_1, d_2 and d_3 and must belong to the ranges defined by $m_{p1,1}$, $m_{p2,1}$ and $\overline{m_{p3,1}}$, respectively. Each assertion can either be true or false given specific values for each dimension at run time, which is necessary for monitoring the QoS characteristics of a service. We can formally define an assertion as:

Definition 2 (Assertion). *An assertion* $q \in \mathcal{Q}, \mathcal{Q} : \mathcal{M}^n$ *is defined as the tuple* $q := (m_i), 1 \leq i \leq n$ *i.e., an ordered set of parameters that should be interpreted as a conjunction.*

Assertions can be combined in a similar fashion to form a *profile*:

Definition 3 (Profile). *A profile* $l \in \mathcal{L}, \mathcal{L} : \mathcal{Q}^k$ *is defined as the tuple* $l := (q_j), 1 \leq j \leq k$ *i.e. an ordered set of assertions (interpreted as a disjunction).*

[2] It can also be seen that $\overline{\overline{m}} = m$.

Thus, by combining assertion q_{p1} with the assertion $q_{p2} = (m_{p1,2}, m_{p2,2}, \overline{m_{p3,2}})$ for the second set of quality dimensions of the service provider in Table 1, we can define the profile $l_p = (q_{p1}, q_{p2})$ which states that for availability between 80–95% and response time between 2–5 seconds, the price of the service per invocation will be €1; with availability between 70–95% and response time between 3–5 seconds the price per invocation will be €0.80. This example is summarized in Table 2 with the remainder of the formalization.

Table 2. Working example: formal specifications

Provider $\mathcal{P}$	$l_p = (q_{p1}, q_{p2})$
	$q_{p1} = (m_{p1,1}, m_{p2,1}, \overline{m_{p3,1}})$ $m_{p1,1} = (d_1, [0.8, 0.95]), m_{p2,1} = (d_2, [2, 5]), \overline{m_{p3,1}} = (d_3, [1, 1])$
	$q_{p2} = (m_{p1,2}, m_{p2,2}, \overline{m_{p3,2}})$ $m_{p1,2} = (d_1, [0.7, 0.95]), m_{p2,2} = (d_2, [3, 5]), \overline{m_{p3,2}} = (d_3, [0.8, 0.8])$
Consumer $\mathcal{C}$	$l_c = (q_{c1})$
	$q_{c1} = (\overline{m_{c1,1}}, \overline{m_{c2,1}}, m_{c3,1})$ $\overline{m_{c1,1}} = (d_1, [0.8, 0.9]), \ \overline{m_{c2,1}} = (d_2, [3, 5]), \ m_{c3,1} = (d_3, [0, 1])$
Dimensions $\mathcal{D}$	$d_1^+ = \texttt{Availability}, \ d_2^- = \texttt{ResponseTime}, \ d_3^- = \texttt{Price}$

4 QoS Contract Formation

Having defined quality parameters, assertions and profiles we can now explain how these are combined to form a contract. We use the theory presented in [3] as the foundation for this work, which defines a contract as the product of applying a binding function, ϑ, to the advertised and desired capabilities of the provider $\mathcal{P}$ and consumer $\mathcal{C}$ respectively.

The function ϑ builds on a subtyping relation in order to identify and provide a pair-wise matching between the elements of the service descriptions. By using subtyping for matchmaking we incorporate into one relation the checking for both structural and semantic similarities between the elements of the provider $\mathcal{P}$ and consumer $\mathcal{C}$. Furthermore, as will we show in the following, it not only allows us to configure the contract between $\mathcal{P}$ and $\mathcal{C}$ while going through the matchmaking process, but also allows for the seamless evolution of providers and consumers without affecting the contract (under certain conditions).

4.1 Subtyping

In order to build the binding function ϑ necessary to form the contract, we first define the *subtyping* relation for the elements of the QoS description. In particular, we define:

Definition 4 (Parameter Subtyping). *For $m = (d, v)$, $m' = (d', v')$ and a subtyping relation $<:$ it holds:*

$$m <: m' \Leftrightarrow d <: d' \land v <: v' \tag{1}$$

Note: For this definition and for the discussion that follows it will be assumed that p stands for either m or $\overline{m}$.

Parameter subtyping requires the further definition of subtyping for quality dimensions and value ranges. For the former we assume that only the same quality dimension can be compared, i.e.,:

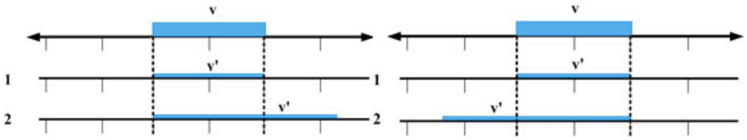
$$d <: d' \sim d \equiv d' \tag{2}$$

Comparing quality dimension types can be enhanced by adding a hierarchy of quality dimensions that allow us to navigate the IS-A relationships between dimensions and discern between more or less generic dimensions (i.e., super- or sub-types). An example of such a taxonomy is the quality model produced by the S-Cube Network of Excellence; their model [9] has a deep class hierarchy that is suitable for this approach. The dimension of `Latency` (the time passed from the arrival of the service request until the end of its execution/service) for example is a type of `ResponseTime` and therefore parameters defined on the two dimensions are comparable. A mapping between the different domain and ranges for the value ranges of each dimension may be required for that purpose. We intend to further investigate the impact of such reasoning to the QoS contract formation in the immediate future.

To express the subtyping of the two value ranges as intervals we use Allen's interval algebra [1] and in particular the *starts* s, *finishes* f and *is equal to* = relations, which are used as follows:

$$v <: v' \Leftrightarrow \begin{cases} v\{=,\mathtt{s}\}v' \text{ for monotonic dimensions} \\ v\{=,\mathtt{f}\}v' \text{ for antitonic dimensions} \end{cases} \tag{3}$$

Equation 3 defines a more generic value range (i.e., super-type) to be a value range that totally contains another (value range) and also contains values from the increasing side of the order of the dimension. Value ranges are therefore generalized by extending their maximum allowed value in the case of monotonic dimensions and decreasing their minimum value in that of antitonic dimensions, as shown in Fig. 2. The result in both cases is to accept/expect a greater range of values in the parameter, while incorporating the values already defined. By using Equations 2 and 3 for Definition 4, and for the parameters in Table 2 it therefore holds that:



(a) Monotonic Dimensions (b) Antitonic Dimensions

Fig. 2. Positioning of value ranges in Equation 3

$$\overline{m_{c1,1}} <: m_{p1,1} \;\wedge\; \overline{m_{c2,1}} <: m_{p2,1} \;\wedge\; \overline{m_{p3,1}} <: m_{c3,1}$$
$$\overline{m_{c1,1}} \not<: m_{p1,2} \;\wedge\; m_{c2,2} <: \overline{m_{p2,1}} \;\wedge\; \overline{m_{p3,2}} \not<: m_{c3,1}$$

This information can be used in formulating the contract between providers and consumers as follows.

4.2 Matching and Mapping Functions

Having defined the subtype relation for quality dimensions and value ranges, we can define the binding function ϑ used for formulating contracts:

Definition 5 (Binding function). *Binding function $\vartheta(q_p, q_c)$ defined over assertions $q_p = (x_1, \ldots, x_n)$ and $q_c = (y_1, \ldots, y_n)$, $q_p \in \mathcal{P}$ and $q_c \in \mathcal{C}$ is:*

$$\vartheta(q_p, q_c) = \{z = (z_1, \ldots, z_n)/\ 1 \leq i \leq n, \begin{cases} x_i <: z_i <: y_i, x_i \in \mathcal{P}^{req}, y_i \in \mathcal{C}^{pro} \\ y_i <: z_i <: x_i, x_i \in \mathcal{P}^{pro}, y_i \in \mathcal{C}^{req} \end{cases} \}$$

where $\mathcal{P}^{pro}, \mathcal{C}^{pro}$ are the subsets representing the output-type parameters in the provider and consumer description respectively (using provided dimensions d_i), and $\mathcal{P}^{req}, \mathcal{C}^{req}$ the input-type parameters (using required dimensions $\overline{d_i}$) [3].

A contract $\mathcal{R}$ between provider $\mathcal{P}$ and consumer $\mathcal{C}$ is the triplet $< \mathcal{P}, \mathcal{C}, \Theta >$ where $\Theta := \{\vartheta(q_p, q_c)/q_p \in \mathcal{P}, q_c \in \mathcal{C}\}$. Following the working example, and from the subtyping relations between the quality dimensions and value ranges discussed in the previous section, we can see it is possible to formulate a contract $\mathcal{R}$ using q_{p1} from the provider and q_{c1} from the consumer as shown in Table 3. Since $\overline{\overline{s}} = s$, it does not matter which perspective we use for the contract terms; for example, in Table 3 we chose to express them using the provider's perspective but could have used the consumer's and achieved the same result.

The binding function, ϑ, combines the steps of matchmaking and contract configuration found in the QoS contract formation phase. It provides the means for checking whether the service consumer's desired capabilities are met by the service provider and, at the same time, generates an intermediary service description based on the possible outcomes. However, a complication in the configuration of the contract is the ϑ value selection policy: from the definition of ϑ, contract terms $(z_1, \ldots, z_n)$ may be configured using different values for each parameter pair (x_i, y_i) as long as they comply to the conditions in Definition 5.

Table 3. Working example: contract example

Provider $\mathcal{P}$	**Contract $\mathcal{R}$**	**Consumer $\mathcal{C}$**
$q_{p1} = (m_{p1,1}, m_{p2,1}, \overline{m_{p3,1}})$	$q_{r1} = (m_{r1,1}, m_{r2,1}, \overline{m_{r3,1}})$	$q_{c1} = (\overline{m_{c1,1}}, \overline{m_{c2,1}}, m_{c3,1})$
$m_{p1,1} = (d_1, [.8, .95])$	$m_{r1,1} = (d_1, [.8, .9])$	$\overline{m_{c1,1}} = (d_1, [.8, .9])$
$m_{p2,1} = (d_2, [2, 5])$	$m_{r2,1} = (d_2, [3, 5])$	$\overline{m_{c2,1}} = (d_2, [3, 5])$
$\overline{m_{p3,1}} = (d_3, [1, 1])$	$\overline{m_{r3,1}} = (d_3, [1, 1])$	$m_{c3,1} = (d_3, [0, 1])$

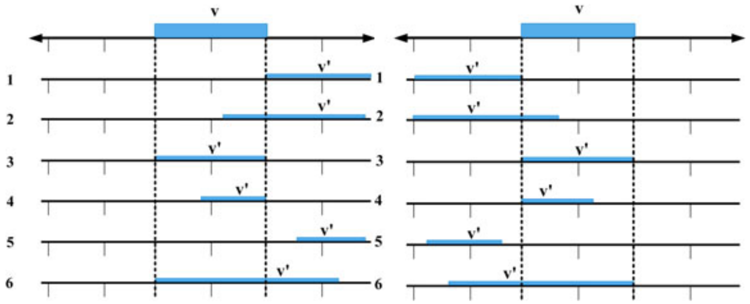
4.3 Strict and Relaxed Constraints

As mentioned above, the definition of subtyping for value ranges can be too strict, as the following example demonstrates: assume that in Table 2 we had $q'_{p1} = (m'_{p1,1}, m_{p2,1}, \overline{m_{p3,1}})$ where $m'_{p1,1} = (d_1^+, [.85, .99])$, i.e., the service provider offers better availability than the consumer requires. Nevertheless, it is not possible to formulate a contract between $\mathcal{P}'$ and $\mathcal{C}$ since according to Equation 3 it holds that $\overline{m_{c1,1}} \not<: m'_{p1,1}$ and therefore $\nexists z / \overline{m_{c1,1}} <: z <: m'_{p1,1}$.

In order to accommodate cases like these we relax the definition of the value range subtyping as an extension of the existing value range and we include the relations *meets* m, *overlaps* o and *takes place before* $<$ (and their inversions $\mathsf{mi}, \mathsf{oi}, >$, together with the inversions of the starts and finishes relations si, fi) from interval algebra as follows:

$$v <: v' \Leftrightarrow \begin{cases} v\{=,<,\mathsf{s},\mathsf{fi},\mathsf{m},\mathsf{o}\}v' & \text{for monotonic dimensions} \\ v\{=,>,\mathsf{f},\mathsf{si},\mathsf{mi},\mathsf{oi}\}v' & \text{for antitonic dimensions} \end{cases} \tag{4}$$

The relevant positioning of v and v' as defined by Equation 4 is shown in Fig. 3. The addition of these relations expands on the nature of the monotonic and antitonic dimensions and allows for additional flexibility, converting the *strict constraints* in the definition of value range subtyping into *relaxed constraints*. The basic difference of between strict constraints (Equation 3) and relaxed constraints (Equation 4) is in the assumption of what constitutes acceptable behavior by each party in the contract. Equation 3 only allows for extending the acceptable boundaries of a dimension towards "better" values – but in order to be safe it always includes the original value range for the dimension.



(a) Monotonic Dimensions (b) Antitonic Dimensions

Fig. 3. Positioning of value ranges in Equation 4

Relaxed constraints remove this restriction to allow all types of "better" values to be included in the acceptable boundaries. This allows, for example, accepting value ranges that do not even partially overlap with the original value range, provided that they are signifying a more favorable value range as for example in case (5) of Fig. 3a and Fig. 3b. This assumes that the other party in the QoS contract formation process is willing to accept such a value range — which is not necessarily true for all cases.

As an example in practice, take the case of a time-critical service-based application, optimized for a response time of between 3–5 seconds. If the system starts experiencing response times between 1–2 seconds this may interfere with its ability to deal with the throughput and, as a result, it may require adapting. Since we can not know what the cost of adapting the application is, and whether or not that is acceptable for the application owner, we may have unintentionally broken the capability of the application to consume the service by improving the service performance. This potentially problematic case appears even if we use only strict constraints; using relaxed constraints increases the severity of the problem, if it occurs. Nevertheless, it is our belief that the explicit benefits of allowing for the evolution of parties in the conversation outweighs the potential disadvantages in applying it to (over-)optimized applications.

To return to the opening example in this section: from Equation 4 it holds that $\overline{m_{c1,1}} <: m'_{p1,1}$ (since they overlap) and therefore it is possible to generate a contract $\mathcal{R}'$ between $\mathcal{P}'$ and $\mathcal{C}$. As we will show in the following section, it is not even unnecessary to formulate a new contract $\mathcal{R}'$ after the evolution of $\mathcal{P}$ to $\mathcal{P}'$ in the case that a contract $\mathcal{R}$ already exists between $\mathcal{P}$ and $\mathcal{C}$: the fact that $m_{p1,1}, <: m'_{p1,1}$ suffices for signifying the compatibility of $\mathcal{P}$ and $\mathcal{P}'$ with respect to the existing contract $\mathcal{R}$. $\mathcal{P}'$ can then use $\mathcal{R}$ for its interactions with $\mathcal{C}$ – without affecting or having to notify the service consumers.

5 Evolution of Providers and Consumers

Since service providers and their consumers may change during the lifetime of the service we need a method of ensuring that changes in either party in the contract do not break its obligations. This means that the *compatibility* between new and old versions of the providers and consumers needs to be assured with respect to the contract. For this purpose we start by defining *assertion compatibility* as a sufficient condition for ensuring the compatibility of provider and consumer versions with respect to an existing contract and show how it can be used:

Definition 6 (Assertion Compatibility). *Assertions* $q = (m_1, \ldots, m_N)$ *and* $q' = (m'_1, \ldots, m'_N)$ *are called compatible and we write* $q <_c q'$ *if it holds that* $q <_c q' \Leftrightarrow m_i <: m'_i, m_i \in \mathcal{S}^{pro}, m'_i \in \mathcal{S}'^{pro} \wedge m'_i <: m_i, m_i \in \mathcal{S}^{req}, m'_i \in \mathcal{S}'^{req},\ 1 \leq i \leq N$

From the definition of the binding function ϑ and using only compatible assertions from Definition 6 we can show that new versions of a provider $\mathcal{P}$ or a consumer $\mathcal{C}$ are compatible to an existing contract $\mathcal{R}$ between them, or in the notation of [3], that $\mathcal{P} \mapsto_R \mathcal{P}'$ and $\mathcal{C} \mapsto_R \mathcal{C}'$.

In order to demonstrate this we start with the provider's profile and assume, without loss of generality, it contains a single different assertion between the two versions, e.g., $q_p \in \mathcal{P}, q_p = (x_1, x_2, \ldots, x_n)$ and $q'_p \in \mathcal{P}', q'_p = (x'_1, x_2, \ldots, x_n)$

with $x_1 \in \mathcal{P}^{pro}$ and $x'_1 \in \mathcal{P}'^{pro}$, respectively. From the definition of assertion compatibility it holds that $q_p <_c q'_p$ iff $x_1 <: x'_1$, and from the definition of the binding function we find $\exists z_1 \in \Theta, y_1 <: z_1 <: x_1$. By combining these two statements we see $y_1 <: z_1 <: x_1 <: x'_1$ or equivalently, $y_1 <: z_1 <: x'_1$ and therefore $\mathcal{P} \mapsto_R \mathcal{P}'$. Using a similar construction we can show that all assertion compatible versions according to Definition 6 are also compatible with respect to an existing contract.

Table 4. Evolution of interacting parties (sample)

Party	**Change**	**Compatible?** **(Strict/Relax)**	**Change**	**Compatible?** **(Strict/Relax)**
$\mathcal{P}$	$m'_{p1,1} = (d_1^+, [.8, .99])$	Yes/Yes	$m'_{p2,1} = (d_2^-, [1, 5])$	Yes/Yes
$\mathcal{P}$	$m'_{p1,1} = (d_1^+, [.7, .9])$	No/No	$m'_{p2,1} = (d_2^-, [3, 6])$	No/No
$\mathcal{P}$	$m'_{p1,1} = (d_1^+, [.85, .99])$	No/Yes	$m'_{p2,1} = (d_2^-, [1, 4])$	No/Yes
$\mathcal{C}$	$\overline{m_{c1,1}}' = (d_1^+, [.8, .99])$	No/No	$\overline{m_{c2,1}}' = (d_2^-, [1, 5])$	No/No
$\mathcal{C}$	$\overline{m_{c1,1}}' = (d_1^+, [.7, .9])$	No/Yes	$\overline{m_{c2,1}}' = (d_2^-, [3, 6])$	No/Yes
$\mathcal{C}$	$\overline{m_{c1,1}}' = (d_1^+, [.8, .85])$	Yes/Yes	$\overline{m_{c2,1}}' = (d_2^-, [4, 5])$	Yes/Yes

Table 4 shows the result of checking for assertion compatibility on a set of possible changes. For example, changing the availability of the service provider from $[.8, .95]$ to $[.8, .99]$ does not change the contract $\mathcal{R}$ irrespective of whether we are using strict or relaxed constraints, since $m_{p1,1} <: m'_{p1,1} \Rightarrow q_{p1} <_c q'_{p1}$ and therefore $\mathcal{P} \mapsto_R \mathcal{P}'$ as before. For $m'_{p1,1} = (d_1^+, [.7, .9])$ on the other hand, $m_{p1,1} \not<: m'_{p1,1}$ and $m'_{p1,1} \not<: m_{p1,1}$ (under either strict or relaxed constraints) and thus $q_{p1} \not<_c q'_{p1} \Rightarrow \mathcal{P} \not\mapsto_R \mathcal{P}'$, i.e., it is an incompatible change to the service provider.

The situation is inversed for the same changes if they occur on the service consumer side: requiring availability between $[.8, .99]$ when it was originally agreed that it will be $[.8, .9]$ is in violation of the formulated contract $\mathcal{R}$ since $\mathcal{P}$ will not be able to provide this range of values. Requiring availability between $[.7, .9]$ may be acceptable using the relaxed constraint definition, assuming the consumer does not have a problem accepting better availability (as discussed above). A similar reasoning can be performed for the antitonic dimension of `ResponseTime` but with the symmetrical results due to the nature of the dimension.

The definition of assertion compatibility therefore provides service developers with the means to reason on the effect of a proposed change to its providers/consumers. It restricts the possible changes to a specific set ruled by the conditions of Definition 6 that can be checked in a straightforward manner. In that sense it limits the options in evolving a service or a service-based application but on the other hand it can guarantee the result of this process is not affecting the opposite party. In [3] it is further discussed how the set of compatible changes can be further extended by allowing the evolution of the contract itself (for structural contracts). A similar approach can be applied here for the same purposes.

6 Validation

We have implemented a system prototype for contract management that can evaluate the modification of provider and consumer according to the predefined contract model, existing policies and constraints used in the paper. The architecture of the system includes two main components, namely, the interfaces of the interacting parties and the contract broker. The provider interface is used for service publication while the requestor interface is used for service requests from the consumers. The contract broker is responsible for establishing and managing the contract. Readers may refer to [6] for a complete discussion of the contract broker.

```
<?xml version=”1.0” encoding=”utf-8”?>
<wsp:policy xmlns:wsp=”http://schemas.xmlsoap.org/ws/2004/09/policy”>
 <wsp:ExactlyOne>
    <wsp:availability minvalue=”80” maxvalue=”90” unit=”percent”/>
    <wsp:workTiming minvalue=”3” maxvalue=”5” unit=”second” />
    <wsp:price minvalue=”1” maxvalue=”1” unit=”euro” />
 </wsp:ExactlyOne>
</wsp:policy>
```

Fig. 4. WS-Policy document of the contract

Particularly, the system uses three main WS-Policy files in XML format for storing policies from provider, consumer and SLA contract. The information retrieved from the provider's file is used to evaluate the compatibility of new providers with the contract while the consumer's file is used to evaluate the compatibility of new consumers with the contract. The system prototype is implemented in C# , using .NET 3.5 and designed based on Windows Presentation Foundation (WPF). Fig. 4 illustrates the WS-Policy document of the contract for the working example applied in the paper.

7 Conclusions and Future Work

In this paper we built on previous work on QoS contract formation and evolution in order to develop an appropriate approach for QoS contracts. For that purpose we propose a flexible approach to deal with QoS parameters formulated by providers and consumers within given value ranges. An algebra for comparing different offered/required values ranges for various parameters and dimensions has been presented as the basis of this effort. A strict and a relaxed interpretation of constraints for QoS dimensions has been incorporated into the algebra and its impact on the QoS contract formation has been discussed. We have also shown how we can discriminate between the case of a variation in QoS that signals a deep change in provisioning service qualities (and which implies contract re-negotiation and possibly the need for adapting the interacting party) and the case where QoS variations can be treated as an evolution of the interacting parties. In particular, we have defined a set of compatibility assertions that allow us to accommodate providers' and consumers' contractual obligations despite changes to them — without the need to redefine the contract between them.

Furthermore, an initial validation phase based on the definition of QoS dimensions using WS-Policy specifications has been implemented. We are currently extending this prototype to include more specific WS-Policy elements, with a user-friendly interface and further logical reasoning on contract violations. In addition, in the future we intend to extend our approach by reasoning on the relations between dimensions and provide a comprehensive QoS contract formation and evolution life-cycle which would allow also for the evolution of the contract itself (as in [3]).

Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement 215483 (S-Cube).

References

1. Allen, J.F.: Maintaining Knowledge about Temporal Intervals. Communications of the ACM 26(11), 832–843 (1983)
2. Andrieux, A., et al.: Web Services Agreement Specification (WS-Agreement). Recommended standard, Open Grid Forum (March 2007)
3. Andrikopoulos, V., et al.: Evolving Services from a Contractual Perspective. In: van Eck, P., Gordijn, J., Wieringa, R. (eds.) CAiSE 2009. LNCS, vol. 5565, pp. 290–304. Springer, Heidelberg (2009)
4. Benbernou, S., et al.: A Survey on Service Quality Description. ACM Computing Surveys (2009) (submitted for publication)
5. Brandic, I., et al.: Towards a meta-negotiation architecture for SLA-aware Grid services. In: Proceedings of the International Workshop on Service-Oriented Engineering and Optimizations 2008 (December 2008)
6. Comuzzi, M., Pernici, B.: A Framework for QoS-Based Web Service Contracting. ACM Transactions on the Web 3(3) (June 2009)
7. Ludwig, H., et al.: Web Service Level Agreement (WSLA) Language Specification. Document WSLA-2003/01/28, IBM Corporation (2003)
8. Maximilien, E.M., Singh, M.P.: Toward Automatic Web Services Trust and Selection. In: Proceedings of the 2nd International Conference on Service Oriented Computing (ICSOC 2004), pp. 212–221 (2004)
9. S-Cube Partners. Quality Reference Model for Service-Based Applications. Deliverable CD-JRA-1.3.2, S-Cube Network of Excellence (March 2009)
10. Seo, Y.-J., et al.: A Study on Web Services Selection Method Based on the Negotiation Through Quality Broker: A MAUT-based Approach. In: Wu, Z., Chen, C., Guo, M., Bu, J. (eds.) ICESS 2004. LNCS, vol. 3605, pp. 65–73. Springer, Heidelberg (2005)
11. Sirin, E., et al.: Filtering and Selecting Semantic Web Services with Interactive Composition Techniques. IEEE Intelligent Systems 19(4), 42–49 (2004)
12. Skene, J., et al.: Precise Service Level Agreements. In: Proceedings of 26th International Conference on Software Engineering (ICSE), pp. 179–188 (2004)

Process Views to Support Compliance Management in Business Processes

David Schumm, Frank Leymann, and Alexander Streule

Institute of Architecture of Application Systems, University of Stuttgart,
Universitätsstraße 38, 70569 Stuttgart, Germany
{Schumm,Leymann,Streule}@iaas.uni-stuttgart.de

Abstract. Compliance has become an important driver in business process management, as it requires profound and traceable changes of the processes. Besides the increasing demand for security, privacy and trust, compliance also needs consistent integration and management of process structures related to compliance. We use the notion of compliance fragments to refer to such structures. In this paper, we discuss the challenges of managing compliance fragments in business processes. Extraction, integration, highlighting and hiding of compliance fragments represent the challenges we refer to. For extraction and hiding of compliance fragments we present an implementation for the process execution language BPEL, based on process view transformation concepts.

Keywords: Process View, Model Transformation, Compliance Fragment.

1 Introduction

From a high level perspective, business process management (BPM) basically consists of three tasks. Process modeling is the first task in the life cycle of a process. In this task a process is designed or changes to an existing process are made. The result is a new or modified process. A process comprises a set of activities which have to be executed in order to achieve a business goal. So-called control flow defines the order in which the activities have to be executed. The second task is the execution of the process. The execution is supervised in process monitoring, which may run parallel to the execution. Process monitoring closes the loop and leads back to process modeling and redesign respectively.

Although, seen from a more technical perspective, there are some more steps in this life cycle. A business process is typically modeled on a high level of abstraction in a language near to business, for instance by using the Business Process Modeling Notation (BPMN) [6]. Technical refinement is a step in between process modeling and execution, here a process is prepared for execution by technical personnel. Possibly, the process also needs to be transformed into a different language for execution, for instance to the Business Process Execution Language (BPEL) [7]. Further steps cover verification, validation, and technical monitoring.

We interpret the term *compliance* as conforming to particular requirements originating from the interpretation of compliance sources [3]. We assume that a

F. Buccafurri and G. Semeraro (Eds.): EC-Web 2010, LNBIP 61, pp. 131–142, 2010.

compliance assessment is made by experts (e.g. lawyers) who interpret the compliance sources and break them down to concrete requirements. Compliance sources can be laws from the executive, regulations like Basel II [14], internal policies, industrial standards and also business agreements. Non-compliance can mean significant punishments to a company. Hence, companies are in urgent need to prevent violations by detecting them and reacting accordingly. Besides the installation of one or more compliance officers who take care that all compliance requirements are met, the business processes that drive the business are affected as well. This also has an impact on the tasks related to the process life cycle described above.

Amongst other things, compliance needs to be addressed in process modeling. Some requirements occur frequently and their realization thus is feasible for reuse. We proposed the notion of process fragments for compliance, abbreviated as *compliance fragments* [11], to represent the realization of compliance requirements concerning a process. A compliance fragment can be understood as a connected subgraph of a process graph which addresses requirements related to compliance. Such a fragment has significantly relaxed completeness and consistency criteria compared to an executable process graph. A process graph consists of nodes which represent the activities of a process, and edges which represent control dependencies.

For the management of these fragments we need several techniques. In order to create reusable compliance fragments we need a technique for extraction of process structures which realize a compliance requirement in terms of activities and control flow. Then, we need a technique for integrating such reusable compliance fragments into other business processes that have to be augmented with compliance. In order to proof compliance to an auditor, we also have to define a mechanism for highlighting integrated compliance fragments. Despite integration and highlighting we also need a method for fading those structures out. In other words we need a way to hide those steps which do not represent the actual "work" in the business process. In [11] we have denoted this as process pollution problem. In summary, compliance fragments serve as reusable process structures which realize particular compliance requirements related to a business process. Process views for compliance, on the other hand, provide a means to work with such structures.

This paper is structured as follows: Section 2 contains references to works related to our approach. Section 3 describes the challenges of managing of compliance fragments in business processes. In Section 4 the process view transformations for extraction and hiding of compliance fragments are elaborated. In Section 5 we discuss the limitations of our approach with respect to compliance management in general. Section 6 gives a short summary of the paper and characterizes future work.

2 Related Work

Due to the relevance of compliance management in business processes, an increasing number of works on this topic exists. Current approaches address all tasks related to the process life cycle, ranging from modeling of process constraints to their verification and checking for violations in monitoring. Most of the approaches are based on annotations that constrain the behavior of a process, preferably using domain-specific languages or formal rules as shown in [15]. Those approaches are very important to formally constrain a process and to formally proof compliance of a process and its execution respectively. However, these solutions do not address how to ensure a consistent specification of requirements in terms of activities and control flow in order to augment a process with compliance. We tackle this issue with the concept of compliance fragments [11], and together with our research partners we combined this concept with the formalization of requirements and process verification [17].

In our former work [11] we proposed two different methods for integrating compliance fragments into a process. The first method (which we called gluing) is to physically copy the fragment into the process. The second method is to make use of Aspect-Oriented Programming (AOP) techniques to augment a process with compliance fragments in a loosely coupled manner. This method is feasible for many compliance requirements, for instance related to auditing and logging. When aspect weaving is applied there is in fact no need for compliance fragment hiding, as the fragment is already separated from the process. However, the integration of some compliance fragments requires a physical redesign of the process, that is why gluing is not avoidable in any case. In particular, this is the case for compliance fragments with multiple entries or multiple exits, such as a compliance fragment for an approval which has one exit for acceptance, and another one for rejection.

Process views are a set of approaches addressing the increasing size of business processes, i.e. concerning the increasing number of activities which are contained in a process. Aside from the application for process abstraction, process views can also be used in other scenarios. In [1] an application of view transformations for extraction of reusable structures from Petri Nets is discussed. This mechanism is related to our approach of fragment extraction, though specified with a different purpose and for a different language. An approach for the generation of a public process for usage in outsourcing scenarios is presented in [5]. The mechanism in [5] is similar to the mechanism of hiding which we propose. However, it is also applied to a different language and limited in extensibility of the supported transformation functions. In [8] an overview on further application scenarios for process views is given. In general, most approaches make use of omission and aggregation of structures, for instance the above mentioned works. We argue that viewing concepts can be utilized as a means to support the management of compliance structures in business processes. To the best of our knowledge there is currently no comparable approach in this field.

3 Managing Compliance Fragments in Business Processes

In this section, we discuss the main challenges of managing compliance fragments in business processes, guided by a running example. A frequent compliance requirement is related to reviewing and assessing a particular situation. Let us assume for example an internal business process for approval of vacations. This process needs to be compliant with the requirements originating from internal policies. For this reason this process contains, among other steps, activities related to checking up on conflicts. To be more precise, it contains a fragment for checking up on conflicts concerning fixed appointments in the requested vacation time.

Extracting Compliance Fragments. To make the implementation of this compliance requirement reusable, the according structures need to be extracted. We propose to add a special tag on the activities that belong to the structures which should be extracted, see the illustration in Figure 1. This tag states that any kind of view transformation has to preserve those structures and all the artifacts (e.g. variables) which are related to them. With this method we need to define a view transformation that triggers the omission of all other activities. The implementation of the transformation needs to preserve the tagged structures and maintain control dependency. The result of the view transformation is a fragment for checking up on appointment conflicts. After a fragment has been extracted it can be stored in a fragment repository [13], ready for reuse, highlighting and hiding. Another possibility for the creation of a fragment is to design it from scratch as discussed in [11]. In Section 4 we present an implementation of this view transformation.

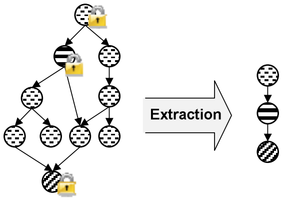


Fig. 1. Compliance Fragment Extraction

Integrating Compliance Fragments. Sometime later the internal business process for approval of business trips also needs to comply with the requirement for checking up on appointment conflicts. For this, we need to integrate the fragment which implements this requirement into the business trip approval process, see Figure 2. For the integration at first the entries and exits of a fragment have to be wired. This can be done by breaking existing control edges in the process and inserting the fragment in between existing structures, otherwise new control edges have to be inserted. To complete the integration, the context of the fragment (variables, etc.) has to be merged with the process context. During integration conflicts have to be resolved. For instance, parameter types used in the fragment possibly have to be adjusted to those used in the process (e.g. Boolean vs. String). Process view transformations are not applicable for this task. Therefore, we are currently working on a methodology for integration of compliance fragments and their related context into a given process.

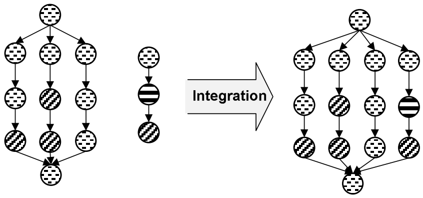


Fig. 2. Compliance Fragment Integration

Highlighting Compliance Fragments. During an audit a company has to provide all relevant information to proof compliance with laws and regulations. As described in [2], internal audits are an important measure, too. Applied to our running example, we need to provide information to an internal auditor on how we addressed the requirement for checking up on appointment conflicts. Process views also refer to the visualization of a process. Therefore, we propose using sub-graph matching algorithms [16] for the identification of known fragments related to compliance. The result of this fragment recognition step provides an input for according highlighting in graphical display of the process, as illustrated in Figure 3.

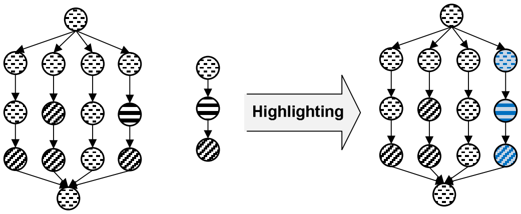


Fig. 3. Compliance Fragment Highlighting

We are currently extending our view transformation framework to support this application. The transformation component is based on Java/DOM, the visualization and modeling component is an extension of an open source process design tool [4]. In order to enable flexible highlighting we modify the predefined paint methods of the process constructs to adjust the visualization, in the same manner as shown in former work [9]. We identified several display properties of activities which can be customized to provide the highlighting (see Figure 4). For instance, using red border color with increased thickness already provides a straightforward solution. More advanced settings (e.g. involving shape size) are conceivable though.

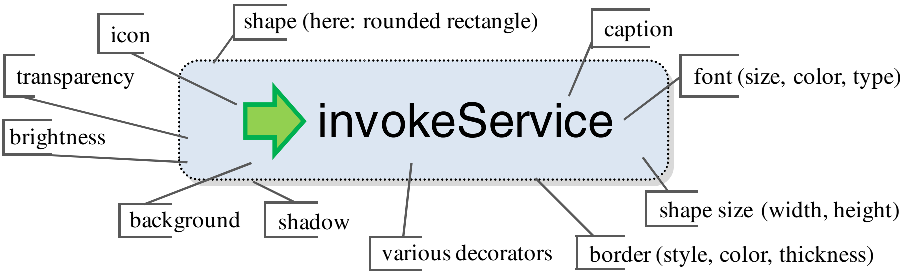


Fig. 4. Graphical Display Properties of Activities

Hiding Compliance Fragments. As mentioned in the introduction, process structures related to compliance sometimes do not represent the actual work that needs to be carried out in a process. If the number of compliance requirements that have to be addressed increases, the process becomes “polluted” and harder to understand.

Therefore, we propose a view transformation for hiding those fragments in order to provide a clear view on an unpolluted process, see Figure 5. In Section 4 we discuss a technical implementation for this method.

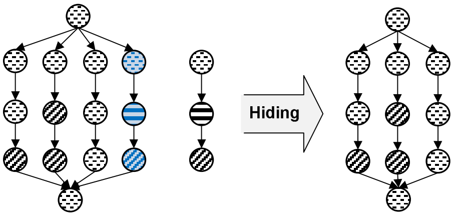


Fig. 5. Compliance Fragment Hiding

4 Extracting and Hiding of Compliance Fragments

In this section, we demonstrate how process viewing concepts can be used to extract and hide compliance fragments. In the work of [10], we have implemented a human-centric, model-driven framework for process view transformations based on the process language BPEL [7]. The BPEL standard provides a brief introduction to the notion of abstract processes. Abstract processes are either used to hide language elements of an executable process or they are not yet fully specified and serve as a process template. In our framework we currently use the metamodel of Abstract BPEL processes to represent compliance fragments. Originally, our framework has been designed to enable a semi-automatic generation of public processes to facilitate efficient process outsourcing as we discuss in [12]. However, in the following we show how to exploit and extend this framework to support also extraction and hiding of compliance fragments.

4.1 Principles of the Process Views

The process view transformation for fragment extraction which we propose requires a manual preparation step before the actual automated transformation can take place. In this preparation step the structures of the process that should be extracted have to be manually tagged for preservation. When we translate this to the example discussed in Section 3 then all structures related to checking up on appointment conflicts have to be tagged. In [10] we have shown how to extend a process design tool (Eclipse BPEL Designer [4]) to provide end-user support for this task. Figure 6 shows how this extension enables the user to add an annotation to selected activities via the context menu. The result of this step is a tagged process.

In addition to tagging, transformation rules which steer the transformation (described in detail in Section 4.2) have to be specified. These rules indicate which process constructs (targets) should be transformed, and which particular transformation operation should be performed (actions). The tagging step eases the definition of such rules, as targets can then be easily defined based on the annotations made. If no annotations are available, then construct attributes like activity name, portType etc. have to be used to select the targets.

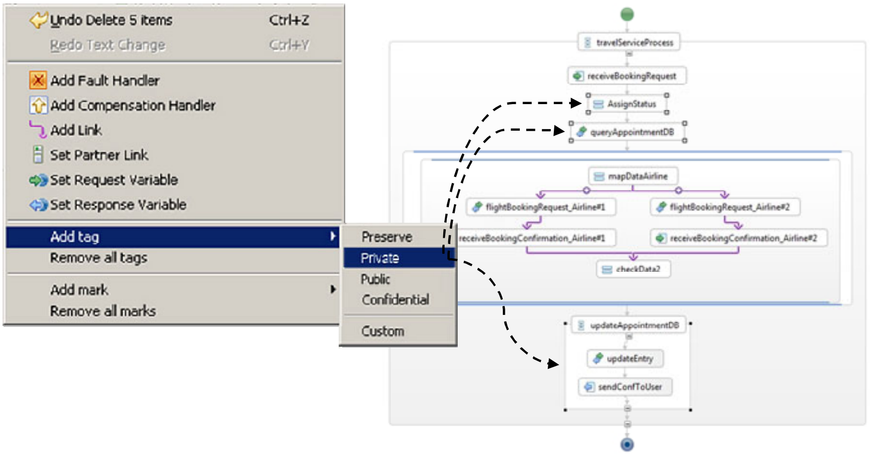


Fig. 6. Extension of the Eclipse BPEL Designer [4] for Annotation of Activities

Eventually, transformation actions are applied to the process, based on the transformation rules and the tagged input process. This results in a process view, i.e. in our extraction example in an abstract process containing only the preserved compliance fragment for checking up on appointment conflicts. The proceeding of extraction is depicted in Figure 7.

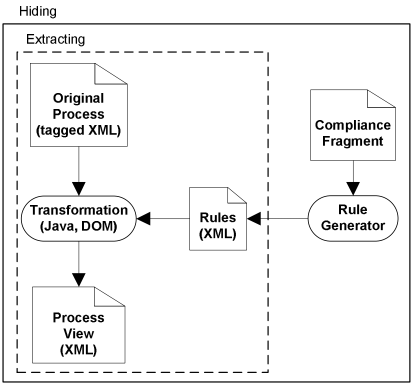


Fig. 7. Viewing Application Principle

For the implementation of the functionality for fragment hiding we have to extend this framework by another component, which we call rule generator, see the right part in Figure 7. The rule generator takes the fragment that should be hidden as input. Translated to our example, the compliance fragment for checking up on appointment conflicts would be such an input. For each activity and control structure in the fragment a rule for its omission is being generated. The generated rules are passed to the transformation component, which applies them to the input process. This transformation results in a process view that does not contain the input fragment anymore. Applied to our example, each activity of the checking up on appointment conflicts fragment would be omitted while preserving control dependencies.

4.2 Specification of the Process View Transformation

We have designed a rule language with a simple grammar which is on the one hand easy to handle, but on the other hand also capable of describing complex transformation statements. With this language a view transformation can be specified as a list of transformation rules. Each rule can target multiple constructs and trigger multiple transformation actions to be applied to these constructs. The language can be easily extended by new actions, parameterization options or new targeting possibilities. Instead of designing a new language from scratch, we could have also used (or extended) existing transformation languages like QVT (Query / View / Transformation) or ATL (ATLAS Transformation Language). However, these well-established languages are already very powerful and finding the minimal requirements a language for process view transformations has to meet was one of our research interests.

A transformation specification has the following structure:

```
<rules>              <!-- an ordered list -->
  <rule>*            <!-- multiple rules may be contained -->
      <actions/>     <!-- which actions have to be applied -->
      <targets/>     <!-- to which constructs -->
  </rule>
</rules>
```

The `<actions>` element denotes which actions have to be applied. It contains at least one `<action>` element. All nested child actions are executed in the order in which they are specified. If an action cannot be performed, e.g. if no construct is being addressed, the action will be skipped. Our framework currently supports three actions that can be used:

i. `actionOmit:` Omitting an arbitrary construct, i.e. the construct is removed from the process. If an activity is omitted, then existing control dependencies are being preserved.
ii. `actionOpaque:` Transforming an activity into an opaque activity [7], i.e. the activity is not removed completely but all implementation details are hidden.
iii. `actionSetAttributeTo:` Changing the value of an attribute of an arbitrary construct. If the value is set to NULL, the attribute is being removed.

The `<targets>` element is used to indicate the target constructs that should be affected by an action. Child elements can either be logical connectors (`<or>`, `<and>`, or `<not>`), or target elements. Logical connectors can be used to combine the different target elements. Our framework currently supports the following targeting possibilities:

i. `tag:` Targeting based on annotations, e.g. activities annotated with the tag “preserve”.
ii. `attribute:` Targeting based on name-value pairs of XML attributes.
iii. `type:` Targeting based on the XML element type.

We can use the rule language to specify a general transformation rule for the extraction of compliance fragments, assuming that the parts of the compliance fragment are annotated with the tag “preserve”. The rule for extraction instructs the transformation to omit all structures which do not belong to the compliance fragment:

```
<rules>
  <rule name="extractFragment" apply="true">
      <actions>
          <actionOmit preserveChildren="true"/>
      </actions>
      <targets>
          <not>
            <tag tagName="preserve" />
          </not>
      </targets>
  </rule>
</rules>
```

For the hiding of a compliance fragment we use a rule generator to automatically create the required rules. For each construct contained in the input fragment, we generate a rule for its omission. The implementation of the omit action preserves consistency of the resulting view. The generated rules are based on the following scheme:

```
<rule name="omit%CONSTRUCT-NAME%" apply="true">
    <actions>
        <actionOmit preserveChildren="true"
                    preserveTransitionConditions="false"/>
    </actions>
    <targets>
        <attribute attributeName="name" value="%CONSTRUCT-NAME%" />
    </targets>
</rule>
```

We currently use a name-based matching for the hiding of fragments. Our framework also allows a matching based on unique identifiers though. Parameters in the elements allow refining the transformation, e.g. `preserveChildren` preserves nested structures from being removed while its parent construct is omitted. This is especially related to structured activities in BPEL like a `<forEach>` loop. `PreserveTransitionConditions` maintains all transition conditions on links in a flow (the graph-based component in BPEL) while only hiding the targeted activity itself.

4.3 Execution of the Process View Transformation

The specification and implementation of transformation actions on BPEL constructs is quite complex as the overall result of the transformation has to be consistent. The omission of activities with multiple control dependencies or the preservation of nested activities is one of the main challenges. For example, when omitting a `<sequence>` the `<sequence>` itself and all child activities are removed, except for the ones tagged with the preserve-tag. However, the result of such a transformation can be ambiguous without further information provided by the user. In this case, we decided to use a `<flow>` as a container because this construct can contain activities without any control dependencies. For this reason this construct is ideal to put the preserved activities "loosely" inside. An enhanced version could show different possible results to the user from which he may choose the one he actually intended. Furthermore, a new `<scope>` that encloses the `<flow>`-container is required. This is necessary because variables defined locally may need to be preserved as well, e.g. if the `<sequence>` contains a structure that defines variables locally.

Another example for ambiguity of the transformation is an activity which should be removed from a `<flow>` as depicted in Figure 8: Figure 8a depicts the original process, where activity X should be omitted. Figure 8b is a consistent solution to this as it maintains control dependency, though it increases complexity. Figure 8c and 8d suggest alternatives, however original semantics are changed compared to 8a. Figure 8e illustrates a solution which inhibits X from being completely removed, but provides simplicity.

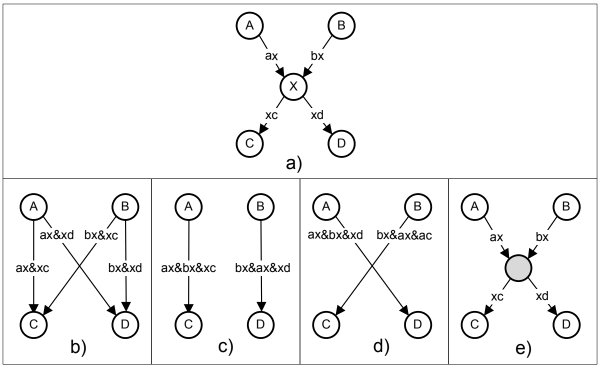


Fig. 8. Omission of Activities in a Flow

Transition conditions have to be handled in order to preserve the semantics of the process. To be consistent, omission can be realized by removing activity X and all adjoining links. Afterwards new links with appropriate transition conditions are inserted. To preserve full coherence of activities, new links containing properly concatenated transition conditions must be inserted to connect every activity to all subsequent activities (see Figure 8b). The dropping of non-needed links is performed similarly to accomplish more adequate visualization (see Figure 8c and 8d), even if semantics are changed and transition condition handling becomes quite complex. In our current implementation we have solved this problem as shown in Figure 8e. It states that X will not be removed completely when executing omission on this activity. X is transformed into an `<opaqueActivity>` automatically.

To speed up post-editing of the resulting process views, cleaning functions are implemented. It is likely that unnecessary constructs are still left in the process which should be removed automatically after the transformation rules have been applied. For instance, unused `<variables>` or `<partnerLinks>` may not longer be needed, because corresponding activities have been omitted or transformed into opaque activities. In addition, structured activities without any nested activities can be removed, e.g. removing an empty `<sequence>` is reasonable. The implementation offers a set of predefined functions to clean up the process.

5 Limitations of the Approach to Manage Compliance

The framework we presented can be extended by further transformation actions, e.g. an aggregation of multiple activities can be used for process abstraction which might ease the work of an auditor. Furthermore, many of the concepts we presented are not limited to the BPEL language and can thus be applied to other process languages as well. The process views which we proposed support the management of compliance fragments. Compliance fragments are capable of addressing compliance requirements which are related to control flow and activities within a process, but it has to be said that compliance management in general comprises many more aspects.

A fragment of a process can neither ensure compliance of the humans which are involved in that process, nor can it control the applications and services which it orchestrates. For instance, a requirement that demands storage of travel expense reports for at least ten years is related to a database which is external to the process. Besides, executing a process for over ten years would not be efficient. Even when focusing only on business process automation, compliance already has an impact on all involved components and related tasks: design, verification, technical refinement, execution, and monitoring. However, many compliance requirements (e.g. related to security, privacy or trust) have an impact on all of the components in the IT infrastructure of a company, which also includes ERP or CRM systems. In addition, compliance also has an impact on the business processes which run outside of the IT systems. This necessitates further methods to manage compliance. For instance, employees have to internalize the code of business conduct of a company - interviews can be used to check if the employees adhere to this code. Thus, compliance fragments are just one aspect of control in an overall solution to compliance management.

6 Conclusion and Future Work

In this paper we discussed the major challenges of managing compliance fragments in business processes. The main contributions of this work comprise a technique for extracting and hiding of compliance fragments based on process view transformations. Moreover, we have discussed a practical implementation of this technique for a language that is commonly used in industry for Web service orchestration. Currently we are extending our framework in order to enable automatic recognition of compliance fragments. Therefore, we utilize algorithms for sub-graph matching in order to recognize compliance fragments that are integrated in a process. With this technique we support an auditor with a tool to check if a particular compliance requirement is addressed and how it is integrated into a process. Furthermore, we are implementing support for compliance fragment integration in order to cover the whole life cycle of compliance fragment management: Extraction, integration, highlighting and hiding. As supporting infrastructure we are developing a view designer component that eases the specification of transformation rules, and a repository for storage and retrieval of processes and compliance fragments [13].

Acknowledgements

The work published in this article was partially funded by the COMPAS project (http://www.compas-ict.eu) under the EU 7th Framework Programme ICT Objective, contract no. FP7-215175.

References

1. Avrilionis, D., Cunin, P.Y., Fernström, C.: OPSIS: a View Mechanism for Software Processes which Supports their Evolution and Reuse. In: Proc. of the 18th International Conference on Software Engineering (ICSE). IEEE Computer Society, Los Alamitos (1996)
2. Caprasse, D., Laurent, J., Reed, W.: Three Lines of Defence: How to take the Burden out of Compliance. In: European Insurance Digest (April 2008)
3. Daniel, F., Casati, F., D'Andrea, V., Strauch, S., Schumm, D., Leymann, F., Mulo, E., Zdun, U., Dustdar, S., Sebahi, S., de Marchi, F., Hacid, M.S.: Business Compliance Governance in Service-Oriented Architectures. In: Proc. of the 23rd IEEE International Conference on Advanced Information Networking and Applications (AINA). IEEE Press, Los Alamitos (2009)
4. Eclipse BPEL Project. Eclipse BPEL Designer (2010), `http://www.eclipse.org/bpel/`
5. Eshuis, R., Grefen, P.: Constructing Customized Process Views. Data & Knowledge Engineering 64(2), 419–438 (2008)
6. Object Management Group (OMG). Business Process Modeling Notation Version 1.2. OMG Standard (2009)
7. Organization for the Advancement of Structured Information Standards (OASIS). Web Services Business Process Execution Language Version 2.0. OASIS Standard (2007)
8. Rinderle, S., Bobrik, R., Reichert, M., Bauer, T.: Business Process Visualization – Use Cases, Challenges, Solutions. In: Proc. of the International Conference on Enterprise Information Systems (ICEIS), pp. 204–211. INSTICC Press (2006)
9. Schumm, D., Karastoyanova, D., Leymann, F., Nitzsche, J.: On Visualizing and Modelling BPEL with BPMN. In: Grid and Pervasive Computing Workshops: 4th International Workshop on Workflow Management (ICWM). IEEE Press, Los Alamitos (2009)
10. Streule, A.: Abstract Views on BPEL Processes. Diploma thesis no. 2889, Universität Stuttgart, Fakultät Informatik, Elektrotechnik und Informationstechnik (2009), `ftp://ftp.informatik.uni-stuttgart.de/pub/library/medoc.ustuttgart_fi/DIP-2889/DIP-2889.pdf`
11. Schumm, D., Leymann, F., Ma, Z., Scheibler, T., Strauch, S.: Integrating Compliance into Business Processes: Process Fragments as Reusable Compliance Controls. In: Proc. of the Multikonferenz Wirtschaftsinformatik, MKWI'10 (2010)
12. Schumm, D., Leymann, F., Streule, A.: Process Viewing Patterns. Accepted for publication at the 14th IEEE International EDOC Conference (EDOC 2010). IEEE Computer Society Press, Los Alamitos (2010)
13. Fragmento - Fragment-oriented Repository. Online Documentation (2010), `http://www.iaas.uni-stuttgart.de/forschung/projects/fragmento/start.htm`
14. Basel Committee on Banking Supervision. International Convergence of Capital Measurement and Capital Standards: A Revised Framework (2006)
15. Sackmann, S., Kähmer, M.: ExPDT: A Policy-based Approach for Automating Compliance. In: Wirtschaftsinformatik (WI), Gabler, vol. 50(5), pp. 366–374 (2008)
16. Ma, Z., Lu, W., Leymann, F.: Query Structural Information of BPEL Processes. In: Proc. of the 4th International Conference on Internet and Web Applications and Services, ICIW (2009)
17. Schumm, D., Turetken, O., Kokash, N., Elgammal, A., Leymann, F., Heuvel, W.v.d.: Business Process Compliance through Reusable Units of Compliant Processes. In: Accepted for Publication at the 1st Workshop on Engineering SOA and the Web (ESW'10), Springer, Heidelberg (2010)

Web Advertising

Ricardo Baeza-Yates

Yahoo! Research
Barcelona, Spain

Abstract. We briefly review the science and the technology behind Web advertising. We first describe the various forms of Web advertising served up by advertising networks such as Google, MSN and Yahoo!. Then we introduce the technical challenges behind problems as: (1) given a user's search query, how do we determine which advertisement(s) to present? (2) how many advertisements we should present on a particular query? (3) how we should price advertisements? The solutions to these problems span research areas ranging from text mining and information retrieval, to the theory of auctions and marketplaces, being called today *computational advertising*.

Keywords: Web advertising, text mining, auction theory, pricing.

1 Summary

Computational advertising is *"a new scientific discipline, at the intersection of information retrieval, machine learning, optimization, and microeconomics. Its central challenge is to find the best ad to present to a user engaged in a given context, such as querying a search engine ("sponsored search"), reading a Web page ("content match"), watching a movie, and IM-ing"* [1].

Computational advertising has two main research problems. The first and best known is matching advertisements to a given query and displaying them in the results page of a search engine. This case is called *sponsored search*. The second problem is finding the right advertisements that should be included on a given page that is being requested by a user while browsing. This case is called *content match*.

The advertisement itself can be of two types: image or text based (or a combination of both). The initial model of display advertisement was mainly images and the search based model is usually text. In many cases, placing the advertisement is usually done by an intermediary, called the *ad-network*. The business model is typically a payment per impression, in the case of display advertisement (PPM), or a payment per click, in the case of sponsored search or content match (PPC). A third and newer model is cost per action in which the advertiser only pays if a specific target action is achieved, such as a commercial transaction.

Computational advertising can be seen as a search problem, where the search input, either a query or the content of a page, has to be matched against a database of advertisements. Every advertisement in the database, usually has at

F. Buccafurri and G. Semeraro (Eds.): EC-Web 2010, LNBIP 61, pp. 143–145, 2010.

least a title, a target URL, and a textual description. One important difference with classical Web search is that the search input, in the case of content match, can be much larger than the advertisements themselves. Because of the small size of the descriptions, pure textual similarity with the query (be it short or long like in content search) will not bring enough relevant advertisements to the users. To address this problem, text descriptions are systematically augmented by large lists of keywords, which are either generated by the advertisers or sometimes offered by the advertisement system.

The matched advertisements must be ranked to only present the "best" ones to the users [4,7,5,8,3]. However, the quality here is not only based on relevance, but also on commercial considerations. In fact, as advertisers pay according to the number of clicks issued by users until they exhaust a fixed budget. When the budget is spent, no more advertisements are shown. This payment is modeled by an auction mechanism where advertisers bid (and hence compete in an open market) on the previously mentioned keywords associated with each text description. The original scheme was invented by the Goto search engine (later renamed Overture and bought by Yahoo!), and the order was just based on the bidding price. However, if no clicks are made for lack of relevance, this model fails. The actual approach adopted by most search engines thus uses a combination of auction bids (where advertisers bid on specific keywords) and predicted relevance models based on expected click through rate (CTR) of the advertisement, which is estimated using past history. The auction mechanism should be truthful. However when there is more than one winner, as in our case, truthful mechanisms are more complicated.

For further research in computational advertising see [2,6,7].

References

1. Broder, A., Josifovski, V.: Introduction to computational advertising. Course at Stanford University (September-December 2009), `http://www.stanford.edu/class/msande239/`
2. Broder, A.Z., Ciccolo, P., Fontoura, M., Gabrilovich, E., Josifovski, V., Riedel, L.: Search advertising using Web relevance feedback. In: CIKM '08: Proceeding of the 17th ACM Conference on Information and Knowledge Management, pp. 1013–1022. ACM, New York (2008)
3. Broder, A.Z., Fontoura, M., Josifovski, V., Riedel, L.: A semantic approach to content advertising. In: Kraaij, W., de Vries, A.P., Clarke, C.L.A., Fuhr, N., Kando, N. (eds.) SIGIR, Amsterdam, The Netherlands, November 2007, pp. 559–566. ACM, New York (2007)
4. Jones, R., Rey, B., Madani, O., Greiner, W.: Generating query substitutions. In: WWW'06: Proceedings of the 15th International Conference on World Wide Web, pp. 387–396. ACM Press, New York (2006)
5. Lacerda, A., Cristo, M., Gonçalves, M.A., Fan, W., Ziviani, N., Ribeiro-Neto, B.A.: Learning to advertise. In: Proceedings of the 29th ACM Int. Conference on Information Retrieval, ACM SIGIR, pp. 549–556 (2006)

6. Radlinski, F., Broder, A., Ciccolo, P., Gabrilovich, E., Josifovski, V., Riedel, L.: Optimizing relevance and revenue in ad search: a query substitution approach. In: SIGIR '08: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 403–410. ACM, New York (2008)
7. Ribeiro-Neto, B.A., Cristo, M., Golgher, P.B., de Moura, E.S.: Impedance coupling in content-targeted advertising. In: SIGIR, pp. 496–503 (2005)
8. Yih, W.-t., Goodman, J., Carvalho, V.R.: Finding advertising keywords on Web pages. In: Carr, L., De Roure, D., Iyengar, A., Goble, C.A., Dahlin, M. (eds.) WWW, Edinburgh, Scotland, UK, pp. 213–222. ACM, New York (2006)

Electronic Trading Environments for Web 3.0

John Debenham[1] and Simeon Simoff[2]

[1] QCIS, FEIT, University of Technology, Sydney, Australia
debenham@it.uts.edu.au
http://www-staff.it.uts.edu.au/∼debenham/
[2] School of Computing & Mathematics, University of Western Sydney, Australia

Abstract. This paper proposes that electronic marketplaces for Web 3.0 can be described through three metaphors: "marketplaces where people are", "marketplaces that are alive and engaging", and "market places where information is valuable and useful". The paper presents the core technologies that enable the perceivable reality of electronic marketplaces. It describes a demonstrable prototype of a Web-based electronic marketplace that integrates these technologies. This is part of a larger project that aims to make informed automated trading an enjoyable reality of Web 3.0.

1 Introduction

A market is commonly defined as a physical or virtual location, where price is determined and buy and sell orders are matched to create trades according to a set of rules that govern the processing of these orders [1]. Electronic markets have been viewed as information systems "that allow buyers and vendors to exchange information about prices and product offerings [2]. This and similar views have guided the development of "soulless" electronic markets, focussed primarily on enabling standardised or complex transaction processes. Thus automation of electronic markets have been focused on the secure backend transaction processing. A recent review of the area (see Chapter 18 "Electronic Marketplaces and Resource Exchanges" in [3]) provides a broader picture from various perspectives, including agent-based negotiation, brokering, and partnership formation. Still, the operation and the interactions in such Web-based electronic markets reflect the dominating content-based systems approach of Web 2.0. Though useful, these electronic markets are far from being realistic trading places.

In this paper we consider electronic (virtual) marketplace to be a regulated space populated by computerised players that represent a variety of human and software traders, intermediaries, and information and infrastructure providers. Such marketplace is where things and traders have presence, constituting a rich interaction space [4]. The agreed regulations operating in the space structure the interactions between the different contributors. We borrow the metaphor from [5].

F. Buccafurri and G. Semeraro (Eds.): EC-Web 2010, LNBIP 61, pp. 146–157, 2010.

1.1 Electronic Marketplaces for Web 3.0

Electronic marketplaces for Web 3.0 should attempt to model the full richness of interactions including natural language communication, gestures, and emotional expression, as well as the cognitive apparatus that underlies these capabilities [6]. Most virtual human research has focused on the cognitive behaviour on the source side of the interaction [7] with a recent shift towards the "recipient" [8].

One inspiring contribution is the Carnegie-Mellon set of requirements for realistic agents, which is based on research in drama and story telling [9]. These include personality, self-motivation, change, social relationships, and "illusion of life". Personality infuses everything that a character does — their behaviour, style, "thought", "emotion", e.g. their unique ways of doing things. Self-motivation assumes that agents have their own internal drives and desires which they pursue whether or not others are interacting with them, and they demonstrate their motivation. Change implies that characters change with time, in a manner consistent with their personality. Behaviour of agents and interactions between them should be in a manner consistent with their social relationships (in turn, these relationships change as a result of the interaction). "Illusion of life" is used as a label for a collection of features such as: pursuing multiple, simultaneous goals and actions, having elements of broad capabilities (e.g. movement, perception, memory, language), and reacting quickly to stimuli in the environment. In this sense convincing does not necessarily mean realistic. We discuss briefly the issues in the Carnegie-Mellon set of features:

Regulations: Norms are part of interactions between trading partners. Collectively they constitute a complex, structured, regulatory system that should be consistent. In a convincing trading environment, in addition to compliance with regulations, some times there could be some modifications based on mutual agreements. Background details to the operationalisation of norms in 3D virtual spaces are considered in [10].

Processes: The structure of the business processes in electronic markets define the narrative of the marketplace. Market players operate in the context of the process structures under the constraints of the regulatory framework.

Spaces: Humans are embodied in space in all their behaviour. They inhabit and operate in it; rely on and use various cues related to space, like pointing and referring to areas of and things in it (for more details see the first two chapters in [11]). This is an essential factor driving the technological conquest for moving us from being on the Internet to gradually being in the Internet space, i.e. towards what is labelled 3D Internet [12]. The evolution relies on several technologies that enable primarily perceptual immersion, including virtual worlds and immersive access to digital content [13]). In terms of the realism of electronic environments, the virtual space is essential part of what constitutes an *intelligent environment populated with intelligent artefacts.* Intuitively, to be realistic electronic markets should have arrangement of their virtual spaces that are aligned with the business processes in them.

Interactions: As a result of their capability to dig out and paste together various pieces of useful information, traders usually are informed to a different

> extent about various aspects of the deals they are pursuing. When necessary they rely on various relations with other traders that they have established over the time. Their decisions are result of the mix of being rational, informed, impulsive, and the ability to influence others and cope with the influences from others. All these nuances impact the richness of market interactions, hence, must be taken into account when considering the interactions in electronic markets.

This paper presents the core technologies developed to address these issues. Section 2 describes the underlying theoretical and practical solutions of demonstrable prototype of a Web-based electronic marketplace. Section 3 presents the machinery that enables market players to act convincingly in the uncertain, diverse and very dynamic environment of Web-based electronic marketplaces. Section 4 concludes.

2 Regulated Virtual Spaces

In order to address regulatory requirements that operate in real world research in multiagent systems adapted social science theories and concepts like norms [14]. Normative multiagent systems relate agent theory and the social sciences such as sociology, philosophy, economics, and legal science. The electronic institutions (EI) methodology and technology for normative multi-agent systems (MAS), developed by IIIA CSIC in Barcelona [15], elegantly formalises and implements the institutional approach for MAS. The work on virtual institutions (VI) [10] developed the institutional approach further for inhabited 2D and 3D spatial environments, including virtual worlds. The VI concept and development methodology [10] extends the EI approach, enabling the implementation of institutional commitments that ensure rich and reliable interaction between embodied entities - avatars, whether they are driven by autonomous agents or humans. Central to the implementation of institutional behavioural norms in the EI methodology is the notion of performative structure which formalises processes in terms of scenes, agent roles and communication language. A set of (business) processes are modeled as a discrete collection of interlinked ordered scenes. The involvement of participants (agents) in these processes is modeled through a set of roles, where roles are related to the scenes by the set of participation rights (constraints) for each role in respective scenes, including the subset of the language that can be used in each scene. The later defines the set of permitted dialogues [15]. The VI concept [10] enables the institutionalisation of a virtual world with respect to a performative structure in terms of (i) the spatial layout of the virtual world that reflects the performative structure of the (business) processes; (ii) the objects and avatars and their behaviour within the institution with respect to their roles; and (iii) the rich interaction based on natural language and embodiment of humans and software agents in the institutionalised world.

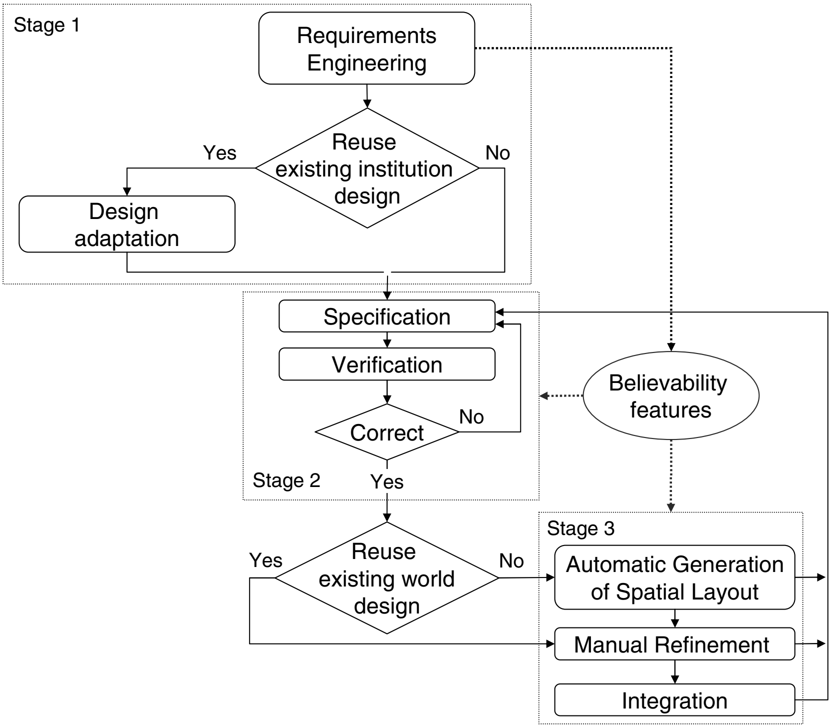
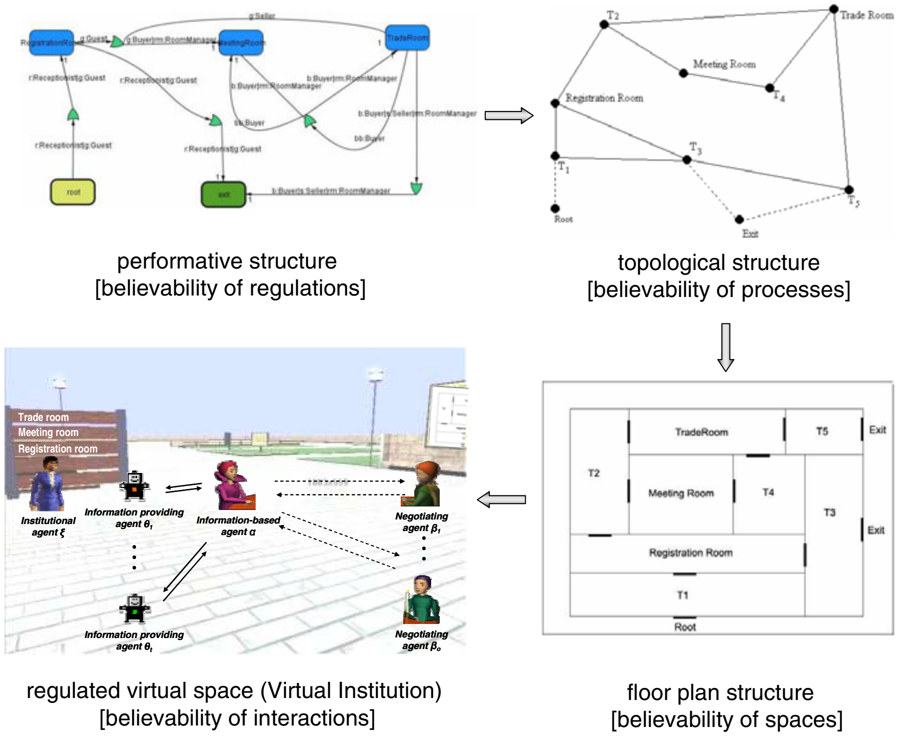


Fig. 1. Extending the VI steps described in [10] for designing electronic markets

We extend the VI methodology [10] to embed realistic components. Figure 1 presents the extended methodology with the steps grouped into three stages. Stage 1 considers the initial requirements engineering of the institutional environment, based on the type of the market in consideration. This stage can generate specific requirements towards the believability of the environment. For instance, if the electronic market is a type of supermarket, then a believability requirement may be the identical arrangement of the spatial layout of the shelves and the presentation of the goods in the same order as in the respective physical shop, emulating what customers are used to. In a property market, the software agent that acts on behalf of a property trader may need to look like ”being well informed” and to have the ability to deal with the new information that supersedes the existing information. The set of believability requirements usually translates into elements of the specification of the performative structure, feeds into Stage 2 and propagates further into the layout of the virtual institution in Stage 3. Requirements that are mostly related to the visual presence, like the style of movement (e.g. trajectories, gestures) as well as believability through graphical appearance are considered directly at Stage 3 and may require refinement of the performative structure.



performative structure
[believability of regulations]

topological structure
[believability of processes]

regulated virtual space (Virtual Institution)
[believability of interactions]

floor plan structure
[believability of spaces]

Fig. 2. Enabling the dimensions of electronic markets during the generation of respective virtual institution

Figure 2 presents a high level view of how the VI technology creates a regulated virtual space on the Web. The topology of the space is extracted from the performative structure and then converted into an initial set of spatial envelopes optimally packed in a bounded institutional space.

The technological architecture of virtual institutions that supports convincing electronic markets is shown in Figure 3. The Normative Institutional Layer, specified by the steps in Stage 1 and Stage 2 in 1) takes care for the functioning of the institution as a normative multi-agent system and relies on the EIDE platform [15]. An institutional governor agent is associated with each 'player' G_1, ..., G_n (whether G_i is a human, a software agent, or another institution). Together with the regulatory mechanism and the execution state of the virtual institution it ensures that the player operates according to the regulatory protocols in each step of the business process. The Intercommunication layer and the Translation Layer enable the causal connection between the institutional infrastructure and the 3D Institutional layer, transforming the actions in the 3D Institutional Layer into messages in the language of the Normative Institution Layer and and vice versa. The role of the Translation Layer is to process interactive 3D content, compliant with the X3D standard [16] and translate it to different virtual world platforms (currently - SecondLife [17]).

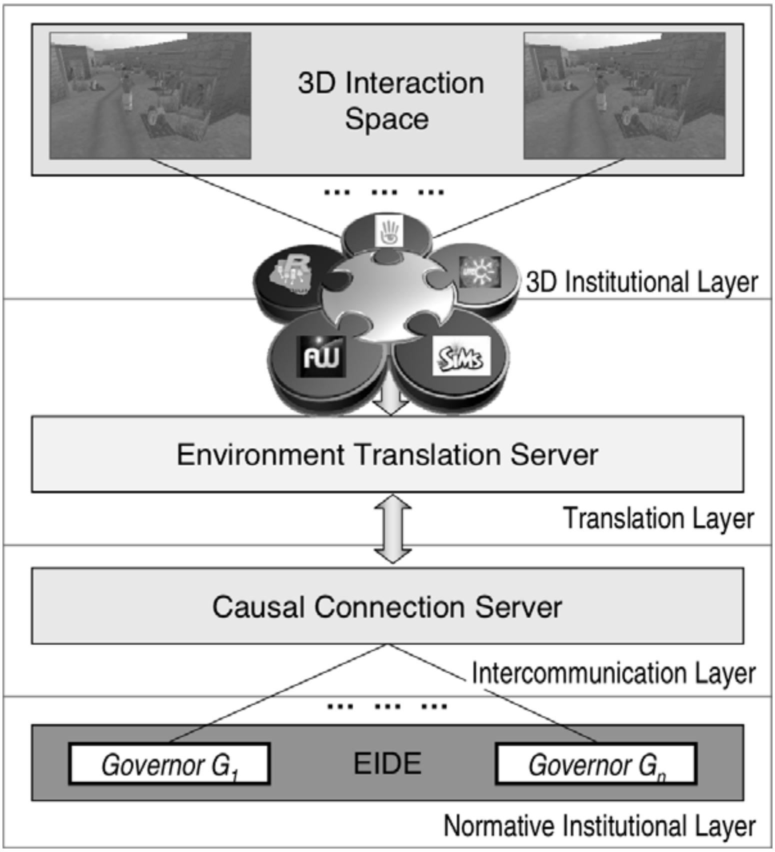


Fig. 3. The extended architecture of virtual institution technology

3 Market Players

3.1 Acting Convincingly

It is one thing for an agent to appear to be convincing, or to move in a convincing way, but another for it to interact to *interact* in a convincing way. For an agent's utterances to be convincing it must act in a way that demonstrates that it understands:

- the significance of each of its utterances *to the observer*
- that on-going interaction are seen as relationships with the other agents that carry implied *social obligations* to act appropriately
- what it should *not* do

A formal model is described that addressed these issues that is based on the observation that an agents beliefs and understanding are necessarily uncertain. The description is from the point of view of agent α and interacts with agent β.

The significance of utterances. If β passes an utterance to α, α evaluates this act in two ways. First, it is valued for the strategic significance of the information that it contains, precisely it is measured as the expected increase in utility that α expects to enjoy given that it has the information — this is the *utilitarian measure*. Second, it is valued because the sending agent *was prepared to divulge* the information in the utterance, precisely it is measured as the decrease in uncertainty that the receiving agent has over the sending agent's private information — this is the *information measure*. All utterances received are qualified by α with a belief probability as described in [27].

From α's point of view, β's *private information* is everything that β knows and that α does not know with certainty. Due to the persistent effect of integrity decay this will include much of what β knows.

An agent may wish to decide which action, $\{a_i\}$, to take where the payoff depends on which state, $\{s_j\}$, the world is in when the action is taken (possibly in the future). The payoff, $\boldsymbol{v_i}$, from taking action a_i is a vector where v_{ij} will be the payoff from taking action a_i and the state of the world is s_j. Let $\boldsymbol{p}$ be the probability mass function of a random variable representing the prior expectation about the state of the world when the action is taken. Then the *expected monetary value* gained by choosing action a_i is $m_i = \boldsymbol{p} \cdot \boldsymbol{v_i}$.

Armed with this information suppose that the agent applies some decision criterion, c, to decide what to do — perhaps c will choose the action with the greatest expected payoff: $\arg\max_i \boldsymbol{p} \cdot \boldsymbol{v_i}$. Now suppose that the agent receives information in an utterance u that enables him to refine his expectation of the state of the world when the action is to be taken $(\boldsymbol{p}|u)$, and that he applies the same criterion c. Then one *utilitarian value* of utterance u to criterion c is the difference between the payoffs of the respective outcomes. For each state of the world s_j let $b_j = \max_i v_{ij}$ i.e. b_j is the 'best' action that the agent can take if the state of the world is s_j then the *expected value of perfect information* is

$$\boldsymbol{p} \cdot \boldsymbol{b} - \max_i \boldsymbol{p} \cdot \boldsymbol{v_i}$$

this is an upper limit on the total value of all possible utterances with respect to the application of criterion c.

Utilitarian measures of information are expressed in terms of: if you know information x when applying criterion y to determine which action to perform then you will gain utility z over not knowing x [26]. That is, they are defined in the context of some decision making act — they do not place an intrinsic value on information. So if an agent learns x at time t and is unaware of what future decisions he will make that will benefit from knowing x, then he will be unable to value x until he knows what those future decisions are. But, by the time he is aware of all of those decisions it may not be possible to reconstruct with certainty how he and the other agents would have behaved if he had *not* known x at time t. In summary, it is only possible to attach an intrinsic utilitarian value to information when the future decisions that are relevant to it are known.

We have described the value *gained* by acquiring information, we now consider the value *lost* by an agent's private information becoming public knowledge — that is, known to all agents in the system. Once information becomes public knowledge it has no tradable value until the integrity of the public's belief of it decays in time.

Utilitarian measures of information may be used when all the relevant future decisions are either known with certainty or a probability distribution expressing their likeliness to occur is known.

α's world model, $\mathcal{M}^t$, is a set of probability distributions. If at time t, α receives an utterance u that may alter this world model then the (Shannon) *information* in u with respect to the distributions in $\mathcal{M}^t$ is: $\mathbb{I}(u) = \mathbb{H}(\mathcal{M}^t) - \mathbb{H}(\mathcal{M}^{t+1})$. Let $\mathcal{N}^t \subseteq \mathcal{M}^t$ be α's model of agent β. If β sends the utterance u to α then the *information* about β within u is: $\mathbb{H}(\mathcal{N}^t) - \mathbb{H}(\mathcal{N}^{t+1})$. We give structure to the measurement of information using an *illocutionary framework* to categorise utterances, and an *ontology*.

The illocutionary framework will depend on the nature of the interactions between the agents. The LOGIC framework for argumentative negotiation [27] is based on five categories: Legitimacy of the arguments, Options i.e. deals that are acceptable, Goals i.e. motivation for the negotiation, Independence i.e: outside options, and Commitments that the agent has including its assets. The LOGIC framework contains two models per agent: first α's model of β's private information, and second, α's model of the private information that β has about α. Generally we assume that α has an illocutionary framework $\mathcal{F}$ and a categorising function $v : U \to \mathcal{P}(\mathcal{F})$ where U is the set of utterances. The power set, $\mathcal{P}(\mathcal{F})$, is required as some utterances belong to multiple categories. For example, in the LOGIC framework the utterance "I will not pay more for Protos[1] than the price that John charges" is categorised as both Option and Independence.

We assume an ontology, and $\mathcal{O}$ denotes its concepts that are organised in an *is-a* hierarchy.[2] δ measures the semantic distance between two concepts c_1 and c_2, for example [28]:

$$\delta(c_1, c_2) = e^{-\kappa_1 l} \cdot \frac{e^{\kappa_2 h} - e^{-\kappa_2 h}}{e^{\kappa_2 h} + e^{-\kappa_2 h}}$$

where l is the shortest path between the concepts, h is the depth of the deepest concept subsuming both concepts, and κ_1 and κ_2 are parameters scaling the contribution of shortest path length and depth respectively.

Acting to Respect Social Obligations. In [27] two central concepts are used to describe relationships and dialogues between a pair of agents. These are *intimacy* — degree of closeness, and *balance* — degree of fairness. Both of these concepts are summary measures of relationships and dialogues, and are expressed in the LOGIC framework as 5 × 2 matrices.

More generally, the intimacy of α's relationship with β_i, I_i^t, measures the amount that α knows about β_i's private information and is represented as real

[1] A fine wine from the 'Ribera del Duero' region, Spain.

[2] A simplified way of understanding an utterance u is as a set of concepts in $\mathcal{O}$, that is $u = \{c_i \mid c_i \in \mathcal{O}\}$.

numeric values over $\mathcal{G} = \mathcal{F} \times \mathcal{O}$. Suppose α receives utterance u from β_i and that category $f \in v(u)$. For any concept $c \in \mathcal{O}$, define $\Delta(u, c) = \max_{c' \in u} \delta(c', c)$. Denote the value of I_i^t in position (f, c) by $I_{i(f,c)}^t$ then:

$$I_{i(f,c)}^t = \rho \times I_{i(f,c)}^{t-1} + (1 - \rho) \times \mathbb{I}(u) \times \Delta(u, c)$$

for any c, where ρ is the discount rate. The *balance* of α's relationship with β_i, B_i^t, is the element by element numeric difference of I_i^t and α's estimate of β_i's intimacy on α.

[29] describes measures of: *trust* (in the execution of contracts), *honour* (validity of argumentation), and *reliability* (of information). The execution of contracts, soundness of argumentation and correctness of information are all represented as conditional probabilities $\mathbb{P}(\varphi'|\varphi)$ where φ is an expectation of what may occur, and φ' is the subsequent observation of what does occur.

[20] describes a single computational framework for these three measures that summarise α's observations of β's behaviour. One of these summary measures is:

$$M(\alpha, \beta, \varphi) = 1 - \sum_{\varphi'} \mathbb{P}_I^t(\varphi'|\varphi, e) \log \frac{\mathbb{P}_I^t(\varphi'|\varphi, e)}{\mathbb{P}^t(\varphi'|\varphi)}$$

where the "1" is an arbitrarily chosen constant being the maximum value that this measure may have, and $\mathbb{P}_I^t(\varphi'|\varphi, e)$ is a distribution of enactments that represent α's "ideal" in the sense that it is the best that α could reasonably expect to happen in *context* e. If α repeatedly observes φ' then the amount of information that those observations convey about the associated commitments, φ, is the *mutual information*: $\mathbb{I}(\varphi'; \varphi) = \mathbb{H}(\varphi') - \mathbb{H}(\varphi'|\varphi)$, this measures the mutual dependence of the two variables, where $\mathbb{I}(\varphi'; \varphi) = \mathbb{I}(\varphi; \varphi')$.

These summary measures are all abstracted using the ontology; for example, "What is my trust of John for the supply of red wine?". These measures are also used to summarise the information in some of the categories in the illocutionary framework. For example, if these measures are used to summarise estimates $\mathbb{P}^t(\varphi'|\varphi)$ where φ is a deep motivation of β's (i.e. a Goal), or a summary of β's financial situation (i.e. a Commitment) then this contributes to a sense of trust at a deep social level.

Knowing what *not* to do. [27] advocates the controlled revelation of information as a way of managing the intensity of relationships. In Section 3.1 we noted that information that becomes public knowledge is worthless, and so respect of confidentiality is vital to maintaining the value of revealed private information. We have not yet described how to measure the extent to which one agent respects the confidentiality of another agent's information — that is, the strength of belief that another agent will respect the confidentially of my information: both by not passing it on, and by not using it so as to disadvantage me.

Consider the motivating example, α sells a case of Protos to β at cost, and asks β to treat the deal in confidence. Moments later another agent β' asks α to quote on a case of Protos — α might then reasonably increase his belief in the proposition that β had spoken to β'. Suppose further that α quotes β' a fair market price for the Protos and that β' rejects the offer — α may decide to further increase this belief. Moments later β offers to purchase another case of Protos for the same cost. α may then believe that β may have struck a deal with β' over the possibility of a cheap case of Protos.

Confidentiality is the mirror image of trust, honour and reliability that are all built by an agent "doing the right thing" — respect for confidentiality is built by an agent *not* doing the wrong thing. As human experience shows, validating respect for confidentiality is a tricky business. One proactive ploy is to start a false rumour (e.g. "My wife is a matador.") and to observe how it spreads. The following reactive approach builds on the Protos example above.

An agent will know when it passes confidential information to another, and it is reasonable to assume that the significance of the act of passing it on decreases in time. In this simple model we do not attempt to value the information passed as in Section 3.1. We simply note the amount of confidential information passed and observe any indications of a breach of confidence.

If α sends utterance u to β "in confidence", then u is categorised as f as described in Section 3.1. C_i^t measures the amount of confidential information that α passes to β_i in a similar way to the intimacy measure I_i^t described in Section 3.1:

$$C_{i(f,c)}^t = \rho \times C_{i(f,c)}^{t-1} + (1 - \rho) \times \mathit{\Delta}(u, c)$$

for any c where ρ is the discount rate; if no information is passed at time t then:

$$C_{i(f,c)}^t = \rho \times C_{i(f,c)}^{t-1}$$

C_i^t represents the time-discounted amount of confidential information passed in the various categories.

α constructs a companion framework to C_i^t, L_i^t is as estimate of the amount of information leaked by β_i represented in $\mathcal{G}$. Having confided u in β_i, α designs update functions J_u^L for the L_i^t. In the absence of evidence imported by the J_u^L functions, each value in L_i^t decays by:

$$L_{i(f,c)}^t = \xi \times L_{i(f,c)}^{t-1}$$

where ξ is in $[0, 1]$ and probably close to 1. The J_u^L functions scan every observable utterance, u', from each agent β' for evidence of leaking the information u, $J_u^L(u') = \mathbb{P}(\beta' \text{ knows } u \mid u' \text{ is observed})$. As previously:

$$L_{i(f,c)}^t = \xi \times L_{i(f,c)}^{t-1} + (1 - \xi) \times J_u^L(u') \times \mathit{\Delta}(u, c)$$

for any c.

This simple model estimates C_i^t the amount of confidential information passed, and L_i^t the amount of presumed leaked, confidential information represented over $\mathcal{G}$. As with most things that information-based agents do, the 'magic' is in the specification of the J_u^L functions. A more exotic model would estimate "who trusts who more than who with what information" — this is what we have elsewhere referred to as a *trust network*. The feasibility of modelling a trust network depends substantially on how much detail each agent can observe in the interactions between other agents.

4 Conclusions

The evolution of the electronic market places is an intrinsic part of the evolution of the Web, hence, it will be essential in Web 3.0 technology. When Web 2.0 is centered around humans engagement, interaction and sharing, we view the forthcoming Web 3.0 to be about placing humans "within" an intelligently behaving Web. Consequently, the form of realism, discussed in this paper, is essential to Web 3.0. Central to this notion is believable agent behaviour, including the smart ways of gaining advantage from being well informed and the ability to utilise relevant information. We also discussed the enabling technology. At the end we would like to sum up using Gary Kasparov's quotation "I sensed an alien intelligence in the program." after the 1997 defeat of the world chess champion by the computer program Deep Blue II (as quoted in [30]).

References

1. Mäkiö, J., Weber, I., Weinhardt, C.: Electronic negotiations — a generic approach with action systems. In: Bauknecht, K., Bichler, M., Pröll, B. (eds.) EC-Web 2004. LNCS, vol. 3182, pp. 135–143. Springer, Heidelberg (2004)
2. Bakos, J.: A strategic analysis of electronic marketplaces. MIS Quarterly 15(3), 295–310 (1991)
3. Voudouris, C., Owusu, G., Dorne, R., Lesaint, D.: Service Chain Management: Technology Innovation for the Service Business. Springer, Heidelberg (2008)
4. Debenham, J., Simoff, S.: An e-Market Framework for Informed Trading. In: Carr, L., Roure, D.D., Iyengar, A., Goble, C., Dahlin, M. (eds.) Proceedings 15th International World Wide Web Conference, WWW-2006, Edinburgh, Scotland (May 2006)
5. Smith, V.: Markets, institutions and experiments. In: Nadel, L. (ed.) Encyclopedia of Cognitive Science. Nature Publishing Group, New York (2003)
6. Magnenat-Thalmann, N., Kim, H., Egges, A., Garchery, S.: Believability and interaction in virtual worlds. In: Proceedings of the 11th International Multimedia Modelling Conference (MMM'05), pp. 2–9 (2005)
7. Tomlinson, B., Yau, M.L., Baumer, E.: Believable agents: Embodied mobile agents. In: Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems AAMAS'06, Hakodate, Japan, pp. 969–976 (2006)
8. Maatman, R.M., Gratch, J., Marsella, S.: Natural behavior of a listening agent. In: Panayiotopoulos, T., Gratch, J., Aylett, R., Ballin, D., Olivier, P., Rist, T. (eds.) IVA 2005. LNCS (LNAI), vol. 3661, pp. 25–36. Springer, Heidelberg (2005)
9. Loyall, A.B.: Believable Agents: Building Interactive Personalities. PhD thesis, Department of Computer Science, Carnegie Mellon University, Pittsburgh (1997)
10. Bogdanovych, A.: Virtual Institutions. PhD thesis, Faculty of IT, University of Technology, Sydney (November 2007)
11. O'Keefe, J., Nadel, L.: The Hippocampus as a cognitive map. Oxford University Press, Oxford (1978)
12. Alpcan, T., Bauckhage, C., Kotsovinos, E.: Towards 3d internet: Why, what, and how? In: Proceedings of the International Conference on Cyberworlds, pp. 95–99 (2007)

13. Favalora, G.: Volumetric 3d displays and application infrastructure. IEEE Computer, 37–44 (August 2005)
14. Boella, G., der Torre, L.V., Verhagen, H.: Introduction to normative multiagent systems. Computational Mathematics and Organisational Theory 12, 71–79 (2006)
15. EIDE: Electronic institution development environment (2005), http://e-institutor.iiia.csic.es/
16. Web3D Consortium: X3d international specification standards (2009), http://www.web3d.org/
17. Linden Lab.: Second life (2009), http://secondlife.com/
18. Arcos, J.L., Esteva, M., Noriega, P., Rodríguez, J.A., Sierra, C.: Environment engineering for multiagent systems. Journal on Engineering Applications of Artificial Intelligence 18 (2005)
19. MacKay, D.: Information Theory, Inference and Learning Algorithms. Cambridge University Press, Cambridge (2003)
20. Sierra, C., Debenham, J.: Information-based agency. In: Proceedings of Twentieth International Joint Conference on Artificial Intelligence IJCAI-07, Hyderabad, India, January 2007, pp. 1513–1518 (2007)
21. Jaynes, E.: Information theory and statistical mechanics: Part I. Physical Review 106, 620–630 (1957)
22. Golan, A., Judge, G., Miller, D.: Maximum Entropy Econometrics: Robust Estimation with Limited Data. In: Financial Economics and Quantitative Analysis. John Wiley and Sons, Inc., Chichester (1996)
23. Paris, J.: Common sense and maximum entropy. Synthese 117(1), 75–93 (1999)
24. Jaeger, M.: Representation independence of nonmonotonic inference relations. In: Proceedings of KR'96, pp. 461–472. Morgan Kaufmann, San Francisco (1996)
25. Halpern, J.: Reasoning about Uncertainty. MIT Press, Cambridge (2003)
26. Lawrence, D.: The Economic Value of Information. Springer, Heidelberg (1999)
27. Sierra, C., Debenham, J.: The LOGIC Negotiation Model. In: Proceedings Sixth International Conference on Autonomous Agents and Multi Agent Systems AAMAS-2007, Honolulu, Hawai'i, May 2007, pp. 1026–1033 (2007)
28. Li, Y., Bandar, Z.A., McLean, D.: An approach for measuring semantic similarity between words using multiple information sources. IEEE Transactions on Knowledge and Data Engineering 15(4), 871–882 (2003)
29. Sierra, C., Debenham, J.: Trust and honour in information-based agency. In: Stone, P., Weiss, G. (eds.) Proceedings Fifth International Conference on Autonomous Agents and Multi Agent Systems AAMAS-2006, Hakodate, Japan, May 2006, pp. 1225–1232. ACM Press, New York (2006)
30. Cross, N.: Designerly Ways of Knowing. Springer, Heidelberg (2006)

On the Benefits of Keyword Spreading in Sponsored Search Auctions: An Experimental Analysis

Michele Budinich[1], Bruno Codenotti[2], Filippo Geraci[2], and Marco Pellegrini[2]

[1] Institute for Advanced Studies, Lucca, Italy
m.budinich@imtlucca.it
[2] Istituto di Informatica e Telematica, CNR – Consiglio Nazionale delle Ricerche,
56100 Pisa, Italy
{bruno.codenotti,filippo.geraci,marco.pellegrini}@iit.cnr.it

Abstract. Sellers of goods or services wishing to participate in sponsored search auctions (SSA) must define a pool of keywords that are matched on-line to the queries submitted by the users to a search engine. Sellers must also define the value of their bid to the search engine for showing their advertisements in case of a query-keyword match. In order to optimize its revenue a seller might decide to substitute a keyword with a high cost, thus likely to be the object of intense competition, with sets of related keywords that collectively have lower cost while capturing an equivalent volume of user clicks. This technique is called *keyword spreading* and has recently attracted the attention of several researchers in the area of sponsored search auctions. In this paper we describe an experimental benchmark that, through large scale realistic simulations, allows us to pin-point the potential benefits/drawbacks of keyword spreading for the players using this technique, for those not using it, and for the search engine itself. Experimental results reveal that keyword spreading is generally convenient (or non-damaging) to all parties involved.

1 Introduction

A very large fraction of consumers use search engines to find information on the web about goods and services before deciding whether to purchase them in the online markets. Search engines take advantage of their key position on the Web to sell advertising space to economic players on search result pages. Indeed, over the last few years, sponsored search advertising has become the dominant source of profits for search engines. Typically sponsored search results appear in two separate parts of the page above and to the right of the results returned by a search engine. Sponsored search results include a title, a short text, and a link referring to a Website. Advertising space comes in the form of slots, which are sold by auctions. When a user submits a given keyword in a query to a search engine, an auction is run among all the advertisers submitting bids for that keyword. The advertisers who wish to display their ads against the search for a keyword participate in the auction by specifying their valuation and a daily

F. Buccafurri and G. Semeraro (Eds.): EC-Web 2010, LNBIP 61, pp. 158–171, 2010.

budget to the search engine. The search engine could use various mechanisms for determining winners and payments, the most popular mechanism being the generalized second price (GSP) auction.

Although GSP looks similar to the classical Vickrey-Clarke-Groves (VCG) mechanism [31,10,18], its properties are very different, i.e., truth-telling is not an equilibrium in GSP [12,30]. Over the last years, several papers of computational flavor have appeared, touching in different ways this paradigm of online advertising, see, e.g., [5,6,12,20,21]. From the viewpoint of a search engine, the *adword problem* consists of assigning a sequence of search keywords to a set of competing bidders, each with a daily spending limit, with the goal of maximizing the revenue generated by these keyword sales. This problem generalizes on-line matching, and this connection has been exploited in [23]. A central problem in adword markets from the point of view of a seller of goods and services is the generation of keywords. Advertisers typically prefer to bid for keywords that have high search volumes; however they may be very expensive, so that it might be reasonable to bid instead for several related and low volume, inexpensive terms that generate roughly the same amount of traffic altogether. Some preliminary work exploring this idea has been done in [1], where however, the emphasis is on the algorithmic aspects of keyword generation, not on the global market phenomena as in the present work.

In this paper we describe a large scale simulator for analyzing the effect of using synonyms for keyword spreading in sponsored search auctions (SSA), and collect a number of evidences about the effects of this strategy. Our simulations involve up to 2M agents bidding for words from a pool of 36K words and 3M queries per experiment (more details in Sections 2 and 3). Our experiments point to the following conclusions:

– using synonyms increases the revenues for all players in the market (Figs. 6a, 7a, 7b); in particular the early adopter agents benefit the most (Figs. 7a, 7b)
– using a VCG payment scheme decreases the agents' benefits with respect to using GSP while not much changes for the search engine (data omitted for lack of space, see [8])
– as the fraction of agents using synonyms increases, the search engine revenues are not significantly affected (Fig. 6b) as well as the costs for the agents not using them (Fig. 8a) while the agents using synonyms have decreasing gains (Fig. 8b)
– budget depletion strategies are shown to rarely be beneficial for the agents, while always increasing the revenue for the search engine, even in presence of keyword spreading (data omitted for lack of space, see [8]).

A problem related to *keyword spreading* is that of *keyword selection* [29], where the economic players try to select at fixed rounds the subset of keywords that maximize revenues while trying to learn basic parameters (such as keyword click-through rates) during the repeated bidding processes. Note that here the viewpoint is that of a single player and that the market, as seen by the seller, is modeled via (known or unknown) time varying probability distributions. In contrast, in our simulations keywords are selected by the agents off-line. We simulate directly the market and the auctions by using a large number of atomic agents each performing simple actions. Previous research on agent-based simulation of adwords markets by Mizuta and Steiglitz [24] was centered on studying the interaction of different classes of players according to their bidding time profiles, e.g. early vs late bidders. Kitts and LeBlanc [19] describe a large scale simulator for adwords markets to investigate several bidding strategies, e.g. random bidding vs. bid to keep relative position, which however do not involve keyword spreading. The architecture of a large scale SSA is described in [4], where it is applied to compare several ranking, pricing and budgeting policies. To the best of our knowledge this is the first large-scale agent-based simulation of the market effects of keyword spreading. The time horizon of our simulations is one bidding day. For this reason we consider as fixed all features whose rate of change is so slow so that it can be approximated by a constant within the time span of one day (e.g. the number of bidder is fixed at the beginning of the day, and their number decreases only when they run out of budget). Other features that can vary with a faster dynamic are modeled as distributions (or with an adaptive behavior), but the parameters of the distribution itself are considered as having a much slower dynamics, therefore such parameters are fixed within the one day time frame.

The rest of this paper is organized as follows. In Section 2 we briefly describe the clustering technique used to build a dictionary of synonyms. In Section 3 we highlight the architecture of our market simulator. In Section 4 we show the main outcomes of our experiments.

2 Keyword Spreading

We explore two alternative ways of performing keyword spreading. One uses the well known Wordnet ontology, the second is based on clustering web pages related to a query as found by a generalist search engine (in our case Google). The two resulting word distributions are different but the measured trends are consistent for both data sets, thus giving high confidence in the robustness of the experimental benchmark.

Wordnet. The most important project for ontologies of words is *WordNet* [26]. Originally proposed by the Cognitive Science Laboratory at Princeton University only for the English language, WordNet has become a reference for the whole information retrieval community, and similar projects are now available in many other languages. WordNet is a handmade semantic lexicon that groups words into sets of synonyms called *synsets*. Intuitively one can replace a word in a text with another from the same synset without changing its semantics. A word can appear in more than one synset if it has more than one meaning. Synsets are arranged as nodes in a graph such that there is an edge to connect two nodes if there is a relation between the two synsets. There are different types of possible relations, an exhaustive list of them can be found in the WordNet web site [25]. Given two synsets X and Y, the most common types of relations in WordNet are: *hypernym* if every X is a “kind of” Y, *hyponym* if Y is a “kind of” X, *holonym* if X is a part of Y and *meronym* if Y is a part of X. In our experiments we took into account only hypernym. Note that this relation is *asymmetric.*

Clustering Google Data. Given a query word, our goal is to find a set of semantically related words whose cost is lower than those of the query. We are not only interested in paradigmatic similarity, i.e., when two words may be mutually exchanged without effects on the semantics of the text, but also to syntagmatic similarity, i.e., when two words significantly co-occur in the same context. To achieve this goal we approach the problem as a *word clustering* [11,22] task. Given a set of objects, clustering attempts to create a partition such that the objects in a cluster are related among them, while objects in different clusters are unrelated. Word clustering requires a corpus of documents related to the query word. To set up such a corpus we redirect the query to *Google* and download pages related to the first 100 results. Each page is later parsed and split to extract a set of sentences. Under the well established hypothesis that co-related words are more likely to stand in the same sentence, all the sentences not containing the query are discarded. We remove from each sentence over-represented words (stop words) that are often “syntactic sugar” and their removal does not affect the semantic content of the sentence. We added to the standard stop word list, a set of words that normally can not be considered stop words, but in the Web environment are considered generic (e.g. “download”). Once filtered, all the sentences are arranged in a term-document matrix whose rows correspond to sentences and whose columns to terms of the corpus. We tested different weighting schemes for terms, and we found that for our purpose a simple binary weighting scheme suffice. For clustering we employed a fast implementation of the FPF clustering algorithm [16] because of its good trade off between speed and accuracy [14]. As distance between pairs of words, i.e., columns of the term-document matrix, we used the well known cosine similarity. FPF is an iterative algorithm. It makes a new cluster at each iteration and populates it by extracting from the other clusters all the elements that are more related to the new cluster. The procedure stops when a given number k of clusters is reached. For word clustering it is impossible to predict in advance a good value for k. The typical approach, with methods such as k-means, is to make a certain number of independent clusterings with different choices of k and select the most appropriate a posteriori. Instead, the iterative nature of FPF allows us to not feeding the number of clusters in advance but check a more appropriate termination condition at each iteration. In our case, at the end of iteration t, FPF checks the cluster $C_t(q)$ containing the query. When the number of elements of this cluster gets below a certain threshold (10 in our case) the algorithm stops and returns, among $C_t(q)$ and $C_{t-1}(q)$, the set whose cardinality is closest to the threshold. This procedure ensures that we find a coherent cluster of words even if the query is not central in that cluster. Note that this relation may be *asymmetric*, although in a subtle way, since different sets of snippets are processed for each query word. Stretching the terminology, we will call words for which the relations defined above hold (over WordNet and Google data) “synonyms”, for lack of a better name, however one should keep in mind that the relations we model is more complex.

3 The Simulator

The starting point in designing the simulator was the collection of some publicly available data on ad auctions, including:

- a large and representative set of words,
- an estimate of the cost of each word,
- an estimate of the number of clicks received by each word.

The Word List. The simulator uses a finite set of words; these words represent all the possible queries that a user can make to the search engine and also all the possible keywords an advertiser can bid on. The core of the word list has been taken from the SCOWL (`http://wordlist.sourceforge.net/`) project (an open source project that maintains a set of word lists for use by spell checkers), and consists of 35867 entries.

The Traffic Estimator. Google maintains an on-line tool (the AdWords Traffic Estimator Sandbox,[1]) developed to aid advertisers in their campaigns. The Traffic Estimator, given a keyword, displays its estimated cost per click (CPC) and the estimated number of clicks per day. The simulator uses this data to estimate some quantities that would otherwise be difficult to generate realistically. Although, as Google itself warns, the data is to be considered only as a guideline, it is of great help for our purposes. The estimated CPC is used in the simulator (averaging the two values given by the Traffic Estimator) as a basis to assign a “real” value to each keyword. The simulator successively employs these values as parameters to generate the agents’ bids and valuations. Clearly the estimated CPC of a term is different from its “real” value. If we were to measure the estimated CPCs in the simulator at the end of a run they would certainly be different from the ones supplied by the Traffic Estimator. Nonetheless their distributions and main features would be similar, and that is enough for the use we make of it. The other parameter that is central to the simulator is the estimated number of clicks per day of each word. Since the simulation considers only the queries that give rise to a click, we can simply consider the estimated number of clicks per day as the distribution of the queries in the simulator. We collected such data for each of the 35867 entries in our dictionary, building a small database that constitutes our initial data set. Table 1a summarizes the main characteristics of the data set. For completeness, we plotted the data collected from Google’s Traffic Estimator. Figure 1a is the distribution of the estimated clicks per day, while Figure 1b shows how estimated average costs per click are distributed.

[1] `https://adwords.google.com/select/TrafficEstimatorSandbox`

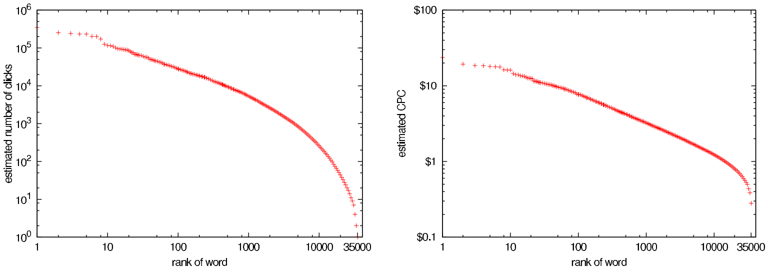
Table 1. Statistics of the click and synonym databases

Number of words	35867
Max. clicks per day	349216
Min. clicks per day	1
Max. CPC	$23.6
Min. CPC	$0.05

(a) Statistics from the CPC and Click Volumes

	clustering	Wordnet
Words with synonyms	18660	12271
Max. synonyms for a word	13	441
Max. terms a word is synonym of	668	146

(b) Statistics from the two synonyms databases: Wordnet and clustering of Google snippets



(a) Estimated number of clicks

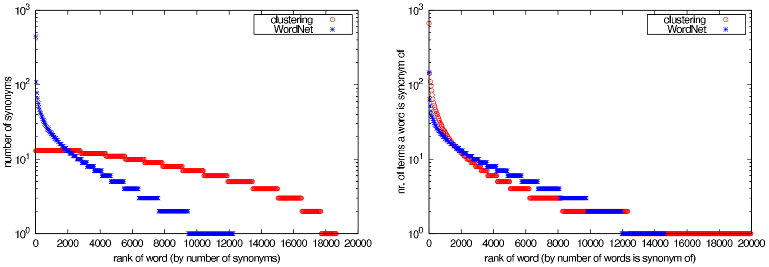
(b) Estimated CPC

Fig. 1. Data gathered from Google's Traffic Estimator, in log-log scale

We want to use our simulator to investigate the behavior of the ad auction mechanism in the presence of agents who make use of keyword spreading. To model such behaviors we need a set of synonyms for each word. The clustering algorithms described in Section 2 produced a list of synonyms for each word. As a reference we have also created a similar list by querying the Wordnet[2] database. Table 1b gives some basic figures on the two resulting data sets, while Figure 2a and Figure 2b show the distribution of the number of synonyms per word and the distribution of the number of terms a word is synonym of.

There is a big difference in the boundary values of the databases: for example, there is a term for which Wordnet gives 441 synonyms; but more important is the difference in the rank distribution (see Fig. 2a). Due to limitations in the computational resources, the clustering imposed a hard limit of 13 on the maximum number of synonyms per word. Nonetheless, as shown clearly by Figure 2a, the majority of the words have more synonyms in the clustering database than in the Wordnet one. Overall we can consider the databases comparable for our purposes, and the experimentally detected trends are consistent in both databases. Starting from a list of words, we have expanded it with various

[2] http://wordnet.princeton.edu/



(a) Number of synonyms per word

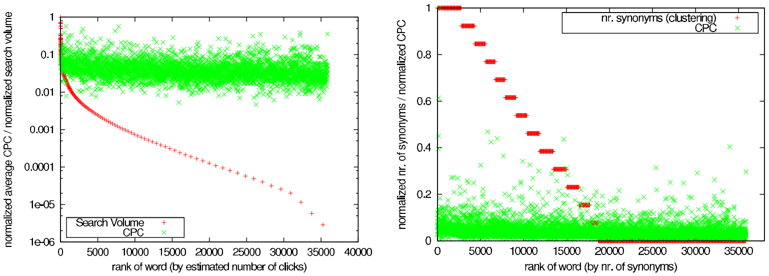
(b) Number of words a term is a synonym of

Fig. 2. Comparison of the two synonyms databases in log-scale

information: prices, number of clicks and synonyms. It seems now a natural question to ask if there is any correlation between these quantities. As a first guess it might seem reasonable to expect at least some correlation. That is, we might expect that some "popular" words receive many clicks and have a high price. Or that words that receive a lot of clicks also happen to have many synonyms. Somewhat surprisingly, an empirical analysis gives a negative result. At a first glance the data set exhibits virtually no correlation between the different values. To give a rough idea of this result we present just two plots, all the other ones being extremely similar. Figure 3a ranks words by estimated number of clicks, and shows these values along the CPCs (both normalized). It looks like there is no order in the CPC values; they appear as if uniformly distributed.

Figure 3b, instead, ranks the words by the number of synonyms they possess, using the Wordnet database, and displays this value along the estimated CPC (again, normalized). As it seems apparent there is no correlation between these quantities. All the other comparisons, e.g. CPCs versus click volume, synonyms versus CPC using the clustering database, give similar results.

All the simulations were carried out using the same static set of agents. To this end, the set of agents was generated once and for all and saved in a file. Its



(a) nr. of clicks and CPC (log-scale)

(b) nr. of synonyms and CPC

Fig. 3. Comparison of the two synonyms databases used

Table 2. Statistics for the bidding agents and bids on a typical keyword

Number of Agents	$2 \cdot 10^6$
Max. bids on a single word	21446
Min. bids on a single word	21
Max. bids per agent	3000
Min. bids per agent	3
Budget Range	[\$1 – \$max. avail.]
Bid Range	[\$0.01 – \$200]
Nr. of slots	4
Clickthrough probabilities	$0.6, 0.25, 0.10, 0.05$

(a) Statistics on the pool of bidding agents

Keyword	"reviews"
"Real" value	\$1.045
Estimated nr. of clicks	12029
Nr. of interested agents	11023
Max. bid	\$3.064
Min. bid	\$0.010
Max. difference $v_i - b_i$	12.5% of v_i

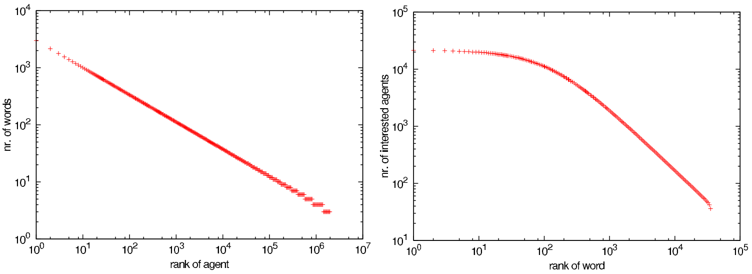
(b) Statistics on the bids for the keyword "review"

main characteristics are presented in Table 2a. In what follows we will refer to this fixed set of agents, words and synonyms as our data set.

Each agent bids on a number of different keywords. If we consider all the agents, these numbers of allocated keywords are distributed as a power law, whose parameters are based on the number of agents, such as to keep a fixed maximum and minimum (to avoid cases in which an agent bids on all of the words, or cases in which there are agents that have not bid on any word at all). Figure 4a plots these values for the data set. The choice of a power law distribution to model the keyword-to-agent distribution is justified by an analogy with real data in the version 1.0 of *Yahoo! Search Marketing advertising bidding data*[3], used also in [7]. The distribution described in [7] fits qualitatively a power law. The real data come for an anonymised log of bids for 1000 keywords with about 10,000 bidders collected in the period 2002/2003, where data was truncated at 50 keywords-per-agent. In order to perform a larger simulation ($2 \cdot 10^6$ bidders, $36 \cdot 10^3$ keywords) we have correspondingly scaled up the power-law curve so to have the number of words-per-agents in a range from a few units to about two thousand.

Another quantity characterizing agents is their budget. We have chosen a uniform distribution with budgets in the range [\$1 – \$100]. The choice of a uniform distribution for the budget-to-agent distribution comes from a series of rather indirect arguments. We could not find any such distribution described in literature, or in publicly available data sets, probably due to the sensitive nature of such data. A few papers that use such a distribution in simulation (e.g. [13] [3]) give no clue as to its shape. Anecdotic remarks [15] report typical budgets are in the orders of hundreds of dollars. A theory of sponsored search auctions for markets with budget constraints has been developed in recent years, and often the budget distribution among bidders is left as a free parameter of

[3] `http://webscope.sandbox.yahoo.com/`



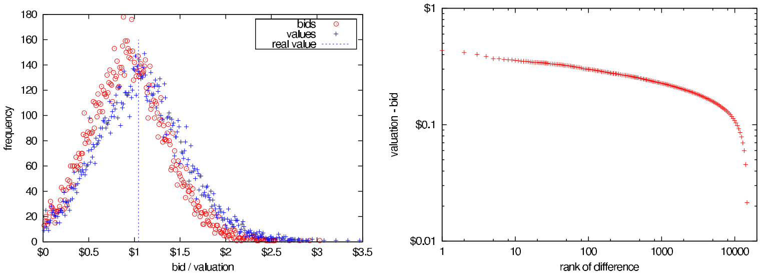
(a) words per agent

(b) agents per word

Fig. 4. Agents per word and words per agent in log-log scale

the theory. An interesting paper of Z. Abrams [2] describes a theory for revenue maximization where a critical parameter is the *bidder budget dominance*, that is the fraction of the total market budget that is allocated to the bidder with the highest budget. The use of a uniform budget distribution in the range $[1, .., 100]$ is consistent with the above considerations. While the ratio of the highest to the lowest bidder can be as high as 100, the bidder budget dominance is very low (at most 10^{-6}). Experiments with a power law distribution give almost identical results. We conjecture that the results we present are qualitatively analogous for any other distribution that has a similar low budget dominance and maximum budget ratio, even if not uniform.

Words. The words come from an open-source dictionary created for the spell-checkers. To each entry we have added the following two pieces of information: estimated number of clicks, and estimated CPC, which we obtained from the tool made available by Google. Given the dictionary, we assign the words to the bidders in such a way that both the number of bidders per keyword and the number of keywords per bidder be distributed according to a power-law. Having fixed the number of words an agent will bid on, the next step is to select them from the dictionary. The simulator does so, and the resulting values (i.e., the number of agents interested in every word) are again distributed as a (different) power law. The parameters controlling such distribution are chosen as to avoid unrealistic scenarios. Figure 4b shows the number of interested agents per word in our data set. As described above, each word is assigned a "real" value based on the data gathered from the Traffic Estimator. Based on this reference value, each agent i will then compute its personal valuation v_i for the keyword. The distribution of the valuations for each agent is a normal distribution whose mean is precisely the "real" value of the word. To increase the variety among agents, each agent has a different variance associated to this normal distribution. Figure 5a shows the distribution of valuations for different agents interested in the same keyword, i.e. "reviews". The agent valuation v_i represents the agents's *Return on Investment* for a click of his advertisement, and it is a private information not disclosed to any of the partners in the auction (either SE or other players), thus difficult to infer from any collected data set where only the bids are known.



(a) Bids and valuations (b) Difference valuations - bids

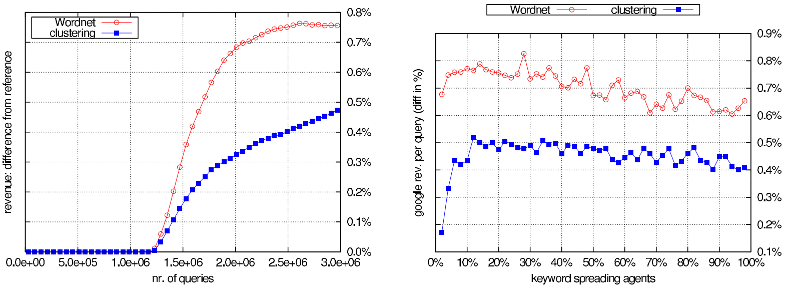
Fig. 5. The bids and valuations for the word “review”, whose “real” value is 1.045

As a final step each agent i must generate a bid b_i. Bids are generated according to the agent's valuation v_i. Only bids such that $b_i < v_i$ are considered, and such that they do not exceed the residual budget of agent i. The quantity $v_i - b_i$ for a given keyword and agent i is proportional to the payoff (or utility) of agent i in a generalized first price auction (where each agent pays the amount of his own bid), thus it is an overestimate for the payoff in a GSP auction. Much research has tackled the scenario of a single agent optimizing the bid b_i in an adaptive manner so to maximize revenue, and the game-theoretic properties of such strategies. We simulate both adaptive and non-adaptive bidding strategies. In [17] two basic bidding strategies are described, the first strategy is to bid high enough so to increase the chances of securing a good rank, the second is to lower the bid so to increase the payoff in case the auction is won. For the non-adaptive case we sample the bid value from a power-law distribution that represents a probabilistic mixed strategy that pursue both goals at once in a balanced manner. Thus we choose to generate the differences $v_i - b_i$ according to a power law distribution. For the adaptive case, we have implemented the equilibrium converging strategy described in [9]. The two cases give similar outcomes and we report the non-adaptive results.

4 Experimental Results

The simulations in this section were all run under the GSP mechanism. Moreover, the keyword spreading mechanism is applied after the first 20% of the queries have been processed. Fig. 6a shows the increment in search engine revenue between a basic simulation (in which no agent ever changes keywords) and one where we allow 20% of the agents to apply keyword spreading, using both the Wordnet database and our clustering techniques. The difference levels-off with a gain of 0.5 to 0.8 of a point. Given the total value of the adwords market this difference is very significant in absolute terms.

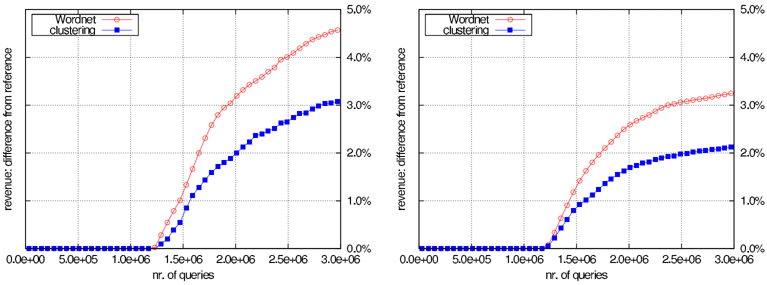
In Fig. 7a we show the revenue increase for the agents that are allowed to change words (20% of total), and in Fig. 7b for those that are not allowed (80% of total). For both groups the revenue increase is positive and levels-off, with the agents in the first group having better performance (in the range 3.0%-4.5%).



(a) Search engine's revenue when 20% of the agents change their keywords with synonyms (Wordnet and clustering data).

(b) Search engine's revenue increase for varying fraction of keyword spreading agents.

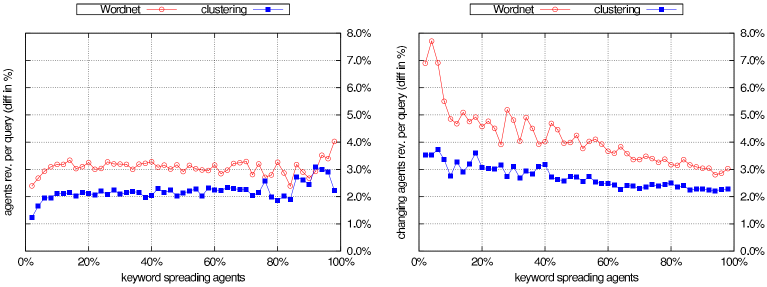
Fig. 6. Revenue increase for the Search Engine



(a) Keyword spreading agents

(b) Non keyword spreading agents

Fig. 7. Revenue increase for agents



(a) Non keyword-spreading agents

(b) Keyword-spreading agents

Fig. 8. Agent's revenue increase for varying fraction of keyword spreading agents

Figures 6b, 8a, 8b show the variation in revenue increase when we vary the fraction of users using keyword spreading from 5% to 95% of the total. While the increased revenue for the search engine and for non-keyword spreading agents is hardly affected, we notice a clear effect of diminishing marginal gain for keyword-spreading agents, with initial gains up to 7% for early adopters, and just 2.5% when the practice is widespread.

We also explore the effect of a class of strategic behavior called *budget depletion strategies*. Here for 20% of the words the top bidder of each such word that does not obtain an advertisement slot will switch to a policy of increasing its bid so to deplete the competition's budget faster, without incurring any additional cost. We explore two cases (a) called "unrealistic" where the strategic agent knows the optimal new bid value, and (b) called "realistic" where the optimal new bid value is sought by small successive increments. SE do not seem to suffer in revenue from this type of strategic agents. In the realistic case, the strategic agents may have to bid above their own valuation, thus risking a negative payoff. In our simulation (data not shown, see [8]) this effect overweights the potential gain from the depletion of competitors' budget.

5 Conclusions

Keyword spreading in sponsored search auction is a technique aiming at extracting more value from the long tail of the distribution of user queries volumes. We performed a simulation with a large number of bidding agents and keywords to expose the possible benefits of this technique for all players involved. We conclude that there are non-negligible economic benefits for the search engine running the auction, and for the bidding agents. There is also a competitive advantage for early adopters. Our simulations are based on publicly available data, on educated guesses as to the shape of some relevant distributions, and on a very large numbers of agents, keywords and queries involved. As a word of caution, we remark that the model for each single agent we employed is rather simple (both adaptive and non-adaptive). Thus, although our results are interesting for search engine companies and bidders, we view them a preliminary investigation. As future research we plan to use more sophisticated user and bidder behavior models (for example by replacing some distributions with "profiles" deducted from real data) so to confirm our findings in a scenario that is more complex in terms of the repertoire of possible individual behaviors. A second future line of research will use more complex market segmentation models and investigate how such models can influence the outcomes of the simulations.

References

1. Abhishek, V., Hosanagar, K.: Keyword Generation for Search Engine Advertising using Semantic Similarity between Terms. In: ICEC 2007 (2007)
2. Abrams, Z.: Revenue maximization when bidders have budgets. In: Proceedings of the ACM-SIAM SODA '06 (2006)
3. Abrams, Z., Mendelevitch, O., Tomlin, J.: Optimal delivery of sponsored search advertisements subject to budget constraints. In: Proc. of the 8th ACM EC, San Diego, California, pp. 272–278 (2007)

4. Acharya, S., Krishnamurthy, P., Deshpande, K., Yan, T.W., Chang, C.C.: A Simulation Framework for Evaluating Designs for Sponsored Search Markets. In: WWW 2007 Sponsored Search Auctions Workshop (2007)
5. Aggarwal, G., Goel, A., Motwani, R.: Truthful auctions for pricing search keywords. In: Proceedings of the 7th ACM EC, pp. 1–7 (2006)
6. Amer-Yahia, S., Lahaie, S., Pennock, D.M.: Towards a generic bidding standard for online advertising. In: Fourth Workshop on Ad Auctions (2008)
7. Auerbach, J., Galenson, J., Sundararajan, M.: An empirical analysis of return on investment maximization in sponsored search auctions. In: Proc. of ADKDD '08, Las Vegas, Nevada, pp. 1–9 (2008)
8. Budinich, M., Codenotti, B., Geraci, F., Pellegrini, M.: On the benefits of keyword spreading in sponsored search auctions: an experimental analysis, Institute for Informatics and Telematics of CNR, TR IIT-10/2009 (2009)
9. Cary, M., Das, A., Edelman, B., Giotis, I., Heimerl, K., Karlin, A.R., Mathieu, C., Schwarz, M.: On Best-Response Bidding in GSP Auctions National Bureau of Economic Research Working Paper No. 13788 (February 2008)
10. Clarke, E.H.: Multipart pricing of public goods. Public Choice 11, 17–33 (1971)
11. Dhillon, I.S., Mallela, S., Kumar, R.: Enhanced word clustering for hierarchical text classification. In: Proc. of 8th ACM SIGKDD, pp. 191–200 (2002)
12. Edelman, B., Ostrovsky, M., Schwarz, M.: Internet advertising and the Generalized Second Price auction: Selling billions of dollars worth of keywords. American Economic Review 97(1), 242–259 (2007)
13. Feldman, J., Muthukrishnan, S., Pal, M., Stein, C.: Budget optimization in search-based advertising auctions. In: Proc. of the 8th ACM EC, San Diego, California, June, pp. 40–49 (2007)
14. Geraci, F., Pellegrini, M., Pisati, P., Sebastiani, F.: A scalable algorithm for high-quality clustering of web snippets. In: Proc. ACM SAC '06, Dijoin, France (2006)
15. Gonen, R., Pavlov, E.: An Adaptive Sponsored Search Mechanism *delta*-Gain Truthful in Valuation, Time, and Budget. In: Deng, X., Graham, F.C. (eds.) WINE 2007. LNCS, vol. 4858, pp. 341–346. Springer, Heidelberg (2007)
16. Gonzalez, T.F.: Clustering to Minimize the Maximum Intercluster Distance. Theor. Comp. Sci. 38(2/3), 293–306 (1985)
17. Ghose, A., Yang, S.: An empirical analysis of sponsored search performance in search engine advertising. In: Proc. of the Intl. Conf. on Web Search and Web Data Mining WSDM '08, New York, pp. 241–250 (2008)
18. Groves, T.: Efficient collective choice when compensation is possible. Review of Economic Studies 46, 227–241 (1979)
19. Kitts, B., LeBlanc, B.J.: A Trading Agent and Simulator for Keyword Auctions. In: Proceedigns of AAMAS 2004, pp. 228–235 (2004)
20. Lahaie, S.: An analysis of alternative slot auction designs for sponsored search. In: Proc. of ACM EC, pp. 218–227 (2006)
21. Lahaie, S., Pennock, D.M., Saberi, A., Vohra, R.V.: Sponsored search auctions. In: Algorithmic Game Theory, ch. 28, pp. 699–716. Cambridge University Press, Cambridge (2007)
22. Li, H., Abe, N.: Word clustering and disambiguation based on co-occurrence data. In: Proc. of the 17th Intl. Conf. on Computational Linguistics, pp. 749–755 (1998)
23. Mehta, A., Saberi, A., Vazirani, U.: Adwords and Generalized Online Matching, J. of ACM 54(5) (2007)
24. Mizuta, H., Steiglitz, K.: Agent-based simulation of dynamic online auctions. In: Proc. of the 32nd Conference on Winter Simulation, Orlando, Florida, San Diego, CA., December 10-13 (2000)

25. Miller, G.A., Fellbaum, C., Tengi, R., Wakefield, P., Poddar, R., Langone, H., Haskell, B.: WordNet: A Lexical Database for English. Princeton University, Princeton (2006)
26. Miller, G.A.: WordNet: An On-line Lexical Database. Int. J. of Lexicography (1990)
27. Richardson, M., Dominowska, E., Ragno, R.: Predicting clicks: estimating the click-through rate for new ads. In: Proc. WWW 2007, pp. 521–530 (2007)
28. Robu, V., Bohte, S., La Poutre, J.A.: The Complex Dynamics of Sponsored Search Markets. In: Cao, L., Gorodetsky, V., Liu, J., Weiss, G., Yu, P.S. (eds.) Agents and Data Mining Interaction. LNCS, vol. 5680, pp. 183–198. Springer, Heidelberg (2009)
29. Rusmevichientong, P., Williamson, D.P.: An adaptive algorithm for selecting profitable keywords for search-based advertising services. In: ACM EC 2006, pp. 260–269 (2006)
30. Varian, H.R.: Position auctions. Int. J. of Industrial Organization 25, 1163–1178 (2007)
31. Vickrey, W.: Counterspeculation, auctions and competitive sealed tenders. J. of Finance 16, 8–37 (1961)

An Optimization Method for Agent's Bidding Strategy in TAC-SCM Game

Xiaoqin Zhang[1], Soheil Sibdari[2], and Saban Singh[1]

[1] Computer and Information Sciences Department
x2zhang@umassd.edu, singhsaban@gmail.com
[2] Decisions and Operations Sciences Department
University of Massachusetts Dartmouth
North Dartmouth, Massachusetts 02747
ssibdari@umassd.edu

Abstract. Intelligent agents have been developed for a number of e-commerce applications including supply chain management. In Trading Agent Competition for Supply Chain Management (TAC SCM), several manufacturer agents compete in a reverse auction in order to sell assembled computers to customers. In this paper, we consider an individual manufacturer agent in TAC SCM and we focus on the sales task. Using a dynamic programming method, the manufacturer agent is able to find an optimized bidding strategy to decide whether to bid for each arriving request for quote (RFQ). The experiment results show that this strategy improves the agent's revenue significantly comparing to several other heuristics in the current practice. This approach can also be applied to similar bidding problems in other e-commerce applications.

Keywords: E-commerce, Supply Chain Management, Bidding Strategy, Trading Agent Competition (TAC).

1 Introduction

Intelligent agent technology has been applied to various stages of Business-to-Customer (B2C) e-commerce process such as product brokering, merchant brokering and negotiation [1]. Meanwhile, agent technology is also very promising in handling a number of complex issues in Business-to-Business (B2B) e-commerce such as supply chain management [2]. The Trading Agent Competition in Supply Chain Management (TAC SCM) is a testbed to stimulate research in this area. The TAC SCM is a simulated computer manufacturing market in which a finite number (currently six) of independent software agents compete in a supply chain game. The agents maximize their profits by competing in a reverse auction to sell computers to a random number of customers. This game is studied over a finite horizon. At the beginning of each period (i.e. day), customers send out requests for quote (RFQs); each RFQ includes the number of requested products and their characteristics, the reserve price per unit, the due date expected, and a penalty the manufacturer needs to pay if the due date is missed. For each RFQ received, the manufacturer agent needs to decide whether to bid on this RFQ and if yes, what the bidding price is. This is referred as the bidding problem. The customer selects

F. Buccafurri and G. Semeraro (Eds.): EC-Web 2010, LNBIP 61, pp. 172–183, 2010.

the best bid to accept. Upon the acceptance of a bid, the manufacturer agent needs to acquire the components from the suppliers. and then conducts a local manufacturing process to assemble products. And, finally, the manufacturer agent receives its payment upon the delivery of the finished products to the customer. Since the manufacturer agent own limited resources, it would be beneficial for the agent to consider future conditions in its current decision. By bidding the right RFQ at the right price, the manufacturer agent can reserve its production capacity for future demand with possibly higher revenue. Current agents in TAC SCM, however, myopically consider the current period and the immediate capacity and decide about the arriving RFQ without considering future demand. Therefore, a practical method that evaluates the agent's policy over the whole sales time horizon can help the agent to maximize its expected profit.

To solve the bidding problem, we first simplify our mathematic model to overcome the complexity that many stochastic components in the problem cause. More specifically, we define a finite set of RFQ types based on all possible combinations of product types, due dates, reserve prices and penalty costs. We then further simplify the bidding problem into a stochastic decision model by estimating the *value* of each RFQ type. This value is estimated using experimental data from running the current TAC SCM game, see Section 5 for the set-up of the game. We use the value as the minimum price that the agent can accept for each product in a given RFQ type. Finally we use stochastic dynamic programming to calculate the optimal bidding strategy that the agent uses in the competition.

In order to provide an evaluation baseline for our model, we also introduce a heuristic using which the agent desires to maximize the revenue only in the current order. Because the agent does not consider the long term impact of its decision on future profit, we call this heuristic *myopic policy*. However, the decision whether to accept or reject a RFQ in a given day impacts the ability of the agent to compete in the market in future. By accepting a RFQ, the agent enters into a contactual agreement and needs to assign resources and facilities to fulfill that order, which reduces its capacity and therefore its future decision of whether to accept or reject a bid.

The literature in TAC SCM is vast and indirectly belongs to a broad stream of supply chain management literature. Most studies focus on the design of the agent. Ketter et al. ([3]) provides a comprehensive survey review of the current literature in TAC SCM agent design. Benisch et. al. (2004) describes the design a specific agent in the TAC SCM game, called Botticelli. This agent competes with other agents to win the customers and to negotiate with suppliers to procure product components. They used stochastic programming approach for the bidding and scheduling problem to determine the optimal solutions. Another optimization technique for the TAC SCM agents is provided by Burke et al. ([4] and [5]). They determine what customer orders to bid on and what prices to bid by combining constraint-based optimization and learning of market conditions. In their model, the agent does this combination to maximize its profit while being restricted by capacity and supply constraints. Kiekkintveld et al. ([6]) address two issues that a manufacturer agent should consider. First, how to deal with the inherent uncertainty in different aspects of the market. And second, how to compete in the market with other agents who play strategically. Greenwald et al. ([7]) presents a bidding strategy for the TAC SCM game using the greedy algorithm. Their marginal bidding provides an incremental solution to incorporate general acceptance conditions such as scheduling and component constraints. However, they have ignored the market competition and focused only on the decision-theoretic optimization problem. For researches directly on TAC SCM see Bell et al. ([8]) and Benisch et al. ([9]).

Our work differs from existing literature because we consider the inter-dependency among time periods. This is the main contribution of this work. We use a mathematical programming that not only considers the expected profit from the immediate bid but also considers the impact of the current decision on the future profit. We specifically categorize the RFQs into different classes (i.e. *RFQ types*) based on their characteristics such as product type, reserve price (maximum bid), due date, etc. We then define the manufacturer agent capacity as the state variable of a discrete event dynamic programming problem with time of the bid as its stage variable. Upon the arrival of a new customer RFQ, the manufacturer agent (here after *agent*) observes the characteristics of the RFQ, reviews its previous contract and decides whether to bid on this RFQ.

In the rest of this paper, we first describe the TAC-SCM game in Section 2, then we present our mathematical model in Section 3. In Section 4 we provide two different proxies in order to compare with our dynamic programming model. Section 5 shows the design of an intelligent agent which uses our mathematical model, and Section 6 provides the details of our experiment and the evaluation results. Section 7 concludes.

2 TAC-SCM Game

The TAC SCM game simulates a real world supply chain scenario. The game is operated over a period of 220 days, each day being simulated as 15 seconds. Six manufacturer agents compete against each other, the one with the most money at the end of the game wins. There are three types of entities in TAC SCM game: customer, manufacturer and supplier. Their interactions are depicted in Figure 1.

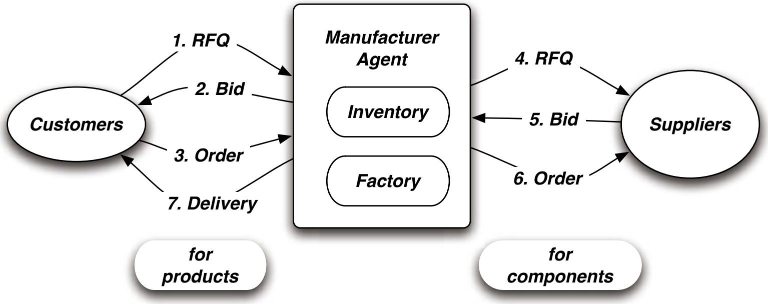


Fig. 1. Manufacturer agent's interactions with customer and suppliers

Customers order PCs from the manufacturers. They send RFQs to all manufacturers, each RFQ consists of the following information.

1. **PC type.** (*productID*) There are 16 types of PCs, which fall into three market ranges, namely low, medium and high range.

2. **Reserve price.** The maximum unit price the customer is willing to pay.
3. **Due date.** The date by then the orders must be shipped to the customer.
4. **Penalty amount.** The amount the manufacturer must pay per day for late delivery.

Manufacturer agents are responsible for producing PCs ordered by customers, it has 2000 production cycles per day. The manufacturer agent receives multiple RFQs from customers every day. For each RFQ, the manufacturer agent decides whether to bid for it. If it decides to bid, it needs to decide the bidding price for this RFQ. The customer selects the best bid and places an order. Upon receiving an order from the customer, the agent then analyzes what types of components and how many of them are needed to fulfill this order. Keeping in mind the components in inventory, it sends RFQs to suppliers for additional components needed. The supplier submits a bid to the manufacturer agent if it is interested in the RFQ for components. The manufacturer agent places an order for the components if it is satisfied with the offer. The manufacturer agent provides its factory a daily production schedule, which specifies the product type (1-16) and the quantity required for production. The completed PC products are shipped to the customer. A warehouse is also provided to the manufacturer agent to store both components and the finished PCs with a daily storage cost charged for each unit. Every 20 days, the manufacturer agent receives a market report with the average price paid for each of the 16 Product types to all the competing manufacturers in the past 20 days. The manufacturer agent can use the information in the report to make better local decisions.

Suppliers are entities that produce the components required to build a PC. There are 8 suppliers in the game. The suppliers are revenue-maximizing agents. They work on make-to-order basis, they do not produce components without an order. Each supplier has a fixed production capacity. When the supplier receives a RFQ from the manufacturer, it checks if it can offer a price less than the reserve price. The supplier can also offer a reduced quantity or it can negotiate on the due-date.

3 Model

Consider an agent who produces a finite number of PC products. To assemble a PC, depending on the product type, the agent needs to spend a specific amount of time that we call *production cycle*. The game is performed during a fixed period of time that we divide it into equal intervals (say T time periods) such that at most one RFQ can be received in each period. We count the time periods in chronological order so that period T represents the last period of the game.

Each RFQ, say RFQ i, consists of *productID*, *quantity*, *reservePrice*, *dueDate*, and *penalty*. The productID determines the configuration of the PCs and determines what components are used in each product (i.e. bill of materials). Currently there are 16 types of productID. In each RFQ, the quantity, q_i, is a discrete variable that is determined uniformly from interval $[q_{min}, q_{max}]$. The customer also specifies a due date dd_i that is a due date of receiving the product. The due date is the current date plus a uniformly chosen order lead time in the interval $[due_{min}, due_{max}]$. The customer also includes their reserve price ρ, which is uniformly chosen in the interval $[\rho_{min}, \rho_{max}]$. Finally, the penalty x is uniformly chosen in interval $[\Psi_{min}, \Psi_{max}]$ and is the cost that the agent is committed to pay to the customer if the product cannot be delivered by the due date [10].

For each RFQ, we define an *offer vector* of $I = \{k, q_i, dd_i, \rho_i, x_i\}$ that specifies the productID, quantity, due dates, reserve price, and penalty cost, respectively.

We now categorize the RFQs by defining *RFQ type* that differentiates between RFQs not only by their PC types but also by their reserve prices, due dates, and penalty costs. For example two RFQs with the same productID, due dates, and reserve price but with different penalty costs are considered two different RFQ types. To define RFQ types we first classify due date, reserve price, and penalty cost as high, medium, and low using appropriate intervals. Therefore there are a total of M RFQ types, which based on our setting $M = 3 \times 3 \times 3 \times 3 = 81$, where the four 3s stand for three types (high, medium, low) of PC type, reservation price, due date, and penalty cost. Now we define P_i as the probability of receiving RFQ type i in each period. At the beginning of each period, upon a RFQ arrival the agent should decide whether to bid and if bid at what price. We use a dynamic programming method to solve this problem in order to maximize its total expected profit from period t until period T.

In order to determine the expected profit of accepting an order based on a RFQ, we need to determine the marginal profit of each RFQ type. We use historical data (i.e. different experiments using the existing game) to estimate the marginal revenue that can be generated from each RFQ. The expected revenue (here after *revenue*) for RFQ $i, i = 1, \cdots, M$ is r_i and we assume that this revenue is given and remains constant over all time periods. Note that r_i consists of all costs to produce the products in RFQ i including material and assembly costs. Using the historical data, other parameters for each product type such as the required capacity and marginal revenue can be estimated. We estimate the number of production cycles that are needed to assemble j units of PCs as specified in RFQ type i by c_{ij} and we define f_i as the marginal revenue from RFQ type i.

Furthermore, we use the historical data to model the impact of competition in the market. As the TAC SCM game is a multi-agent game, we also measure the interaction between different agents using the historical data. We measure the probability of bid acceptance by customers as a function of the bid price made by the agent. In other words, we determine the probability that the offers made by other manufacturer agents are less appealing to the customer. A mathematical model that addresses this competition explicitly that provides the Nash equilibrium is an interesting extension of our model. We define $g_i(x)$ as the probability of accepting a bid of price x by the customer. Determining the appropriate bidding price x is another a problem that is left for future work. Currently, we use a heuristic described in Section 4.2 to determine the bidding price for RFQ type i, hence $g_i(x)$ can be simplified as g_i.

Lets define $J(c,t)$ as the agent's expected profit at beginning of period t when its production capacity is c. The following dynamic program can be used to calculate the optimal decision of whether to bid or to ignore a RFQ of type i.

$$J(c,t) = \sum_{i=1}^{M} \sum_{j=1}^{K} P_i * Q_{ij} * max(J(c,t+1), g_i * (j * f_i + J(c - c_{ij}, t+1)) + (1 - g_i) * J(c, t+1)) \quad (1)$$

with boundary conditions of $J(0,t) = J(c,T) = 0$ for all values of $c \geq 0$ and $0 \leq t \leq T$, and Table 1 explains those parameters.

Table 1. Parameters Used in Equation 1

Name	Meaning
P_i	The probability of receiving a RFQ of type i.
Q_{ij}	The probability of requesting j units of products in a RFQ if type i.
g_i	Probability that a bid for RFQ type i will be accepted.
f_i	Revenue from each unit of product in RFQ type i
c_{ij}	The number of production cycles needed to produce j units of products in RFQ type i.
i	The RFQ type. Currently ranges from 1 to 81.
K	Maximum number of products in an RFQ . Currently ranges from 1 to 20.

Equation 1 is based on the following principle. In order to maximize its expected profit, an agent with c available production cycles at time t should accept a RFQ of type i if the profit it makes from this RFQ plus the maximum profit it can make at time $t+1$ with the rest of production cycles after satisfying the requests in this RFQ is greater than the maximum profit it can make at time $t+1$ with c available production cycles. Otherwise, the agent should not bid for this RFQ. The values for P_i, Q_{ij}, g_i and f_i were gathered from the experiments described in Section 5. For tractability, We assume that the agents does not hold any inventory, or equivalently, we assume that inventory and storage costs are zero.

4 Baseline Agents

To evaluate our approach, we first build a manufacturer agent, named Agent B using some of the best practices found in TAC-SCM literature. We then modify Agent B by using the optimized bidding strategy found with the model presented in Section 3, this modified agent is named Agent A. Since each competition has six manufacturer agents, there are dummy agents provided by the game designer to fill the entries. Here we then describe how the dummy agent and Agent B make decisions in the competition.

4.1 Dummy Agent

A dummy agent bids for a RFQ if both the following criteria are satisfied.

1. The due date for the RFQ is greater than 5 days from the current date, given the component ordering process takes a minimum of 5 days and the dummy agent only orders components from the suppliers after receiving the PC order.
2. $ReservePrice > (BasePrice * 0.9)$.

The bid price is calculated using the following formula:
$BidPrice = (BasePrice * 0.9) +$
$[ReservePrice - (BasePrice * 0.9)] * (1.0 - RandomFactor * PriceDiscountFactor)$
$BasePrice$ is the sum of the nominal price of all components for the PC, which is given in the game specification. $RandomFactor$ is a random value between 0 and 1, and $PriceDiscountFactor$ is 0.3.

The dummy agent sends RFQs for components to suppliers after receiving PC orders. The reserve price is set to 0, which means there is no constraint on the supplier's bid price, and the due date for the supplier RFQ is set to 2 days from the current date. The dummy agent produces to fulfill the orders with the most recent due dates within the limitation of components and production cycles. Within the limitation of available finished PCs, orders with immediate due dates are shipped from the inventory.

4.2 Intelligent Agent B

Agent B uses an inventory-driven strategy to select customer RFQs, it only bids for RFQs according to what is presently available in its inventory. By doing so, the agent avoids paying penalties for overcommitting itself, since the quantity of PCs it can produce is constrained by the availability of components and the factory cycles. This strategy was used by the agent SouthamptonSCM in 2004 [11]. More specifically, the RFQs are considered one at a time. A bid is sent for the RFQ if there are enough factory cycles available to meet the required quantity of products in the RFQ, and there is enough inventory available for production of the products requested in the RFQ.

When the agent decides to bid for a RFQ, the required quantity instead of the actual quantity in the RFQ is deducted from both the inventory and the production capacity, because the bid on this RFQ may not be accepted by the customer.
$RequiredQuantity = ActualQuantity * AcceptanceRate$
$AcceptanceRate = \frac{AverageOrderCount}{AverageOfferCount}$
The average order count and average offer count are taken over the past 5 days.

The bid price for a RFQ is set as:
$BidPrice = max(MinimumPrice, EstimatePrice)$
$MinimumPrice = BasePrice * 1.1$ (to guarantee profit margin of at least 10%)
The $EstimatePrice$ is set differently depending on the number of days (d):

- If $d > 20$, $EstimatePrice = CostFactor * AverageProductPrice$
  $AverageProductPrice$ is taken from the 20-day market report.
- If $d <= 20$, no market report is available yet, then:
  $EstimatePrice = CostFactor * BasePrice$

$CostFactor = LowestProductPrice_{day\#(d-1)}))/BasePrice$

The end game strategy in sales involves the following tactics.

1. The acceptance rate is calculated over of period of 2 days instead of 5 days. Usually in the end of the game, there is a sharp peak in acceptance rate from one day to another owing to the fact that other agents are more reluctant to send bids then.
2. The predicted acceptance rate is increased by 70% from the past 2-day record, due to the same reason described above.
3. The agent provides a discount of 30% on all offers in order to deplete inventory.

In order to maintain high factory utilization (80-100%), a very high inventory level is required. So the inventory threshold for components is fixed as 1200 and that for finished PCs is set as 40. On day 0 (the 1st day), an initial order is placed for all the components. The amount of components ordered is equal to the threshold value 1200.

Additional RFQs are sent to the suppliers to maintain the fixed threshold 1200 throughout the game. The idea of ordering large quantity of components on the first 2 days and then maintaining a threshold is adopted from Agent Mertacor's design [12]. The RFQ for components is created as the follows.

1. The reserve price of component RFQ as:
   $ReservePrice = BasePrice + (0.1 * BasePrice)$.
   If $d > 20$, $BasePrice$ is the average market price from the market report.
   If $d <= 20$, $BasePrice$ is the mean cost of the component according to the agent's record over the past days.
2. The due day is set to 4 days from the current day.

The end game strategy in procurement is applied from after the 200th day. This strategy involves decreasing the threshold for components drastically.

1. Threshold = 400 (Day 201 to Day 205)
2. Threshold = 100 (Day 206 to Day 210)
3. Threshold = 0 (Day 211 to Day 219), the agent stops ordering components.

Agent B uses a greedy approach to handle the production. It caters to orders whose due date is closer. In addition to producing PCs for orders, it also maintains a threshold of finished PCs (40 PCs of each type) if enough components and production cycles are available [13]. Agent B also uses the greedy approach to handle delivery, it ships PCs available in the inventory to meet deadlines even if those PCs were produced for a different order [13].

5 Optimizing the Bidding Strategy - Agent A

The main difference between Agent A and B is that Agent A uses the Expected Profit Matrix (EPM) to decide whether to bid on a customer RFQ. To build the matrix, we first categorize the RFQs in the following way according to the value ranges given in the specification document [10].

- Product type - (Low, Mid, High)
- Due date - (Short [3-6 days], Mid [6-9 days], Long [9-12 days])
- Reserve price - (Low [75 - 90%], Mid [90 - 100%], High [110 - 125%] of base price)
- Penalty - (Low [5-8%], Mid [8-12%], High [12-15%] of reserve price)

To generate EPM - $J_M(c, t)$, the expected profit the manufacturer agent can make at time t with production capacity as c (Equation 1), the following data are gathered from the experiments for each RFQ type i: the probability of getting such RFQ (P_i), the probability of having j units of products in one RFQ (Q_{ij}), the acceptance rate g_i; the revenue from each unit of product in one RFQ (f_i). Among these data, revenue f_i and acceptance rate g_i are highly dependent on what kind of strategies other agents in the competition are using. Since we are planning two types of evaluation tests:

- One Intelligent Agent (Agent A or B) + 5 Dummy agents
- Agent A + Agent B + 4 Dummy agents

we collected data in the following two experiment settings and generated two estimated profit matrices that are used by Agent A in the above two tests, respectively:

- One Agent B + 5 Dummy agents
- 2 Agent B + 4 Dummy agents

10 games were run for each setting and the average values are used to generate the matrix. Dynamic programming method is used to generate a matrix of the size 22000 (c) x 320 (t). The size was chosen for the following reasons:

- The maximum due date for any RFQ cannot be more than 12 days from the date of it being received. So, 11 days was chosen as the planning period. Hence, the total capacity at our disposal was (2000*11).
- According to the specification document [10], the maximum number of RFQs that can be sent out by the customer in one day is 320. So we split one day as 320 time slices and there is no more than one RFQ arriving on within any one time slice.

Agent A is similar to Agent B, except the following modifications. Agent A has a planning period of 11 days, it uses the EPM $J_M(c,t)$ to decide whether to bid for a RFQ or ignore it.

```
For ($t = 1, t <= 320, t++$)
        for the RFQ $R$ of type $i$ received at $t$,
                if bid for $R$, $revenue_A = g_i * (J_M(c - c', t + 1) + Revenue(R))$
                $+(1 - g_i) * J_M(c, t + 1)$, where $c'$ is the capacity needed for $R$
                if ignore $R$, $revenue_B = J_M(c, t + 1)$
                if $revenue_A > revenue_B$, then bid for $R$, otherwise ignore R.
```

Agent A uses a capacity vector to prevent it from overbidding and incurring penalty. The capacity vector is a 11 day window ranging from (current day + 1) to (current day + 11). Initially, the available cycles on each day are 2000. Every day before the RFQ selection process, the capacity required for current orders is deducted. The remaining capacity can then be used for the selection process. To further prevent penalties incurred by overbidding, the acceptance rate for Agent A is inflated by 10% from Agent B's calculation of acceptance rate, and the daily production cycles is also deflated by 10% (100 cycles). The amount of components in the inventory is deflated by 200. For example, if there are 600 components of component A, then bidding is done as if there were only 400 components. The reason for the deduction is that the a production schedule is created after the sales process, so in the sales process a certain portion of the inventory needs to be reserved for the current days production.

6 Experiment Results

In this section, we present the experimental results in two different settings. In this first setting, two types of competitions have been run. One competition is among Agent A and five dummy agents, and another one is among Agent B and five dummy agents. Each type competition is repeated 8 times and the average values and standard deviations are shown in Table 2. The average revenue made by Agent A (16.26M) is almost twice as the average revenue made by Agent B (8.01M). Both agents receive about the

Table 2. Results for the First Setting: one Intelligent agent and 5 dummy agents

	Agent A		Dummy Agents		Agent B		Dummy Agents	
	Average	STDV	Average	STDV	Average	STDV	Average	STDV
Revenue (in Millions)	16.26	3.63	6.78	3.89	8.01	4.13	4.16	1.84
Factory Utilization (%)	95.21	0.78			91.40	0.77		
Total RFQ	40712	3946			40521	3487		
Total Offer	16451	321			14063	300		
Total Order	7538	97			6512	133		
Penalty (in Millions)	0.797	0.356			0.156	0.137		
Penalized RFQ	137	44			36	22		

same number of RFQs from customer, Agent A actually produces 15.7% more orders compared to Agent B. Agent A maintains a slightly higher factory utilization percentage (95.21%) compared to Agent B (91.40%). So we can conclude that most of the extra revenue made by Agent A owes to carefully selecting of RFQs to respond by using the expected profit matrix.

In the second setting, the competition is among two intelligent manufacturer agents (A and B) and four dummy agents. The competition is repeated 10 times and the average values and standard deviations are shown in Table 3. In this setting, Agent A and B compete directly with other 4 dummy agents. The average revenue gained by Agent A (16.66M) is 28% more than Agent B (13.0M). Both agents receives exactly the same set of RFQs from customer, Agent A responded to 8.7% more RFQs and received 12.9% more orders from customer comparing to Agent B. So part of the extra revenue obtained by Agent A should be attributed to the usage of the expected profit matrix. The improvement of agent A's performance over agent B is less significant in this setting compared to the first setting. The reason could be the bigger gap between the environment (2 Agent B + 4 dummy agents) where the data is collected to generate the EPM and the environment (Agent A + Agent B + 4 dummy agents) where the EPM is applied. The more realistic data the EPM is built upon, the better performance can be achieved when using the EPM.

Table 3. Results for the Second Setting: Agent A, B and 4 dummy agents

	Agent A		Agent B		Dummy Agents	
	Average	STDV	Average	STDV	Average	STDV
Revenue (in Millions)	16.66	2.82	13.0	5.2	7.6	1.9
Factory Utilization (%)	91.46	3.54	86.68	5.17		
Total RFQ	39726	4226	39726	4226		
Total Offer	24732	913	22473	1210		
Total Order	7186	315	6364	259		
Penalty (in Millions)	0.662	0.151	0.361	0.301		
Penalized RFQ	131	32	76	59		

7 Conclusions

In this paper, we describe a supply chain scenario in Trading Agent Competition (TAC), an environment that allows the testing of various approaches for several interrelated problems. We focus on the RFQ selection problem for manufacturer agents. We proposed to build an expect profit matrix using dynamic programming method and then use this matrix to make bidding decisions. We implement this approach in an intelligent agent - Agent A, which is otherwise the same as Agent B - an intelligent agent that combines some of the best practices in literature for other problems. The experiment results show that the modification for Agent A improves its performance significantly, hence demonstrate the power of the formal methods on solving supply chain problem. The future work includes extending this model to reason about the best bidding price, and to find Nash equilibriums by explicitly modeling the competition in the market.

References

1. Pivk, A., Gams, M.: Intelligent agents in e-commerce. Electrotechnical Review 67(5), 251–260 (2000)
2. Fox, M.S., Barbuceanu, M., Teigen, R.: Agent-oriented supply-chain management. International Journal of Flexible Manufacturing Systems 12(2-3), 165–188 (2000)
3. Ketter, W., Collins, J., Gini, M.: A survey of agent designs for TAC SCM. In: AAAI-08 Workshop on Trading Agent Design and Analysis, TADA-08 (2008)
4. Burke, D.A., Brown, K.N., Tarim, S.A., Hnich, B.: Learning market prices for a real-time supply chain management trading agent. In: AAMAS 2006 Workshop on Trading Agent Design and Analysis (2006)
5. Burke, D.A., Brown, K.N.: A constraint based agent for TAC SCM. Working paper (2008)
6. Kiekintveld, C., Miller, J., Jordan, P.R., Wellman, M.P.: Controlling a supply chain agent using value-based decomposition. In: The Proceedings of 7th ACM Conference on Electronic Commerce, pp. 208–217 (2006)
7. Greenwald, A., Naroditskiy, V., Odean, T., Ramirez, M., Sodomka, E., Zimmerman, J., Cutler, C.: Marginal Bidding: An Application of the Equimarginal Principle to Bidding in TAC SCM, pp. 217–239. Springer, Heidelberg (2009)
8. Bell, S., Benisch, M., Benthall, M., Greenwald, A., Tschantz, M.C.: Multi-period online optimization in tac-scm: The supplier offer acceptance problem. In: Proc. Workshop on Trading Agent Design and Analysis at the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004), pp. 21–27 (2004)
9. Benisch, M., Greenwald, A., Grypari, I., Lederman, R., Naroditskiy, V., Tschantz, M.: Botticelli: A supply chain management agent. In: Third International Conference on Autonomous Agents and Multiagent Systems, pp. 1174–1181 (2004)
10. Collins, J., Arunachalam, R., Sadeh, N., Eriksson, J., Finne, N., Janson, S.: The supply chain management game for the 2007 trading agent competition. Technical Report CMU-ISRI-07-100, Carnegie Mellon University, Pittsburgh, PA 15213 (2006)

11. He, M., Rogers, A., Luo, X., Jennings, N.R.: Designing a successful trading agent for supply chain management. In: Nakashima, H., Wellman, M.P., Weiss, G., Stone, P. (eds.) AAMAS 2006, pp. 1159–1166. ACM, New York (2006)
12. Kontogounis, I., Chatzidimitriou, K.C., Symeonidis, A.L., Mitkas, P.A.: A robust agent design for dynamic scm environments. In: Antoniou, G., Potamias, G., Spyropoulos, C., Plexousakis, D. (eds.) SETN 2006. LNCS (LNAI), vol. 3955, pp. 127–136. Springer, Heidelberg (2006)
13. Collins, J., Ketter, W., Gini, M., Agovic, A.: Software architecture of the minnetac supply-chain trading agent. Technical Report 07-006, University of Minnesota, Dept. of Computer Science and Engineering, Minneapolis, MN (2007)

Concession Behaviour in Automated Negotiation

Fernando Lopes[1] and Helder Coelho[2]

[1] LNEG – National Research Institute
Estrada do Paço do Lumiar 22, 1649-038 Lisbon, Portugal
fernando.lopes@ineti.pt

[2] University of Lisbon, Department of Computer Science
Bloco C6, Piso 3, Campo Grande, 1749-016 Lisbon, Portugal
hcoelho@di.fc.ul.pt

Abstract. Traditional negotiation, conducted face-to-face and via mail or telephone, is often difficult to manage, prone to misunderstanding, and time consuming. Automated negotiation promises a higher level of process efficiency, and more importantly, a faster emergence and a higher quality of agreements. The potential monetary impact has led to an increasing demand for systems composed of software agents representing individuals or organizations and capable of reaching efficient agreements. At present, work on automated negotiation has generated many useful ideas and concepts leading to important theories and systems. Yet, the design of software agents with negotiation competence largely lacks systematic, traceable, and reproducible approaches, and thus remains more an art than a science. Against this background, this paper presents a model for software agents that handles two-party and multi-issue negotiation. The model incorporates various concession strategies and negotiation tactics. Concession strategies are computationally tractable functions that define the tactics to be used both at the outset and throughout negotiation. Tactics, in turn, are functions that specify the short-term moves to be made at each point of negotiation.

Keywords: Automated negotiation, Negotiation strategies, Negotiation tactics, Multi-agent systems.

1 Introduction

Negotiation is a discussion among conflicting parties with the aim of reaching agreement about a divergence of interests [13]. The list of situations that can be handled by negotiation is endless. Some situations are purely competitive, as when the parties have completely opposed interests. Other situations are purely cooperative, as when the parties have perfectly compatible interests. Most situations are mixed-motive, containing elements of both competitive and cooperative situations – the parties' interests are imperfectly correlated [14]. There are, however, several characteristics common to most negotiation situations, including [6]: (i) two or more parties, (ii) a conflict among the parties, and (iii) an individual preference to search for agreement rather than to appeal to a higher authority, to permanently break off contact, or to fight openly.

F. Buccafurri and G. Semeraro (Eds.): EC-Web 2010, LNBIP 61, pp. 184–194, 2010.

Negotiation may involve two parties (bilateral negotiation) or more than two parties (multilateral negotiation) and one issue (single-issue negotiation) or many issues (multi-issue negotiation). Also, negotiation may proceed through several distinct phases or stages, notably a beginning or initiation phase, a middle or problem-solving phase, and an ending or resolution phase [6]. The initiation phase focuses on preparation and planning for negotiation − it is marked by each party's efforts to emphasize points of difference and to posture for positions. The problem-solving phase seeks a solution for a dispute − it is characterized by extensive interpersonal interaction, strategic maneuvers, and movement toward a mutually acceptable agreement. The resolution phase focuses on details and implementation of a final agreement.

Traditional negotiation, conducted face-to-face and via mail or telephone, is often difficult to manage, prone to misunderstanding, and time consuming [1]. Negotiators are typically satisfied with the final outcome and, in many instances, proudly describe it. However, they frequently view conflict-laden situations with a fundamentally more distrustful, win-lose attitude than is necessary or desirable, and settle for outcomes that are worse for them than other available solutions [19]. They often fail to achieve agreements on the Pareto optimal or efficient frontier (*i.e.*, the locus of achievable joint evaluations from which no joint gains are possible [17]).

Automated negotiation promises a higher level of process efficiency, and most importantly, a faster emergence and a higher quality of agreements. The potential monetary impact has led to an increasing demand for systems composed of software agents representing individuals or organizations and capable of reaching mutually beneficial agreements (e.g., the industrial trend toward agent-based supply chain management). Yet, the design of software agents with negotiation competence largely lacks systematic, traceable, and reproducible approaches, and thus remains more an art than a science. There is much further work to be done, and some current ideas and concepts are likely to be substantially altered as researchers move ahead (but see [10]).

Against this background, this paper presents a model for software agents that handles two-party and multi-issue negotiation. The model incorporates a bilateral negotiation protocol, a set of concession strategies, and a set of negotiation tactics. The protocol formalizes the set of possible tasks that the agents can perform during the course of negotiation. The strategies and tactics formalize the tasks that each agent should perform to negotiate effectively. More specifically, the strategies define the tactics to be used both at the beginning and during the course of negotiation. The tactics formalize the individual moves to be made at each point of the negotiation process.

This paper builds on our previous work in the area of automated negotiation. In particular, it extends the work presented in [7,8,9] by introducing precise definitions for the key components of our model. Also, it formalizes concession strategies as computationally tractable functions that specify the tactics to be used both at the outset and throughout negotiation. Furthermore, at every period, strategies state whether bargaining should continue or terminate.

The remainder of the paper is structured as follows. Section 2 presents a negotiation model for software agents. Section 3 discusses related work and compares the negotiation model with other existing models. Finally, section 4 presents concluding remarks and indicates future avenues of research.

2 A Negotiation Model for Software Agents

Let $\mathcal{A}=\{a_1, a_2\}$ be the set of autonomous agents (negotiating parties). Both the number of agents and their identity are fixed and known to all the participants. The negotiation issues $\{x_1, \ldots, x_n\}$ are quantitative in nature and defined over continuous domains $\{D_1, \ldots, D_n\}$, respectively. For each issue x_k, the range of acceptable values is represented by the interval $D_k=[min_k, max_k]$. The issues are also known to all the participants.

Preparation and planning are often considered the foundations for success in negotiation [6]. Accordingly, effective negotiators often make efforts to perform a number of activities before starting to bargain, including: (i) prioritizing the issues, and (ii) defining realistic, pessimistic, and optimistic targets. Prioritization usually involves two steps, namely deciding which issues are most important and which are least important, and determining whether the issues are connected or separate. Priorities can be set in a number of ways (e.g., to use standard techniques, such as the nominal group technique). For the sake of simplicity, we consider that negotiators set priorities by ranking-order the issues.

Target setting usually involves defining three key points for each issue at stake:

1. *the resistance point or limit* – the point where negotiators decide to stop the negotiation rather than to continue, because any settlement beyond this point is not minimally acceptable;
2. *the target point or level of aspiration* – the point where negotiators realistically expect to achieve a settlement;
3. *the optimistic point or asking price* – the best deal negotiators could possibly hope to assume.

We present below precise definitions for these intuitions.

Definition 1 (Issue, Agenda). *A negotiation issue is a resource to be allocated or a consideration to be resolved in negotiation. The negotiating agenda is the set* $\mathcal{I}=\{x_1, \ldots, x_n\}$ *of issues to be deliberated during negotiation.* ■

Definition 2 (Priority, Weight). *The priority* prt_k^i *of an agent* $a_i \in \mathcal{A}$ *for an issue* $x_k \in \mathcal{I}$ *is a number that represents the importance of* x_k*. The weight* w_k^i *is a number that represents the preference for* x_k*.* ■

Definition 3 (Limit, Target Point, Optimistic Point). *The limit* lim_k^i *of an agent* $a_i \in \mathcal{A}$ *for an issue* $x_k \in \mathcal{I}$ *is the ultimate fallback position for* x_k*, the point beyond which* a_i *is unwilling to concede on* x_k*. The target point* trg_k^i *is the point at which* a_i *is satisfied with the value of* x_k*. The optimistic point* opt_k^i *is the most preferred or ideal value for* x_k*.* ■

2.1 The Negotiation Protocol and Negotiators' Preferences

The negotiation protocol is an alternating offers protocol [11]. Two agents or players bargain over the division of the surplus of $n \geq 2$ distinct issues. The players determine an allocation of the issues by alternately submitting proposals at times in $\mathcal{T} = \{1, 2, \ldots\}$. This means that one proposal is made per time period $t \in \mathcal{T}$, with an agent, say $a_i \in \mathcal{A}$, offering in odd periods $\{1, 3, \ldots\}$, and the other agent $a_j \in \mathcal{A}$ offering in even periods $\{2, 4, \ldots\}$. The agents have the ability to unilaterally opt out of the negotiation when responding to a proposal.

The negotiation process starts with a_i submitting a proposal $p^1_{i \to j}$ to a_j in period $t = 1$. The agent a_j receives $p^1_{i \to j}$ and can either accept the offer (`Yes`), reject it and opt out of the negotiation (`Opt`), or reject it and continue bargaining (`No`). In the first two cases the negotiation ends. Specifically, if $p^1_{i \to j}$ is accepted, negotiation ends successfully and the agreement is implemented. Conversely, if $p^1_{i \to j}$ is rejected and a_j decides to opt out, negotiation terminates with no agreement. In the last case, negotiation proceeds to the next time period $t = 2$, in which a_j makes a counter-proposal $p^2_{j \to i}$. The tasks just described are then repeated. Once an agreement is reached, the agreed-upon allocations of the issues are implemented.

Definition 4 (Proposal). *Let $\mathcal{A}$ be the set of negotiating agents and $\mathcal{I}$ the set of issues at stake in negotiation. Let $\mathcal{T}$ be the set of time periods. A proposal $p^t_{i \to j}$ submitted by an agent $a_i \in \mathcal{A}$ to an agent $a_j \in \mathcal{A}$ in period $t \in \mathcal{T}$ is a vector of issue values:*

$$p^t_{i \to j} = (v_1, \ldots, v_n)$$

where v_k, $k = 1, \ldots, n$, is a value of an issue $x_k \in \mathcal{I}$. ■

Definition 5 (Agreement, Possible Agreements). *An agreement is a proposal accepted by all the negotiating agents in $\mathcal{A}$. The set of possible agreements is:*

$$\mathcal{S} = \{(v_1, \ldots, v_n) \in \mathbb{R}^n : \ v_k \in D_k, \ \textit{for } k = 1, \ldots, n\}$$

where v_k is a value of an issue $x_k \in \mathcal{I}$. ■

Negotiators should express their own preferences to rate and compare incoming offers and counter-offers. The most common way to model the preferences of the negotiating agents is probably to define a utility function over all possible outcomes [4,17]. Let $\mathcal{I} = \{x_1, \ldots, x_n\}$ be the agenda and $\mathcal{D} = \{D_1, \ldots, D_n\}$ the set of issue domains. We consider that each agent $a_i \in \mathcal{A}$ has a continuous utility function: $U_i : \{D_1 \times \ldots \times D_n\} \cup \{\texttt{Opt}, \texttt{Disagreement}\} \to \mathbb{R}$. Accordingly, when the utility for a_i from one outcome is greater than from another outcome, we assume that a_i prefers the first outcome over the second. The outcome `Opt` is interpreted as one of the agents opting out of the negotiation in a given period of time. Perpetual disagreement is denoted by `Disagreement`.

Now, the additive model is probably the most widely used in multi-issue negotiation – the parties assign numerical values to the different levels on each issue and add them to get an entire offer evaluation [17]. This model is simple and intuitive, and therefore well suited to the purposes of this work. We consider that each agent a_i has a scoring or single-issue (marginal) utility function for each issue at stake in negotiation, *i.e.*, a function that gives the score a_i assigns to a value of an issue x_k. For convenience, scores are kept in the interval [0,1]. Additionally, as mentioned above, we consider that a_i has a multi-issue utility function to rate offers.

Definition 6 (Multi-Issue Utility Function). *Let $\mathcal{A}=\{a_1,a_2\}$ be the set of negotiating agents and $\mathcal{I}=\{x_1,\ldots,x_n\}$ the negotiating agenda. The utility function U_i of an agent $a_i\in\mathcal{A}$ to rate offers and counter-offers takes the form:*

$$U_i(x_1,\ldots,x_n) \;=\; \sum_{k=1}^{n} w_k^i \times V_k^i(x_k)$$

where:

(i) *w_k^i is the weight of a_i for an issue $x_k\in\mathcal{I}$;*
(ii) *$V_k^i(x_k)$ is the (marginal) utility function of a_i for x_k.* ∎

Negotiation may end with either agreement or no agreement. Failure to agree can occur in two ways: (i) either party decides to opt out unilaterally, or (ii) the two do not agree to any proposal. The resistance points or limits play a key role in reaching agreement when the parties have the ability to unilaterally opt out of the negotiation – they define the worst agreement for a given party which is still better than opting out. For each agent $a_i\in\mathcal{A}$, we will denote this agreement by $\hat{s}_i\in\mathcal{S}$. Hence, $\hat{s}_i$ will be the least-acceptable agreement for a_i, *i.e.*, the worst (but still acceptable) agreement for a_i. The set of all agreements that are preferred by a_i to opting out will be denoted by S_i.

Definition 7 (Least-acceptable Agreement, Acceptable Agreements). *The least-acceptable agreement for an agent $a_i\in\mathcal{A}$ is defined as: $\hat{s}_i=(lim_1^i,\ldots,lim_n^i)$, where lim_k^i, $k=1,\ldots,n$, is the limit of a_i for an issue $x_k\in\mathcal{I}$. The set of acceptable agreements for a_i is:*

$$S_i = \{s\colon\; s\in\mathcal{S},\; U_i(s)\;\geq\; U_i(\hat{s}_i)\}$$

where $U_i(\hat{s}_i)$ is the utility of $\hat{s}_i$ for a_i. ∎

Perpetual disagreement is the least-preferred or worst outcome, *i.e.*, disagreement is even worse than opting out. Thus, the agents prefer any agreement in any given time period over the continuation of the negotiation process indefinitely. Formally, and more precisely, we state the following:

(1) (Acceptable agreements versus opting out). For every agent $a_i\in\mathcal{A}$ and acceptable agreement $s\in S_i$, $U_i(s)\geq U_i(\texttt{Opt})$.
(2) (Opting out versus Disagreement). For every agent $a_i\in\mathcal{A}$, $U_i(\texttt{Opt}) > U_i(\texttt{Disagreement})$.

2.2 Concession Strategies

The opening offer and the initial concessions are two central elements of negotiation [19]. When negotiation begins, the parties are faced with a fundamental question – should the opening offer be exaggerated, more toward the optimistic point, or modest, somewhat closer to the limit? The main advantages of an exaggerated initial offer are [12]: (i) negotiators can concede further and hence elicit more counterconcessions from their opponent, and (ii) negotiators' later demands are likely to look generous. However, an exaggerated opening offer frequently communicates an attitude of toughness that may be harmful to long-term relationships. Also, it may be seen as too high by the other party and therefore summarily rejected. By contrast, an opening offer seen as reasonable or modest by the other party could perhaps have been higher, either to leave more room for movement or to achieve a higher settlement.

After the first round of offers, other fundamental question is, what concessions are to be made? Negotiators can choose to make none, holding firm and insisting on their original positions. By taking a firm position, negotiators attempt to capture most of the initial bargaining or settlement range (defined by the opening offers of both parties). However, there is the very real possibility that firmness will be reciprocated – one or both parties may become intransigent and withdraw completely. Negotiators can also choose to make some concessions, being flexible and changing their original positions. Flexibility often keeps negotiation going – the more flexible one party seems to be, the more the other party will believe that a settlement is possible. Hence, if concessions are to be made, another fundamental question is, how large should they be?

Concession strategies are computationally tractable functions that model significant opening positions and typical patterns of concessions. They specify the tactics to be used at the outset of negotiation (to prepare the initial offers). Also, at each step of negotiation, they specify the tactics to be used in preparing counter-offers. Furthermore, concession strategies state whether bargaining should continue or terminate. The words "computationally tractable functions" presume that agents are able to compute concession strategies in a reasonable amount of time. A formal definition of a generic strategy follows.

Definition 8 (Concession Strategy). *Let $\mathcal{A}$ be the set of negotiating agents, $\mathcal{I}$ the negotiating agenda, $\mathcal{T}$ the set of time periods, and $\mathcal{S}$ the set of possible agreements. Let $a_i \in \mathcal{A}$ be the first agent to submit a proposal and T_i his set of tactics. A concession strategy $C_i : T_i \times T_i \times \mathcal{T} \rightarrow \mathcal{S} \cup \{\texttt{Yes}, \texttt{No}, \texttt{Opt}\}$ for a_i is a function with the following general form:*

$$C_i = \begin{cases} \textit{apply } O_i(x_k) \textit{ and offer } p^1_{i\rightarrow j}\,, & \textit{if } a_i\textit{'s turn and } t=1 \\ \textit{reject } p^{t-1}_{j\rightarrow i} \textit{ and quit}, & \textit{if } a_j\textit{'s turn and } U_i(p^{t-1}_{j\rightarrow i}) < U_i(\hat{s}_i) \\ \textit{apply } Y_i(x_k, f^i_k) \textit{ and prepare } p^t_{i\rightarrow j} & \textit{if } a_j\textit{'s turn and } U_i(p^{t-1}_{j\rightarrow i}) \geq U_i(\hat{s}_i) \\ \textit{if } U^*_i \geq 0 \textit{ accept } p^{t-1}_{j\rightarrow i} \textit{ else reject}, & \\ \textit{offer compromise } p^t_{i\rightarrow j}\,, & \textit{if } a_i\textit{'s turn and } t>1 \end{cases}$$

where:

(i) $p^1_{i\to j}$ *is the opening offer of* a_i, $p^{t-1}_{j\to i}$ *is the offer of* a_j *for period* $t-1$ *of negotiation, and* $p^t_{i\to j}$ *is the offer of* a_i *for the next period* t *of negotiation;*
(ii) *for each issue* $x_k \in \mathcal{I}$, $O_i(x_k)$ *is an opening negotiation tactic,* $Y_i(x_k, f^i_k)$ *is a concession tactic, and* $f^i_k \in [0,1]$ *is a real number that defines the magnitude of a concession on* x_k*, referred to as the concession factor of* a_i *for* x_k (*see subsection* 2.3*, below*)*;*
(iii) $U_i(\hat{s}_i)$ *is the utility of the least-acceptable agreement for* a_i*;*
(iv) $U^*_i = U_i(p^{t-1}_{j\to i}) - U_i(p^t_{i\to j})$ ■

Two explanatory and cautionary notes are in order here. First, notation is being abused somewhat, by using C_i rather than $C_i(O_i(x_k), Y_i(x_k, f^i_k), t)$. The abuse helps improve readability, however, and meaning will always be clear from context. Second, tactics are functions of a single issue rather than a vector of issues. This permits great flexibility, since it allows agents to model a wide range of concession behaviors (e.g., large initial demands and slow concession making).

Interestingly, bargainers sometimes have different strengths of preference for the issues at stake – they place greater emphasis on some key issues and make significant efforts to resolve them favourably. Hence, they often yield on less important or low-priority issues, in the hope that their opponent will make compensating concessions [17,19]. A generic low-priority concession making strategy for a_i takes the form (again, the definition slightly abuses notation):

$$C^i_{LP} = \begin{cases} \text{apply } O_i \text{ and offer } p^1_{i\to j}\,, & \text{if } a_i\text{'s turn and } t=1 \\ \text{reject } p^{t-1}_{j\to i} \text{ and quit}\,, & \text{if } a_j\text{'s turn and } U_i(p^{t-1}_{j\to i}) < U_i(\hat{s}_i) \\ \text{apply } Y_i \text{ to } \widehat{I}_i \text{ and prepare } p^t_{i\to j} & \text{if } a_j\text{'s turn and } U_i(p^{t-1}_{j\to i}) \geq U_i(\hat{s}_i) \\ \text{if } U^*_i \geq 0 \text{ accept } p^{t-1}_{j\to i} \text{ else reject}\,, & \\ \text{offer compromise } p^t_{i\to j}\,, & \text{if } a_i\text{'s turn and } t>1 \end{cases}$$

where $\widehat{I}_i \subset \mathcal{I}$ is the set of issues that are of lower priority to a_i. The definition of a specific strategy involves basically the specification of a particular opening negotiation tactic (e.g., the tactic "`starting optimistic`") and a key concession tactic to apply to the issues of low priority (e.g., the tactic "`moderate`"). These and other negotiation tactics are defined in the next subsection.

2.3 Negotiation Tactics

Negotiation tactics are functions that model the short-term moves designed to enact high-level strategies. The following two groups of tactics will receive the preponderance of our attention in this paper:

1. *opening negotiation tactics* – functions that specify the demands to be made at the outset of negotiation;
2. *concession tactics* – functions that model the concessions to be made throughout negotiation.

As mentioned above, tactics are functions of a single issue.

Opening Negotiation Tactics. Skilled negotiators often start with high demands to leave room for later movement and hence elicit counterconcessions from their opponent [12]. High initial demands also protect limits from detection and underestimation (this is a concern about image loss). If limits are detected by the opponent, he may become unwilling to accept a better offer than the least-acceptable one, dooming all higher aspirations. If limits are underestimated, the opponent may become committed to unacceptable demands, fostering breakdown of negotiation. Thus, to avoid these dual dangers, bargainers typically place their demands well above their limits as a sort of smoke screen. Furthermore, high initial demands are also partly designed to protect target points (this is a concern about position loss). Clearly, bargainers often need to move in concert with their opponent toward mutually acceptable agreements. This means starting higher than targets and only moving down to them in coordination with the opponent [14].

Noticeably, starting high frequently communicates an attitude of toughness that can be reciprocated by the opponent, thus making negotiation "difficult to resolve" [6]. Hence, should bargainers start with a firm, determined stance, or adopt a position of moderateness and understanding? It follows that bargainers often decide how much to demand on the basis of the concessions they expect from their opponent − the farther the opponent is expected to concede, the more will be demanded (this phenomenon is referred to as tracking).

In general, three levels of initial demand are commonly discussed in the negotiation literature [6,15]: extreme or high, reasonable or moderate, and modest or low. They have motivated the definition of the following opening negotiation tactics:

1. *starting optimistic* − specifies a value for an issue close to the optimistic point;
2. *starting realistic* − specifies a value for an issue in the range defined by the target and the optimistic points;
3. *starting pessimistic* − specifies a value for an issue in the range defined by the target and the resistance points.

A formal definition of the tactic "`starting optimistic`" follows.

Definition 9 (Starting Optimistic). *Let $\mathcal{A}=\{a_1,a_2\}$ be the set of negotiating agents and $\mathcal{I}=\{x_1,\ldots,x_n\}$ the negotiating agenda. Let $\mathcal{D}=\{D_1,\ldots,D_n\}$ be the set of issue domains. The tactic starting optimistic of an agent $a_i \in \mathcal{A}$ for an issue $x_k \in \mathcal{I}$ takes the form:*

$$O_i(x_k) = opt_k^i + \epsilon$$

where:

(i) $\epsilon > 0$ *is small;*
(ii) opt_k^i *is the optimistic point of a_i for x_k.* ■

The definition of the other two tactics is essentially identical to that of "`starting optimistic`", and is therefore omitted.

Concession Tactics. Concessions are a powerful aspect of negotiation – without them, in fact, some researchers consider that negotiation would not exist [19]. A concession is usually defined as a change of offer in the supposed direction of the other party's interests that reduces the level of benefit sought. Concession rate is the speed at which demand level declines over time. A bargainer's demand level can be thought of as the level of benefit to the self associated with the current demand or offer [12,15].

Practically speaking, bargainers often enter negotiation expecting concessions. Their opening position may be good for both sides and might have been the final settlement if the parties started negotiation from different points. Even so, bargainers generally resent a take-it-or-leave-it approach – an offer that may have been accepted had it emerged as a result of concession making may be rejected when it is thrown on the table and presented as a *fait accomply* [6]. Ample research evidence indicates that the parties feel better about a settlement when negotiation has involved a progression of concessions [14].

A formal definition of a generic concession tactic follows (in the interests of readability, and without loss of generality, we consider that $a_i \in \mathcal{A}$ wants to maximize $x_k \in \mathcal{I}$).

Definition 10 (Concession Tactic). *Let $\mathcal{A}=\{a_1, a_2\}$ be the set of negotiating agents, $\mathcal{I}=\{x_1, \ldots, x_n\}$ the negotiating agenda, and $\mathcal{D}=\{D_1, \ldots, D_n\}$ the set of issue domains. A concession tactic $Y_i \colon D_k \times [0,1] \rightarrow D_k$ of an agent $a_i \in \mathcal{A}$ for an issue $x_k \in \mathcal{I}$ is a function with the following general form:*

$$Y_i(x_k, f_k^i) = x_k - f_k^i(x_k - lim_k^i)$$

where:

(i) *f_k^i is the concession factor of a_i for x_k;*
(ii) *lim_k^i is the limit of a_i for x_k.* ■

Negotiators may consider strikingly different patterns of concessions as negotiation unfolds. However, the following three levels of concession magnitude are commonly discussed in the negotiation literature [6,14]: large, substantial, and small. To this we would add two other levels: null and complete. Accordingly, we consider the following five concession tactics:

1. *stalemate* – models a null concession on an issue x_k at stake;
2. *tough* – models a small concession on x_k;
3. *moderate* – models a substantial concession on x_k;
4. *soft* – models a large concession on x_k;
5. *accommodate* – models a complete concession on x_k.

These and other similar tactics are defined by considering specific values for the concession factor f_k^i. In particular, the "`stalemate`" tactic is defined by $f_k^i = 0$ and the "`accommodate`" tactic by $f_k^i = 1$. The other three tactics are defined by considering values for f_k^i in different ranges (e.g., the "`tough`" tactic by $f_k^i \in]0.00, 0.05]$, the "`moderate`" tactic by $f_k^i \in]0.05, 0.15]$, and the "`soft`" tactic by $f_k^i \in]0.15, 0.20]$).

3 Related Work

Artificial intelligence (AI) researchers have investigated the design of agents with negotiation competence from two main perspectives: a theoretical or formal mathematical perspective and a practical or system-building perspective. Researchers following the theoretical perspective have attempted mainly to develop formal models of negotiation, *i.e.*, models for describing, specifying, and reasoning about the key features of negotiating agents. To this end, they have drawn heavily on game-theoretic and economic methods (see, e.g., [2,5,18]). On the other hand, researchers following the practical perspective have attempted mainly to develop computational models of negotiation, *i.e.*, models for specifying the key data structures of negotiating agents and the processes operating on these structures. They have drawn heavily on social sciences techniques for understanding interaction and negotiation (see, e.g., [3,10,16]).

Overall, various researchers have developed models that incorporate specific protocols (notably, the alternating offers protocol) and libraries of negotiation strategies (notably, concession strategies). However, the authors are aware of no similar efforts to define strategies as functions that specify the tactics to be used both at the outset and throughout negotiation. Tactics, in turn, are defined as functions that specify the short-term moves to be made at each point of negotiation. Our interest lies mainly in formalizing important strategies and tactics motivated by rules-of-thumb distilled from good behavioral practice in real-life negotiations.

4 Conclusion

This paper has presented a model for software agents that handles two-party and multi-issue negotiation. The model incorporates a bilateral negotiation protocol, a set of concession strategies, and a set of negotiation tactics. The protocol is an alternating offers protocol. The strategies are computationally tractable functions that define the tactics to be used both at the beginning and during the course of negotiation. The words "computationally tractable functions" presume that agents are able to compute the strategies in a reasonable amount of time. Furthermore, at every period of negotiation, the strategies state whether bargaining should continue or terminate. The tactics are functions that specify the individual moves to be made at each point of the negotiation process.

Autonomous agents equipped with the negotiation model are currently being developed. Our aim for the future is to perform a number of experiments to empirically evaluate the key components of the agents. Also, notice that the task of designing and implementing agents with negotiation competence involves the consideration of insights from multiple relevant research areas. Accordingly, we also intend to develop an interdisciplinary framework for automated negotiation – game-theoretic (strategic) and behavioural negotiation theories should mutually reinforce each other and contribute to richer negotiators.

References

1. Bichler, M., Kersten, G., Strecker, S.: Towards a Structured Design of Electronic Negotiations. Group Decision and Negotiation 12, 311–335 (2003)
2. Fatima, S., Wooldridge, M., Jennings, N.: A Comparative Study of Game Theoretic and Evolutionary Models of Bargaining for Software Agents. Artificial Intelligence Review 23, 185–203 (2005)
3. Jennings, N., Faratin, P., Lomuscio, A., Parsons, S., Wooldridge, M., Sierra, C.: Automated Negotiation: Prospects, Methods and Challenges. Group Decision and Negotiation 10, 199–215 (2001)
4. Keeney, R., Raiffa, H.: Decisions with Multiple Objectives: Preferences and Value Tradeoffs. John Wiley & Sons, Chichester (1976)
5. Kraus, S.: Strategic Negotiation in Multi-Agent Environments. MIT Press, Cambridge (2001)
6. Lewicki, R., Barry, B., Saunders, D., Minton, J.: Negotiation. McGraw Hill, New York (2003)
7. Lopes, F., Mamede, N., Novais, A.Q., Coelho, H.: A Negotiation Model for Autonomous Computational Agents: Formal Description and Empirical Evaluation. Journal of Intelligent & Fuzzy Systems 12, 195–212 (2002)
8. Lopes, F., Mamede, N., Novais, A.Q., Coelho, H.: Negotiation Strategies for Autonomous Computational Agents. In: 16th European Conference on Artificial Intelligence (ECAI-04), pp. 38–42. IOS Press, Amsterdam (2004)
9. Lopes, F., Mamede, N., Novais, A.Q., Coelho, H.: Negotiation Among Autonomous Agents: Experimental Evaluation of Integrative Strategies. In: 12th Portuguese Conference on Artificial Intelligence, pp. 280–288. IEEE Computer Society Press, Los Alamitos (2005)
10. Lopes, F., Wooldridge, M., Novais, A.Q.: Negotiation Among Autonomous Computational Agents: Principles, Analysis and Challenges. Artificial Intelligence Review 29, 1–44 (2008)
11. Osborne, M., Rubinstein, A.: Bargaining and Markets. Academic Press, San Diego (1990)
12. Pruitt, D.: Negotiation Behavior. Academic Press, New York (1981)
13. Pruitt, D.: Social Conflict. In: Gilbert, D., Fiske, S., Lindzei, G. (eds.) The Handbook of Social Psychology, vol. 2, pp. 470–503. McGraw-Hill, New York (1998)
14. Pruitt, D., Carnevale, P.: Negotiation in Social Conflict. Open University Press, Philadelphia (1993)
15. Pruitt, D., Kim, S.: Social Conflict: Escalation, Stalemate, and Settlement. McGraw Hill, New York (2004)
16. Rahwan, I., Ramchurn, S., Jennings, N., McBurney, P., Parsons, S., Sonenberg, L.: Argumentation-based Negotiation. The Knowledge Engineering Review 18, 343–375 (2004)
17. Raiffa, H.: The Art and Science of Negotiation. Harvard University Press, Cambridge (1982)
18. Sandholm, T.: Distributed Rational Decision Making. In: Weiss, G. (ed.) Multi-Agent Systems – A Modern Approach to Distributed Artificial Intelligence, pp. 201–259. MIT Press, Cambridge (1999)
19. Thompson, L.: The Mind and Heart of the Negotiator. Prentice-Hall, Englewood Cliffs (2005)

Bilateral Negotiation in a Multi-agent Supply Chain System

Fernando Lopes[1] and Helder Coelho[2]

[1] LNEG – National Research Institute
Estrada do Paço do Lumiar 22, 1649-038 Lisbon, Portugal
fernando.lopes@ineti.pt

[2] University of Lisbon, Department of Computer Science
Bloco C6, Piso 3, Campo Grande, 1749-016 Lisbon, Portugal
hcoelho@di.fc.ul.pt

Abstract. A supply chain is a set of organizations directly linked by flows of services from suppliers to customers. Supply chain activities range from the ordering and receipt of raw materials to the production and distribution of finished goods. Supply chain management is the integration of key activities across a supply chain for the purposes of building competitive infrastructures, synchronizing supply with demand, and leveraging worldwide logistics. This paper addresses the challenges created by supply chain management towards improving long-term performance of companies. It presents a multi-agent supply chain system composed of multiple software agents, each responsible for one or more supply chain activities, and each interacting with other agents in the execution of their responsibilities. Additionally, this paper presents the key features of a negotiation model for software agents. The model handles bilateral multi-issue negotiation and incorporates an alternating offers protocol, a set of logrolling strategies, and a set of negotiation tactics.

Keywords: Autonomous agents, Multi-agent supply chain system, Automated negotiation, Bargaining.

1 Introduction

Multi-agent systems (MAS) are ideally suited to represent problems that have multiple problem solving entities and multiple problem solving methods [3]. The major motivations for the increasing interest in MAS research include the ability to solve problems in which data, expertise, or control is distributed, the ability to allow inter-operation of existing legacy systems, and the ability to enhance performance along the dimensions of computational efficiency, reliability, and robustness. Agent technology has been used to solve real-world problems in a range of industrial and commercial applications, including manufacturing, process control, telecommunications, air traffic control, information management, electronic commerce, and business process management (see, e.g., [11]).

F. Buccafurri and G. Semeraro (Eds.): EC-Web 2010, LNBIP 61, pp. 195–206, 2010.

A supply chain is a set of organizations directly linked by flows of services from suppliers to customers. Supply chain activities range from the ordering and receipt of raw materials to the production and distribution of finished goods. Supply chain management (SCM) is the integration of key activities across a supply chain for the purpose of improving long-term performance. SCM encompasses the planning and management of all activities involved in sourcing, procurement, conversion, and logistics. It also includes the crucial components of collaboration and coordination with channel partners (e.g., suppliers, intermediaries, and customers). In essence, SCM integrates supply and demand management within and across companies. The main objectives include building competitive infrastructures, leveraging worldwide logistics, synchronizing supply with demand, and measuring performance globally.

Supply chain management in general and multi-agent supply chain systems in particular have received some attention lately (see, e.g., [1,2]). However, despite the prominent models proposed in the literature, most challenges created by SCM are still waiting to be addressed more thoroughly. At present, there is a need to develop computational tools to help manage the complexity of SCM. Against this background, the purpose of this paper is twofold:

1. *to present a multi-agent supply chain system* – the system is composed of a collection of software agents, each responsible for one or more supply chain activities, and each interacting with other agents in the execution of their responsibilities;
2. *to present the key features of a negotiation model for software agents* – the model handles bilateral multi-issue negotiation and incorporates an alternating offers protocol, a set of logrolling strategies, and a set of negotiation tactics.

Logrolling strategies are computationally tractable functions that define the tactics to be used both at the beginning and during the course of negotiation. The words "computationally tractable functions" presume that agents are able to compute the strategies in a reasonable amount of time. Furthermore, at every period of negotiation, the strategies state whether bargaining should continue or terminate. Negotiation tactics are functions that specify the individual moves to be made at each point of the negotiation process.

This paper builds on our previous work in the area of automated negotiation [6,7,8]. In particular, it introduces precise definitions for logrolling strategies. It also lays the foundation for performing an experiment to investigate the performance of agents operating in a supply chain system and equipped with our negotiation model.

The remainder of the paper is structured as follows. Section 2 is devoted to negotiation between software agents. Section 3 describes a multi-agents supply chain system and illustrates how software agents equipped with our model operate in a negotiation setting. Finally, related work and concluding remarks are presented in sections 4 and 5 respectively.

2 Multi-agent Negotiation

Negotiation is a discussion among conflicting parties with the aim of reaching agreement about a divergence of interests. Negotiation may involve two parties (bilateral negotiation) or more than two parties (multilateral negotiation) and one issue (single-issue negotiation) or many issues (multi-issue negotiation). This section presents the key features of a model for software agents that handles two-party and multi-issue negotiation.

2.1 Pre-negotiation

Pre-negotiation is the process of preparing and planning for negotiation and involves mainly the creation of a well-laid plan specifying the activities that negotiators should attend to before actually starting to negotiate [5]. Let $\mathcal{A}=\{a_1, a_2\}$ be the set of autonomous agents (negotiating parties). Both the number of agents and their identity are fixed and known to all the participants. Let $\mathcal{I}=\{x_1, \ldots, x_n\}$ be the negotiating agenda – the set of issues to be deliberated during negotiation. The issues are quantitative in nature and defined over continuous domains. Let $\mathcal{D}=\{D_1, \ldots, D_n\}$ be the set of issue domains. For each issue x_k, the range of acceptable values is represented by the interval $D_k=[min_k, max_k]$. The issues are also known to all the participants.

Effective pre-negotiation requires that negotiators prioritize the issues and define the targets. Priorities are set by ranking-order the issues, *i.e.*, by defining the most important, the second most important, and so on. The priority prt_k^i of an agent $a_i \in \mathcal{A}$ for an issue $x_k \in \mathcal{I}$ is a number that represents the importance of x_k. The weight w_k^i is a number that represents the preference for x_k. The resistance point or limit lim_k^i is the ultimate fallback position for x_k, the point beyond which a_i is unwilling to concede on x_k. The level of aspiration or target point trg_k^i is the point at which a_i is satisfied with the value of x_k. The asking price or optimistic point opt_k^i is the most preferred or ideal value for x_k.

Additionally, effective pre-negotiation requires that negotiators agree on an appropriate protocol that defines the rules governing the interaction. The negotiation literature describes several protocols that vary significantly depending on the type and amount of information exchanged between agents (see, e.g., [9,15]). Simple protocols allow agents to exchange only proposals, *i.e.*, solutions to the problem they face. Richer protocols allow agents to provide feedback on the proposals they receive. This feedback often takes the form of critiques, *i.e.*, comments on which parts of proposals are acceptable or unacceptable. Sophisticated protocols allow agents to provide arguments to support their negotiation stance.

Most complex protocols make, however, considerable demands on any implementation, mainly because they appeal to very rich representations of the agents and their environments. Therefore, we consider a simple alternating offers protocol [10]. Two agents or players bargain over the division of the surplus of $n \geq 2$ distinct issues. The agents determine an allocation of the issues by alternately submitting proposals at times in $\mathcal{T} = \{1, 2, \ldots\}$.

The negotiation process starts with an agent, say $a_i \in \mathcal{A}$, submitting a proposal $p^1_{i\to j}$ to the other agent $a_j \in \mathcal{A}$ in period $t=1$. The agent a_j receives $p^1_{i\to j}$ and can either accept the offer (`Yes`), reject it and opt out of the negotiation (`Opt`), or reject it and continue bargaining (`No`). In the first two cases the negotiation ends. Specifically, if $p^1_{i\to j}$ is accepted, negotiation ends successfully and the agreement is implemented. Conversely, if $p^1_{i\to j}$ is rejected and a_j decides to opt out, negotiation terminates with no agreement. In the last case, negotiation proceeds to the next time period $t=2$, in which a_j makes a counter-proposal $p^2_{j\to i}$. The tasks just described are then repeated. Once an agreement is reached, the agreed-upon allocations of the issues are implemented.

The negotiation procedure, labelled the "joint-offer procedure", involves bargaining over the allocation of the entire endowment stream at once. A proposal $p^t_{i\to j}$ submitted by an agent $a_i \in \mathcal{A}$ to an agent $a_j \in \mathcal{A}$ in period $t \in \mathcal{T}$ is a vector $(v_1, \ldots, v_n)$ of issue values. An agreement is a proposal accepted by all the agents in $\mathcal{A}$. The set of possible agreements is $\mathcal{S} = \{(v_1, \ldots, v_n) \in \mathbb{R}^n\}$, where $v_k \in D_k$, for $k = 1, \ldots, n$, is a value of an issue $x_k \in \mathcal{I}$.

The players' preferences are modelled by defining a utility function over all possible outcomes. More specifically, we consider that each agent $a_i \in \mathcal{A}$ has a continuous utility function: $U_i : \{D_1 \times \ldots \times D_n\} \cup \{\texttt{Opt}, \texttt{Disagreement}\} \to \mathbb{R}$. Accordingly, when the utility for a_i from one outcome is greater than from another outcome, we assume that a_i prefers the first outcome over the second. The outcome `Opt` is interpreted as one of the agents opting out of the negotiation in a given period of time. Perpetual disagreement is denoted by `Disagreement`.

Now, the additive model is probably the most widely used in multi-issue negotiation − the parties assign numerical values to the different levels on each issue and add them to get an entire offer evaluation [16]. This model is simple and intuitive, and therefore well suited to the purposes of this work. The utility function U_i of a_i to rate offers and counter-offers takes the form:

$$U_i(x_1, \ldots, x_n) \;=\; \sum_{k=1}^{n} w^i_k \times V^i_k(x_k)$$

where w^i_k is the weight of a_i for an issue $x_k \in \mathcal{I}$ and $V^i_k(x_k)$ is the scoring (or marginal) utility function of a_i for x_k, *i.e.*, the function that gives the score a_i assigns to a value of an issue x_k.

Negotiation may end with either agreement or no agreement. Failure to agree can occur in two ways: (i) either party decides to opt out unilaterally, or (ii) the two do not agree to any proposal. The resistance points or limits play a key role in reaching agreement when the parties have the ability to unilaterally opt out of the negotiation − they define the worst agreement for a given party which is still better than opting out. For each agent $a_i \in \mathcal{A}$, we will denote this agreement by $\hat{s}_i \in \mathcal{S}$. Hence, $\hat{s}_i$ will be the least-acceptable agreement for a_i, *i.e.*, the worst (but still acceptable) agreement for a_i. The set of all agreements that are preferred by a_i to opting out will be denoted by S_i. Perpetual disagreement is the least-preferred or worst outcome, *i.e.*, disagreement is even worse than opting out.

2.2 Actual Negotiation

Actual negotiation is the process of moving toward agreement (usually by an iterative exchange of offers and counter-offers). The negotiation protocol defines the states (e.g., accepting a proposal), the valid actions of the agents in particular states (e.g., which messages can be sent by whom, to whom, at what stage), and the events that cause states to change (e.g., proposal accepted). It marks branching points at which agents have to make decisions according to their strategies. Thus, at each step of negotiation, agents often need to follow their strategies to choose among different possible actions to execute.

Negotiation strategies are often implemented through a variety of tactics [13,14]. The line between strategies and tactics often seems indistinct, but one major difference is that of scope. Tactics are short-term moves designed to enact broad (or high-level) strategies − they are structured, directed, and driven by strategic considerations [5]. Accordingly, in this work strategies are computationally tractable functions that define the tactics to be used both at the beginning and during the course of negotiation. The words "computationally tractable functions" presume that agents are able to compute the strategies in a reasonable amount of time. Also, at every period of negotiation, strategies state whether bargaining should continue or terminate. Tactics, in turn, are functions that specify the short-term moves to be made at each point of negotiation.

Negotiation strategies can reflect a variety of behaviours and lead to strikingly different outcomes. However, logrolling is commonly discussed in the behavioral negotiation literature − two parties agree to exchange concessions on different issues, with each party yielding on issues that are of low priority to himself and high priority to the other party [17]. Accordingly, logrolling will receive the preponderance of our attention in this paper.

Logrolling Strategies. Most well-intended negotiators tend to believe that, above all, success depends on the creativity to devise agreements that yield considerable gain to both negotiating parties. They see the essence of negotiation as expanding the "pie" of available resources, as pursuing joint gains. They are essentially value creators − they attempt to probe below the surface of the other party's true needs to locate mutually superior solutions [12].

Logrolling is possible only when several issues are under consideration and the parties have different priorities among these issues. The parties then agree to exchange concessions on (part or all) of the issues, each party winning on the issues he places greater emphasis. In this way, each party gets the fraction of his demands that he deems most important. Clearly, a theory of logrolling in complex agendas is of particular importance to both human and automated negotiation. However, there are important questions still waiting to be addressed more thoroughly. We highlight the following: which issues will be grouped for the exchange of concessions? Relevant efforts to answer this questions include the theory of appropriate exchange and the principle of equivalence [14]. But it is clear that much more research still needs to be performed. In this work, we consider the following three subsets of the agenda for each agent $a_i \in \mathcal{A}$:

- a subset I_i^+, containing the issues of higher priority to a_i (and are also believed to be of lower priority to his opponent a_j);
- a subset I_i^-, containing the issues of lower priority to a_i (and are also believed to be of higher priority to a_j);
- a subset $I_i^{\pm}$, containing the remaining issues of the agenda ($\mathcal{I} = I_i^+ \cup I_i^{\pm} \cup I_i^-$).

Negotiators have frequently something to offer that is relatively less valuable to them than to their opponent, and thus, the subsets I_i^+ and I_i^- are typically non-empty. These two subsets contain the logrolling issues, *i.e.*, the issues that can be logrolled to make profitable trade-offs. By contrast, the subset $I_i^{\pm}$ contains both the distributive issues (the parties' interests are directly opposed) and the compatible issues (the parties have coordinated interests). A formal definition of a generic logrolling strategy follows.

Definition 1 (Logrolling Strategy). *Let $\mathcal{A}$ be the set of negotiating agents, $\mathcal{I}$ the negotiating agenda, $\mathcal{T}$ the set of time periods, and $\mathcal{S}$ the set of possible agreements. Let $a_i \in \mathcal{A}$ be the first agent to submit a proposal, T_i his set of tactics, and $a_j \in \mathcal{A}$ his opponent. Let I_i^+ be the set of issues that are of higher priority to a_i (and are believed to be of lower priority to a_j), I_i^- the set of issues that are of lower priority to a_i (and are believed to be of higher priority to a_j), and $I_i^{\pm}$ the remaining issues of the agenda. A logrolling strategy $L_i \colon T_i \times T_i \times T_i \times T_i \times \mathcal{T} \rightarrow \mathcal{S} \cup \{\texttt{Yes}, \texttt{No}, \texttt{Opt}\}$ for a_i is a function with the following general form:*

$$L_i = \begin{cases} \textit{apply } O_i(x_k) \textit{ and offer } p^1_{i\rightarrow j}\,, & \textit{if } a_i\textit{'s turn and } t=1 \\ \textit{reject } p^{t-1}_{j\rightarrow i} \textit{ and quit}, & \textit{if } a_j\textit{'s turn and } U_i(p^{t-1}_{j\rightarrow i}) < U_i(\hat{s}_i) \\ \textit{apply } Y_i^+(x_k, f_k^i) \textit{ to } I_i^+ & \textit{if } a_j\textit{'s turn and } U_i(p^{t-1}_{j\rightarrow i}) \geq U_i(\hat{s}_i) \\ \textit{apply } Y_i^{\pm}(x_k, f_k^i) \textit{ to } I_i^{\pm} & \\ \textit{apply } Y_i^-(x_k, f_k^i) \textit{ to } I_i^- & \\ \textit{prepare } p^t_{i\rightarrow j} & \\ \textit{if } U_i^* \geq 0 \textit{ accept } p^{t-1}_{j\rightarrow i} \textit{ else reject}, & \\ \textit{offer logrolling solution } p^t_{i\rightarrow j}\,, & \textit{if } a_i\textit{'s turn and } t>1 \end{cases}$$

where:

(i) *$p^1_{i\rightarrow j}$ is the opening offer of a_i, $p^{t-1}_{j\rightarrow i}$ is the offer of a_j for time period $t-1$ of negotiation, and $p^t_{i\rightarrow j}$ is the offer of a_i for the next time period t of negotiation;*

(ii) *for each issue $x_k \in \mathcal{I}$, $O_i(x_k)$ is an opening negotiation tactic;*

(iii) *$Y_i^+(x_k, f_k^i)$, $Y_i^{\pm}(x_k, f_k^i)$ and $Y_i^-(x_k, f_k^i)$ are concession tactics, and f_k^i is the concession factor of a_i for x_k (see below);*

(iv) *$U_i(\hat{s}_i)$ is the utility of the least-acceptable agreement for a_i;*

(v) *$U_i^* = U_i(p^{t-1}_{j\rightarrow i}) - U_i(p^t_{i\rightarrow j})$* ∎

Two explanatory and cautionary notes are in order here. First, notation is being abused somewhat, by using L_i rather than $L_i(O_i(x_k), Y_i^+(x_k, f_k^i), Y_i^\pm(x_k, f_k^i), Y_i^-(x_k, f_k^i), t)$. The abuse helps improve readability, however, and meaning will always be clear from context. Second, tactics are functions of a single issue rather than a vector of issues. This permits great flexibility, since it allows agents to model a wide range of negotiation behaviors.

Logrolling can be insightful or simply emerge from concession making. Typical strategies that lead to logrolling solutions include:

1. *starting high and conceding strategically* – negotiators adopt an optimistic opening position, slightly reduce their low-priority demands (and they believe are of high priority to their opponent), and hold firm on their high-priority demands (and they believe are of low priority to their opponent);
2. *starting high and negotiating creatively* – negotiators adopt an optimistic opening position, substantially reduce their low-priority demands (and they believe are of high priority to their opponent), and hold firm on their high-priority demands (and they believe are of low priority to their opponent).

The definition of these and other relevant strategies involves basically the specification of particular tactics. For instance, the strategy "`starting high and negotiating creatively`" is defined by considering the opening negotiation tactic "`starting optimistic`" and the concessions tactics "`moderate`" and "`stalemate`" (but see below).

Opening Negotiation Tactics. Opening negotiation tactics are functions that specify the demands to be made at the outset of negotiation. The following three tactics are commonly discussed in the behavioral negotiation literature [5,14]:

1. *starting optimistic* – specifies a value for an issue close to the optimistic point;
2. *starting realistic* – specifies a value for an issue in the range defined by the target and the optimistic points;
3. *starting pessimistic* – specifies a value for an issue in the range defined by the target and the resistance points.

A formal definition of the tactic "`starting optimistic`" follows (the definition of the other two tactics is essentially identical, and is therefore omitted).

Definition 2 (Starting Optimistic). *Let $\mathcal{A} = \{a_1, a_2\}$ be the set of negotiating agents and $\mathcal{I} = \{x_1, \dots, x_n\}$ the negotiating agenda. Let $\mathcal{D} = \{D_1, \dots, D_n\}$ be the set of issue domains. The tactic starting optimistic of an agent $a_i \in \mathcal{A}$ for an issue $x_k \in \mathcal{I}$ takes the form:*

$$O_i(x_k) = opt_k^i + \epsilon$$

where:

(i) $\epsilon > 0$ *is small;*
(ii) opt_k^i *is the optimistic point of a_i for x_k.* ∎

Concession Tactics. Concession tactics are functions that model the concessions to be made throughout negotiation. Practically speaking, negotiators may consider strikingly different patterns of concessions as negotiation unfolds. However, the following three levels of concession magnitude are commonly discussed in the negotiation literature [13,17]: large, substantial, and small. To this we would add two other levels: null and complete. Accordingly, we consider the following five concession tactics:

1. *stalemate* – models a null concession on an issue x_k at stake;
2. *tough* – models a small concession on x_k;
3. *moderate* – models a substantial concession on x_k;
4. *soft* – models a large concession on x_k;
5. *accommodate* – models a complete concession on x_k.

A formal definition of a generic concession tactic follows (without loss of generality, we consider that $a_i \in \mathcal{A}$ wants to maximize $x_k \in \mathcal{I}$).

Definition 3 (Concession Tactic). *Let $\mathcal{A}=\{a_1, a_2\}$ be the set of negotiating agents, $\mathcal{I}=\{x_1, \ldots, x_n\}$ the negotiating agenda, and $\mathcal{D}=\{D_1, \ldots, D_n\}$ the set of issue domains. A concession tactic $Y_i\colon D_k \times [0,1] \rightarrow D_k$ of an agent $a_i \in \mathcal{A}$ for an issue $x_k \in \mathcal{I}$ is a function with the following general form:*

$$Y_i(x_k, f_k^i) = x_k - f_k^i(x_k - lim_k^i)$$

where:

(i) f_k^i *is the concession factor of* a_i *for* x_k*;*
(ii) lim_k^i *is the limit of* a_i *for* x_k*.* ■

The five tactics are defined by considering specific values for the concession factor f_k^i. In particular, the "`stalemate`" tactic is defined by $f_k^i = 0$ and the "`accommodate`" tactic by $f_k^i = 1$. The other three tactics are defined by considering values for f_k^i in different ranges (e.g., the "`tough`" tactic by $f_k^i \in]0.00, 0.05]$, the "`moderate`" tactic by $f_k^i \in]0.05, 0.15]$, and the "`soft`" tactic by $f_k^i \in]0.15, 0.20]$).

3 Agents for Supply Chain Management

Multi-agent systems have generated lots of excitement in recent years because of their promise as a new paradigm for conceptualizing and implementing complex software systems. Central to the design and effective operation of a multi-agent system are a core set of problems and research questions, notably [3]:

1. *the design problem* – how to formulate, describe, decompose, and allocate different problems and synthesize results among a group of intelligent agents?
2. *the coordination problem* – how to ensure that agents act coherently in making decisions or taking action, accommodating the non-local effects of local decisions and avoiding harmful interactions?

The design problem is focused on the domain the system is intended to solve in a distributed manner. This problem consists mainly in distributing different supply chain activities across a number of agents. A typical distribution involves at least the following agents:

1. *sales agent* – responsible for acquiring orders from customers, negotiating with customers, and handling customer requests for modifying or canceling orders;
2. *logistics agent* – responsible for coordinating the plants and distribution centers of a manufacturing enterprise: it manages the movement of materials and products across the supply chain, from the suppliers of raw materials to the customers of finished goods;
3. *scheduling agent* – responsible for scheduling and rescheduling the activities of a manufacturing enterprise;
4. *resource management agent* – responsible for dynamically managing the availability of resources in order to execute the scheduled activities;
5. *supplier agents* and *customer agents* – the suppliers sell raw materials and the customers buy finished goods.

The agents are essentially computer systems capable of flexible autonomous action in order to meet their design objectives.

The coordination problem is focussed on ensuring that agents act in a tightly coordinated manner in order to effectively achieve their objectives. This problem is addressed, at least in part, by designing agents that are able to coordinate their activities through negotiation. Specifically, for the case of a supply chain system, the agents are charged with executing actions towards the achievement of their private goals and, thus, conflicts inevitably occur among them. Negotiation is the predominant process for resolving conflicts.

Let us introduce a specific scenario involving interaction between the sales agent and the logistics agent:

> David, the director of Sales, has lined up two new orders for a total of 15000 men's suits: one for 10000 and the other for 5000 men's suits. Martin, the director of Logistics, has already stated that it will take four months to make the suits. Together, they will gross over a million Euros, with a fine profit for the company. The problem is that Martin insists that the job will take four months and David's customer wants a two-month turnaround. Also, David claims that he can't afford to lose the customer. David and Martin are discussing and, so far, have accomplished little more than making each other angry. However, they can resolve their differences by negotiating a mutually beneficial agreement.

The remainder of this section illustrates how software agents equipped with the proposed model operate in the Sales-Logistics scenario. In particular, we demonstrate both how negotiation evolves and how software agents use different logrolling strategies (and their associated opening negotiation and concession tactics).

Table 1. Major issues, preferences, limits and targets (Sales agent)

Negotiation Issue	Weight	Limit	Target Point
quantity_1	0.350	9500	10000
date_1	0.300	1.25 (5 weeks)	1.00 (4 weeks)
quantity_2	0.175	4000	5000
date_2	0.175	1.50 (6 weeks)	1.00 (4 weeks)

For illustrative purposes, we consider the negotiation process from the viewpoint of David. There are four major issues of concern: `quantity_1`, `date_1`, `quantity_2` and `date_2`. The first two issues are the most important to David due to the inherent customer demands − he wants fast action on the 10000 suit order. Also, after a period of consultation with the customer, David concludes that he is overly firm about the 10000 suit order (and is willing to wait up to five weeks for 10000 suits, or ultimately 9500 suits), but he is moderately firm about the 5000 suit order (and is willing to wait up to six weeks for only 4000 suits). Table 1 shows the four issues, the (normalized) weights, the limits, and the target points of the Sales agent.

Figure 1 shows the joint utility space for David and Martin. The abscissa represents the utility to David, and the ordinate represents the utility to Martin. The solid line OCO' represents the Pareto optimal frontier (*i.e.*, the locus of achievable joint evaluations from which no joint gains are possible [16]). The small squares depict a few options for settling the issues at stake.

Now, we take up a few logrolling strategies, one at a time, and examine their nature and their impact on the negotiation outcome. As noted, it is of higher priority for Sales to get fast action on the 10000 suit order than the 5000 suit order. Suppose now that it is of higher priority for Logistics to handle the 5000 suit order (and to avoid the 10000 suit order). These two departments have the makings of a logrolling deal − each party can yield on issues that are of low priority to himself and high priority to the other party. Accordingly, David and Martin can reach the following solution: a 4-week schedule for 9750 suits and a 6-week schedule for 4500 suits. This agreement is represented by point A in Figure 1 and provides a (normalized) benefit of 0.562 to each agent.

Noticeably, logrolling strategies can permit negotiators to fully exploit the differences in the valuation of the issues to capitalize on optimal agreements. In this way, David and Martin can pursue specific logrolling strategies and agree on a four-week schedule for 10000 suits and a six-week schedule for 4000 suits. This agreement lies along the efficient frontier and is represented by point B in Figure 1 − it provides a (normalized) benefit of 0.65 to each party.

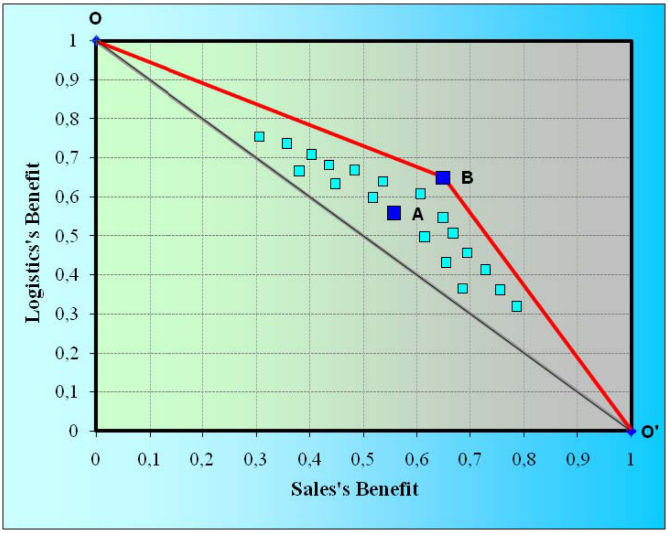


Fig. 1. Joint utility space for the Sales-Logistics negotiation situation

4 Related Work

Artificial intelligence researchers have paid a great deal of attention to automated negotiation over the past decade and a number of prominent models have been proposed in the literature (see, e.g., [4,9,15]. The majority of models primarily use either game-theoretic techniques or methods from the social sciences as a basis to develop autonomous negotiating agents. Furthermore, most models incorporate specific protocols (notably, the alternating offers protocol) and libraries of negotiation strategies (notably, concession and logrolling strategies). However, despite the power and elegance of existing models, we are aware of no similar efforts to define logrolling strategies as functions that specify the tactics to be used both at the outset and throughout negotiation. Tactics, in turn, are defined as functions that specify the short-term moves to be made at each point of negotiation.

5 Conclusion

This article has presented a simplified multi-agent supply chain system composed of a collection of software agents, each responsible for one or more supply chain activities, and each interacting with other agents in the execution of their responsibilities. Additionally, the article has presented the key features of a model for software agents that handles two-party and multi-issue negotiation. The model incorporates an alternating offers protocol, a set of logrolling strategies, and a set of negotiation tactics.

Autonomous agents equipped with the negotiation model are currently being developed. Our aim for the future is to perform a set of inter-related experiments to empirically evaluate the key components of the agents operating in the supply chain system. Each experiment will lay the foundation for subsequent experimental work.

References

1. Burgess, K., Singh, P., Koroglu, R.: Supply Chain Management: Structured Literature Review and Implications for Future Research. International Journal of Operations & Production Management 26, 703–729 (2006)
2. Fox, M., Barbuceanu, M., Teigen, R.: Agent-Oriented Supply Chain Management. International Journal Flexible Manufacturing Systems 12, 165–188 (2000)
3. Jennings, N., Sycara, K., Wooldridge, M.: A Roadmap of Agent Research and Development. Autonomous Agents and Multi-Agent Systems 1, 7–38 (1998)
4. Jennings, N., Faratin, P., Lomuscio, A., Parsons, S., Wooldridge, M., Sierra, C.: Automated Negotiation: Prospects, Methods and Challenges. Group Decision and Negotiation 10, 199–215 (2001)
5. Lewicki, R., Barry, B., Saunders, D., Minton, J.: Negotiation. McGraw Hill, New York (2003)
6. Lopes, F., Mamede, N., Novais, A.Q., Coelho, H.: A Negotiation Model for Autonomous Computational Agents: Formal Description and Empirical Evaluation. Journal of Intelligent & Fuzzy Systems 12, 195–212 (2002)
7. Lopes, F., Mamede, N., Novais, A.Q., Coelho, H.: Negotiation Strategies for Autonomous Computational Agents. In: 16th European Conference on Artificial Intelligence (ECAI-04), pp. 38–42. IOS Press, Amsterdam (2004)
8. Lopes, F., Mamede, N., Novais, A.Q., Coelho, H.: Negotiation Among Autonomous Agents: Experimental Evaluation of Integrative Strategies. In: 12th Portuguese Conference on Artificial Intelligence, pp. 280–288. IEEE Computer Society Press, Los Alamitos (2005)
9. Lopes, F., Wooldridge, M., Novais, A.Q.: Negotiation Among Autonomous Computational Agents: Principles, Analysis and Challenges. Artificial Intelligence Review 29, 1–44 (2008)
10. Osborne, M., Rubinstein, A.: Bargaining and Markets. Academic Press, San Diego (1990)
11. Pechoucek, M., Marik, V.: Industrial Deployment of Multi-agent Technologies: Review and Selected Case Studies. Autonomous Agents and Multi-Agent Systems 17, 397–431 (2008)
12. Pruitt, D.: Negotiation Behavior. Academic Press, New York (1981)
13. Pruitt, D., Carnevale, P.: Negotiation in Social Conflict. Open University Press, Philadelphia (1993)
14. Pruitt, D., Kim, S.: Social Conflict: Escalation, Stalemate, and Settlement. McGraw Hill, New York (2004)
15. Rahwan, I., Ramchurn, S., Jennings, N., McBurney, P., Parsons, S., Sonenberg, L.: Argumentation-based Negotiation. Knowledge Eng. Review 18, 343–375 (2004)
16. Raiffa, H.: The Art and Science of Negotiation. Harvard University Press, Cambridge (1982)
17. Thompson, L.: The Mind and Heart of the Negotiator. Prentice-Hall, Englewood Cliffs (2005)

Receiving Recommendations and Providing Feedback: The User-Experience of a Recommender System

Bart P. Knijnenburg[1], Martijn C. Willemsen[1], and Stefan Hirtbach[2]

[1] Eindhoven University of Technology, Human Technology Interaction group, P.O. Box 513,
5600 MB Eindhoven, The Netherlands
{B.P.Knijnenburg,M.C.Willemsen}@tue.nl
[2] European Microsoft Innovation Center GmbH, Ritterstraße 23, 52072 Aachen
stefah@microsoft.com

Abstract. This paper systematically evaluates the user experience of a recommender system. Using both behavioral data and subjective measures of user experience, we demonstrate that choice satisfaction and system effectiveness increase when a system provides personalized recommendations (compared to the same system that provides random recommendations). We furthermore demonstrate that despite privacy issues, this higher choice satisfaction and system effectiveness increases users' intention to provide feedback about their preference. Due to an intention-behavior gap, this may however not necessarily influence the users' actual feedback behavior.

Keywords: recommender systems, user study, evaluation, user experience, choice satisfaction, system effectiveness, privacy.

1 Introduction

Recommender systems recommend items (e.g. books, movies, laptops) to users based on their stated preferences. In many of these systems users indicate their preference by rating presented items (e.g. from one to five stars). These systems predict the users' rating value of new items based on their rating history, and then present items with the highest predicted rating as recommendations. Although the research field of recommender systems has mainly focused on increasing the predictive accuracy of prediction algorithms [1], several researchers contend that a good recommender system needs more than high prediction accuracy [2], [3]. Researchers have therefore proposed to focus more on the user experience of recommender systems by means of a user-centric development [4] and evaluation [5] process.

The current paper takes up on this user-centric approach to recommender system evaluation. The focus on the user experience allows us to address two important issues concerning the interaction between users and recommender systems. The first issue concerns an unproven premise of most recommender system research, namely that users want to receive personalized recommendations. By comparing a system that provides personal recommendations to the same system without such personalization, we can determine whether personalized recommendations have any effect on the user

F. Buccafurri and G. Semeraro (Eds.): EC-Web 2010, LNBIP 61, pp. 207–216, 2010.

experience at all. The second issue concerns a necessary and often implied condition for recommender systems to work: They can only give good recommendations if users provide feedback by rating items. Users may however not always do this. By analyzing users' feedback behavior and intentions, we can determine what drives and inhibits their propensity to provide feedback.

2 Evaluating the User Experience

In this section, we use related work and the evaluation framework defined in [5] to construct a path model (Fig 1) that can be used to systematically analyze the effect of personalized recommendations on the users' perceptions, evaluations and intentions when using the system.

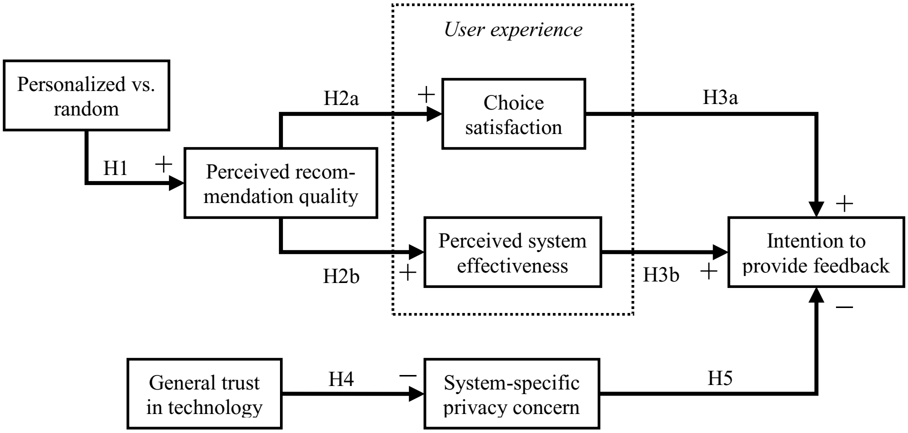

User experience
Personalized vs. random
H1 +
Perceived recom-mendation quality
H2a +
Choice satisfaction
H2b +
Perceived system effectiveness
H3a +
H3b +
Intention to provide feedback
General trust in technology
H4 −
System-specific privacy concern
H5 −

Fig. 1. The path model describing the hypothesized mechanisms underlying the user experience of recommender systems

We test the path model in an experiment with the Microsoft ClipClub system (Fig 2), a Silverlight application with video clips covering lifestyle and entertainment topics. In order to control the causal relations in the model, we manipulate the recommendation quality by randomly assigning our participants to a version of ClipClub that provides personalized recommendations (the **personalized condition**), or to a version that provides random clips as 'recommendations' (the **random condition**). We hypothesize that the higher recommendation quality in the personalized condition leads to a better user experience and a higher intention to provide feedback to the system. All path model constructs are measured using post-experimental questionnaires.

2.1 The Effect of Personalized Recommendations

Several researchers have indicated the limitations of algorithmic performance as an evaluation metric for recommender systems. Although experiments show that users

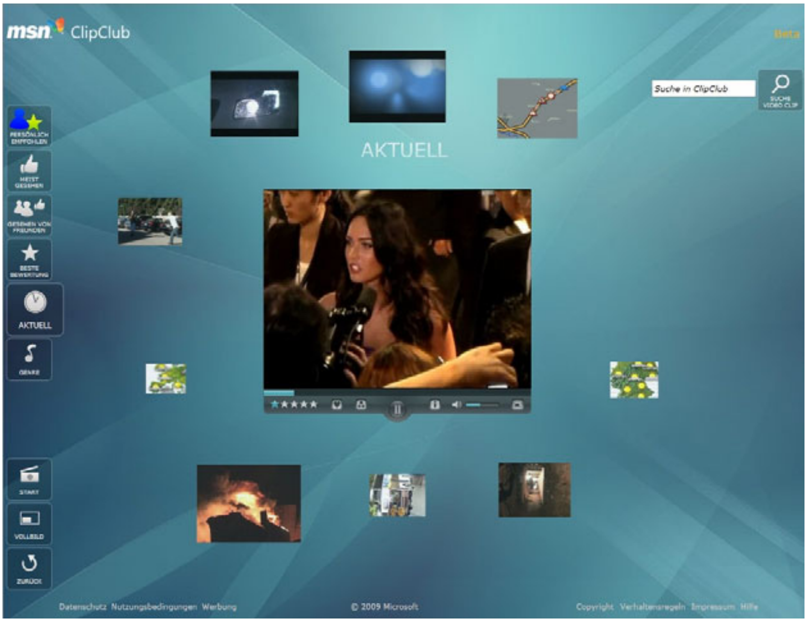


Fig. 2. The ClipClub prototype with our modifications: social network features are disabled and the link to the section with recommendations is made more salient

are able to notice differences in prediction accuracy [6], [3], McNee et al. reason that even observably higher predictive accuracy can sometimes lead to *lower* usefulness of recommendations [2], [4]. Attempts have been made to develop evaluation metrics that more closely match the user experience (e.g. the multi-corridor evaluation metric [7], [8]), but few have used a direct measure of user experience (i.e. asking the users about their experience).

The need for user-centric evaluation metrics is evident in the few studies that actually do present user-centric evaluations. In a comparison of six recommender algorithms, McNee et al. [9] found that the algorithm that provided the best predictions was rated least helpful by the users. Torres et al. [10] found that the algorithm with the lowest predictive accuracy among five resulted in the highest user satisfaction. Finally, Ziegler et al. [3] found that diversifying the recommendation set resulted in lower (real and observed) accuracy, but, up to a certain level, a more positive subjective evaluation.

Based on these findings, it seems to be important to make a distinction between the *perception* and the *evaluation* of recommendation quality. Concerning perception, our path model (Fig 1) includes a measure of perceived recommendation quality, which can determine whether users notice differences in algorithmic performance. Concerning evaluation, the model includes two evaluative measures that determine how the recommendations influence the user experience: choice satisfaction and perceived system effectiveness. The choice satisfaction measure allows us to test whether providing personalized recommendations increases the satisfaction of users with the items they choose. Evidence for this hypothesis is mixed: An extensive review of the field by Xiao and Benbasat [1] finds seven papers reporting increased decision quality for personalized recommendations, two that are inconclusive, and one reporting a decrease in decision quality. The perceived system effectiveness measure allows us to test whether personalized recommendations increase how effective users believe the system to be.

In terms of our path model (Fig 1), we hypothesize that personalized recommendations (compared to random) increase the perceived quality of the presented recommendations (H1), and that the increased quality in turn increases the users' choice satisfaction (H2a) and perceived system effectiveness (H2b). Furthermore, we hypothesize that the effect of personalized recommendations is also observable in the users' behavior in terms of item search (less browsing and shorter overall viewing time) and viewing behavior (more clips watched from beginning to end). If users indeed show reduced browsing and increased consumption, we take this as an objective indicator of higher system effectiveness. Similarly, Häubl et al. report that users examine fewer options when they receive personalized recommendations [11]. The effects of personalized recommendations on the decision time are however ambiguous [1].

2.2 Drivers and Inhibitors of User Feedback

Since feedback aides the personalization process, we hypothesize that the amount of feedback provided to the system is influenced by the effect of personalized recommendations on choice satisfaction and system effectiveness: The more beneficial it seems to be, the more feedback users provide. There is however a tradeoff between user experience-related benefits and the users' privacy concerns [12], [13].

Especially in an online context users are concerned about providing personal information and occasionally refuse to give personal information to certain websites [14]. Moreover, several researchers have found that general and system-specific privacy concerns (i.e. the concern for a specific system's respect for the user's privacy) influence users' willingness to disclose personal information [15], [16], [17], [18]. Nonetheless, a more detailed survey found that 80% of the respondents were usually or always comfortable disclosing personal taste preferences [19]. In terms of our path model (Fig 1), we predict that users' privacy concerns are higher when users have less trust in technology in general (H4) and that higher privacy concerns decrease users' intention to provide feedback (H5).

One could argue that users would be more willing to provide feedback when this improves their experience. Spiekermann et al. [20] suggest that the usefulness of the feedback can overrule initial privacy concerns. However, an overview of privacy surveys estimates the proportion of users willing to give personal information in return for a personalized experience only between 40 and 50% [18]. Explicitly indicating the benefit of providing feedback seems to increase the amount of feedback given by the users [21], [22]. Considering the fact that users can observe differences in prediction accuracy [6], [3] one could argue that for a recommender system the benefit of providing feedback should be evident even without an explicit indication. However, to date no study has considered the effects of personalized recommendations on the amount of input provided to the system [1].

Based on these findings, we expect that personalized recommendations increase the intention to provide feedback, and that this effect is mediated by the better user experience in this condition. Specifically, we hypothesize that users with a higher choice satisfaction and perceived system effectiveness have a higher intention to provide feedback (in Fig 1, H3a and H3b respectively).

In light of privacy and user feedback it is important to note the apparent discrepancy between users' stated privacy-defending strategies and their actual behavior [18]: Users may say that they intend to restrict their feedback due to privacy concerns without actually showing this behavior in practice. Spiekermann et al. [20] show that users provided the same amount of information to the system despite differences in stated privacy concerns. The current study will measure both users' intentions to provide feedback as well as their actual feedback behavior. We expect at most a weak correlation between the two.

3 Method

From September 10 to October 29, 2009 the European Microsoft Innoviation Center (EMIC) carried out an experiment using two slightly modified versions of the MSN ClipClub system (Fig. 2). The goal of the experiment was to test the user experience as described in the path model of Fig 1, as well as possible correlations between these subjective measures and observable user behavior.

3.1 Participants

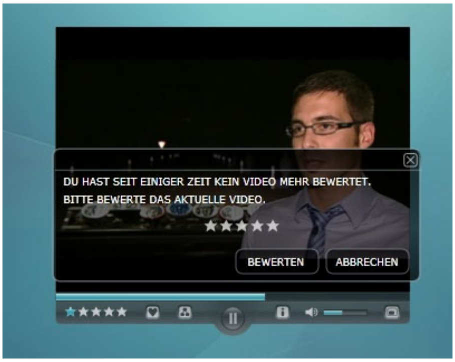
43 participants[1] completed the experiment, 25 in the random and 18 in the recommender condition. The main incentive for participation was a raffle of one 100 Euro and ten 20 Euro electronic gift certificates. Participants, 65% male, were all German and had an average age of 31 (SD = 9.45). Gender and age did not correlate significantly with any of our outcome variables, with the notable exception that females had an overall higher choice satisfaction than males (p < .05). This did not influence any of our main results.

3.2 System

The Microsoft ClipClub system features redacted video clips covering lifestyle and entertainment topics. The content as well as the system itself is in German. A pre-experimental instruction explained the rating feature and its effect on recommendations. The recommendations section was highlighted (blue/yellow icon), and opening this section before rating any clips explained that recommendations would only show up after rating clips. If a participant would not rate any items for five minutes, a ”rating-probe” would pop up asking the user to rate the current clip (Fig 3). Participants were allowed to close this pop-up without rating. After rating the clip, participants were transported to the recommendations. Note that even though all participants were told that rating clips would change the recommendations this action had no beneficial effect on the recommendations in the random condition (i.e. the recommendations changed randomly).

[1] In our experiment all path model constructs are causally linked to the two conditions. 43 data points are therefore expected to contain enough explanatory power to test our model.

The algorithm employed in the personalized condition is a Vector Space Model Engine, an algorithm that uses the tags associated to a clip to create a vector of each clip in the space built by all available tags (the length of the vector depends on the impact the tags have). The system also creates a tag vector for the subset of clips rated by the user, and recommends clips with a tag vector similar to the created tag vector (in terms of cosine similarity). Older ratings are logarithmically discounted, as are older items.



DU HAST SEIT EINIGER ZEIT KEIN VIDEO MEHR BEWERTET.
BITTE BEWERTE DAS AKTUELLE VIDEO.

Fig. 3. Requesting a rating from the user (pops up after 5 minutes without rating activity)

3.3 Procedure

Participants first entered demographic details, after which they were shown an instruction on how to use the ClipClub system. They then used the system freely for at least 30 minutes. Users could perpetually rate items and inspect recommendations in any given order; they were thus able to reflect on their experience in deciding whether to provide further ratings (thereby allowing ample opportunity to let their feedback behavior be influenced by choice satisfaction and perceived system effectiveness). Our rating-probe asked users to rate the current item whenever 5 minutes passed without the user rating any items. Each user therefore provided at least 6 ratings, unless they ignored the rating-probe. The median number of ratings per user was 15. After the experiment, users completed our questionnaire, and entered an email address for the raffle.

3.4 Questionnaires

Post-experimental questionnaires consisted of 40 statements[2] to which participants could agree or disagree on a 5-point scale. The questionnaires were analyzed using Factor Analysis in two batches. The first analysis, analyzing 23 items with Alpha Factoring extraction and Oblimin rotation (delta = -0.5), provided three factors that explained 64% of the variance: recommendation set quality (7 items, e.g. “The recommended videos fitted my preference”), system effectiveness (6 items, e.g. “The recommender is useless”, reverse-coded) and choice satisfaction (9 items, e.g. “The

[2] The questionnaires are available from the authors upon request.

videos I chose fitted my preference”). One item was deleted due to low communality. The KMO statistic of sampling adequacy was 0.833; well above the prescribed minimum of 0.60.

The second analysis, analyzing 17 items with the same settings, provided three factors that explained 60% of the variance: general trust in technology (4 items, e.g. “I’m less confident when I use technology”, reverse-coded), system-specific privacy concern (5 items, e.g. “I feel confident that ClipClub respects my privacy”) and intention to rate (5 items, e.g. “I like to give feedback on the items I’m watching”). Three items were deleted having low communalities. The KMO was 0.730.

3.5 Process Data (Click Stream)

In order to link the subjective metrics developed in the questionnaires to observable behavior, all clicks in the interface were logged. We processed the logs for total viewing-time, total number of clicked and completed clips, number of self-initiated ratings, and number of canceled rating requests.

4 Results

Fig 4 shows the statistical results of our path model (Fig 1). The model has a good fit, with a non-significant χ^2 of 13.210 (df = 13, p = .4317), a CFI of .996, an RMSEA between 0 and 0.153 (90% confidence interval) and an SRMR of .094.

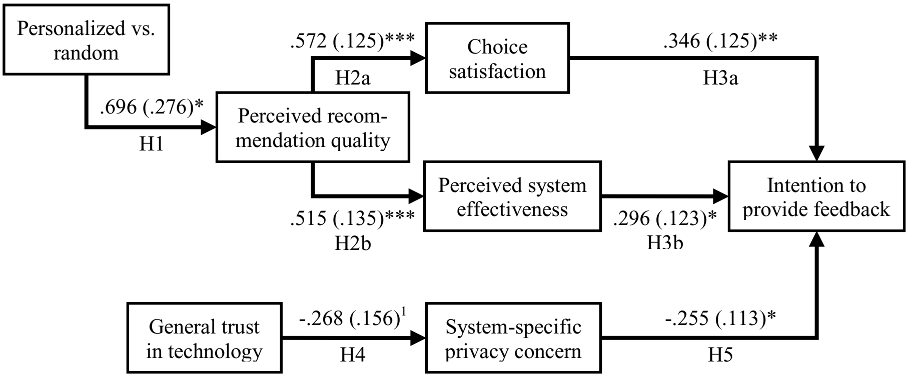


Fig. 4. Coefficients (and standard deviations) of regression between the concepts under study, as taken from our structural equation model (1 $p_{\text{one-sided}} < .05$, * $p < .05$, ** $p < .01$, *** $p < .001$)

Providing personalized recommendations (as compared to random recommendations) increases the perceived quality of the recommendations (H1), which in turn increases the choice satisfaction (H2a) and the system effectiveness (H2b). The process data provided further evidence for these hypotheses. The number of clips watched from beginning to end is significantly higher in the personalized condition than in the random condition (M = 15.17 vs. M = 10.15, t(43) = 2.642, p < .05, r = .37), indicating more consumption in the personalized condition. At the same time, the number of clicked clips and the total viewing time[3] were both negatively correlated with subjective system effectiveness (r = -.405, p < .01 and r = -.341, p < .05 respectively), showing that a higher system effectiveness is related to reduced browsing activity.

Users with a higher choice satisfaction and system effectiveness are more willing to provide feedback (H3a,b). The willingness to provide feedback decreases however when users have a higher system-specific privacy concern (H5), which in turn increases when users have a lower general trust in technology (H4). Consistent with this, the number of canceled rating probes (popping up after five minutes without rating) is significantly lower in the personalized condition than in the random condition (M = .33 vs. M = 1.22, t(33.94) = -2.398, p < .05, r = .38), and is also negatively correlated with intention to provide feedback (r = -.364, p < .05). However, the total number of provided ratings was not significantly correlated with users' intention to provide feedback, indicating the predicted intention-behavior gap.

5 Discussion

The results provide several insights in the user experience of recommender systems, the effect of personalized recommendations, and feedback intentions. Participants were able to notice the higher recommendation quality in the personalized condition. This in turn positively influenced the perceived system effectiveness and choice satisfaction. The increased number of finished clips in the personalized condition provides objective evidence that the manipulation was successful. Users that rate the system more effective show reduced browsing behavior as they click on fewer clips but at the same time watch more clips entirely from beginning to end. This complies with the finding that effective systems allow users to more selectively watch content [1].

Our results also provide an insight in the factors that influence the users' intention to provide feedback. Our path model shows that the intention to provide feedback is influenced by choice satisfaction, the perceived effectiveness of the system and the system-specific privacy-concern. In other words, initial privacy concerns can be overcome when users find out that providing feedback is beneficial. The higher number of canceled rating requests in the random condition indicates that users may refuse to provide feedback when this has no benefit. At the same time, however, we find an intention-behavior gap: a higher intention to provide feedback did not result in more actual feedback.

6 Future Work

Several directions for future work can be suggested that may solidify our results. For example, the confirmed benefit of personalized recommendations would stand on firmer ground if it would be confirmed in several other systems and with a higher number and a more diverse range of participants. The same holds for the privacy-usefulness tradeoff for providing feedback. Other aspects that may influence user experience and

[3] One outlier with an abnormally long viewing time was ignored for this analysis.

willingness to rate may be incorporated in future work to get a more detailed understanding of the mechanisms underlying the user-recommender interaction.

Our experiment found significant results comparing a system employing a recommender algorithm against a system that provides random 'recommendations'. The difference in accuracy between these two conditions is substantial; future work could test for similar differences comparing two algorithms that only moderately differ in accuracy.

Microsoft's business model for the ClipClub system is advertisement revenue. At intervals, ads are displayed at the beginning of a clip. Personalized recommendations decrease the number of clicked clips, which means that users may watch *fewer* ads, which would *reduce* advertisement revenue. In the current experiment there was however no significant decrease in viewed ads, and it is expected that in the long run higher system effectiveness will increase the number of return visits, which will eventually result in higher sustained revenue. A longitudinal study of the system may provide more insights in this matter.

Acknowledgements. We gratefully acknowledge the funding of our work through the European Commission FP7 project MyMedia (www.mymediaproject.org) under the grant agreement no. 215006. For inquiries please contact info@mymediaproject.org.

References

1. Xiao, B., Benbasat, I.: E-Commerce Product Recommendation Agents: Use, Characteristics, and Impact. Mis. Quarterly 31, 137–209 (2007)
2. McNee, S.M., Riedl, J., Konstan, J.A.: Being Accurate Is Not Enough: How Accuracy Metrics Have Hurt Recommender Systems. In: CHI'06 Extended Abstracts on Human Factors in Computing Systems, pp. 1097–1101. ACM, New York (2006)
3. Ziegler, C.N., McNee, S.M., Konstan, J.A., Lausen, G.: Improving Recommendation Lists Through Topic Diversification. In: Proceedings of the 14th International Conference on World Wide Web, pp. 22–32. ACM, New York (2005)
4. McNee, S.M., Riedl, J., Konstan, J.A.: Making Recommendations Better: an Analytic Model for Human-Recommender Interaction. In: CHI'06 Extended Abstracts on Human Factors in Computing Systems, pp. 1103–1108. ACM, New York (2006)
5. Knijnenburg, B.P., Meesters, L.M.J., Marrow, P., Bouwhuis, D.G.: User-Centric Evaluation Framework for Multimedia Recommender Systems. In: UCMedia 2009 Workshop Experience Design and Evaluation of Social UCM Applications (2009) (in press)
6. Cosley, D., Lam, S.K., Albert, I., Konstan, J.A., Riedl, J.: Is Seeing Believing?: How Recommender System Interfaces Affect Users' Opinions. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 585–592. ACM, New York (2003)
7. Herlocker, J.L., Konstan, J.A., Terveen, K., Riedl, J.T.: Evaluating Collaborative Filtering Recommender Systems. ACM Transactions on Information Systems 22, 5–53 (2004)
8. Kohrs, A., Merialdo, B.: Using Category-Based Collaborative Filtering in the Active WebMuseum. In: Proceedings of IEEE International Conference on Multimedia and Expo., pp. 351–354. IEEE Press, New York (2000)

9. McNee, S. M., Albert, I., Cosley, D., Gopalkrishnan, P., Lam, S.K., Rashid, A.M., Konstan, J.A., Riedl, J.: On the Recommending of Citations for Research Papers. In: Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work, pp. 116–125. ACM, New York (2002)
10. Torres, R., McNee, S.M., Abel, M., Konstan, J.A., Riedl, J.: Enhancing Digital Libraries With TechLens+. In: Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries, pp. 228–236. ACM, New York (2004)
11. Häubl, G., Dellaert, B.G.C., Murray, K.B., Trifts, V.: Buyer Behavior in Personalized Shopping Environments. In: Karat, C.M., Blom, J.O., Karat, J. (eds.), pp. 207–229. Springer, Netherlands (2004)
12. Kobsa, A.: Personalized Hypermedia and International Privacy. Communications of the ACM 45, 64–67 (2002)
13. Hui, K.L., Tan, B.C.Y., Goh, C.Y.: Online Information Disclosure: Motivators and Measurements. ACM Transactions on Internet Technology (TOIT) 6, 415–441 (2006)
14. Resnick, P., Varian, H.R.: Recommender Systems. Communications of the ACM 40, 56–58 (1997)
15. Metzger, M.J.: Privacy, Trust, and Disclosure: Exploring Barriers to Electronic Commerce. Journal of Computer-Mediated Communication (2006), `http://jcmc.indiana.edu/vol9/issue4/metzger.html`
16. Herlocker, J.L.: Explanations in Recommender Systems. In: CHI'99 Workshop on Interacting With Recommender Systems (1999), `http://www.patrickbaudisch.com/interactingwithrecommendersystems/WorkingNotes/JonHerlockerExplanationsInRecommenderSystems.pdf`
17. Chellappa, R.K., Sin, R.G.: Personalization Versus Privacy: An Empirical Examination of the Online Consumer's Dilemma. Information Technology and Management 6, 181–202 (2005)
18. Teltzrow, M., Kobsa, A.: Impacts of User Privacy Preferences on Personalized Systems. In: Karat, C., Blom, J.O., Karat, J. (eds.) Designing Personalized User Experiences in eCommerce, pp. 315–332. Springer, Netherlands (2004)
19. Ackerman, M.S., Cranor, L.F., Reagle, J.: Privacy in E-Commerce: Examining User Scenarios and Privacy Preferences. In: Proceedings of the 1st ACM Conference on Electronic Commerce, pp. 1–8. ACM, New York (1999)
20. Spiekermann, S., Grossklags, J., Berendt, B.: E-Privacy in 2nd Generation E-Commerce: Privacy Preferences Versus Actual Behavior. In: Proceedings of the 3rd ACM Conference on Electronic Commerce, pp. 38–47. ACM, New York (2001)
21. Brodie, C., Karat, C.M., Karat, J.: Creating an E-Commerce Environment Where Consumers Are Willing to Share Personal Information. In: Karat, C., Blom, J.O., Karat, J. (eds.) Designing Personalized User Experiences in eCommerce, pp. 185–206. Springer, Netherlands (2004)
22. Kobsa, A., Teltzrow, M.: Contextualized Communication of Privacy Practices and Personalization Benefits: Impacts on Users' Data Sharing and Purchase Behavior. In: Martin, D., Serjantov, A. (eds.) PET 2004. LNCS, vol. 3424, pp. 329–343. Springer, Heidelberg (2005)

Comparing Techniques for Preference Relaxation: A Decision Theory Perspective

Maciej Dabrowski[1] and Thomas Acton[2]

[1] Digital Enterprise Research Institute Galway
National University of Ireland Galway, Ireland
maciej.dabrowski@deri.org
[2] Business Information Systems Group
J.E. Cairnes School of Business & Economics
National University of Ireland Galway, Ireland
thomas.acton@nuigalway.ie

Abstract. This research proposes a decision aid based on a novel type of preference relaxation, which enables consumers to easily make quality choices in online multiattribute choice scenarios. In contrast to filtering and recommendation mechanisms that are a potential solution to this problem, our method combines decision theory with preference relaxation and enables consumers to consider high-quality alternatives they initially eliminated. We compare our approach with existing methods using a set of 2650 car advertisements gathered from a popular advertiser website. We discuss the potential impact of our method on decision quality and give an overview of implications for practitioners and researchers.

Keywords: Decision Theory, Recommender Systems, Preference Relaxation, eCommerce.

1 Introduction

Online stores tend to provide large numbers of products with a variety of features. Consumers making purchase decisions are often unable to evaluate all available alternatives in great depth, and so seek to reduce the amount of information processing involved[1]. To prevent information overload online retailers provide product search and filtering functionality, usually by requesting users to fill in a form asking about the requirements that a desired product has to satisfy (their preferences). This process is used, for example, when searching for a used car (http://carzone.ie/), or a flight (http://orbitz.com/) on popular websites, and is referred to as *preference-based search* [2] . Although such choice-based approaches are prevalent, both users and retailers can find them unsatisfying. One of the major reasons is that users are often not able to correctly transform their preferences into requirements using online forms [2], and thus they are rarely provided with the information they need.

In this paper we study the impact of a preference relaxation mechanism on consumer decision making, and implement it as a decision aid. We argue that

F. Buccafurri and G. Semeraro (Eds.): EC-Web 2010, LNBIP 61, pp. 217–228, 2010.

during the process of preference-based filtration of an initial, very large, set of product alternatives consumers can eliminate products they might later consider valuable. We introduce a method that uses preference relaxation to extend the initial value preference and to include initially filtered out alternatives of potential high utility for further consideration. As such, a consumer is able to revise her criteria, consider more products and choose a configuration she finds the most suitable, but which may not fully fit her initial preference. In this paper we describe a model of a decision aid implementing our method, and present results of a simulation-based study using 2650 car advertisements gathered from one of the most popular websites in Europe.

2 Theoretical Background

2.1 Information Filtering and Recommender Systems

Information Filtering techniques typically perform a progressive removal of non-relevant content based on the information in a user profile acquired either in an implicit (e.g. studying user behavior) or an explicit (e.g. asking user to state his preferences) manner. These techniques provide a theoretical foundation for building recommender systems [3] that enable content personalization - an important stream of research in e-commerce.

Numerous studies [4,2] use recommendations to improve consumer decision-making. Providing a consumer with a relevant (similar to their stated preferences) yet diverse (so that they can discover new opportunities and adjust their preference model) set of alternatives has become an important research problem [5]. According to the *Look-ahead* principle [2],"suggestions should not be optimal under the current preference model, but should provide high likelihood of optimality when an additional preference is stated". Furthermore, dynamism in user preferences [6] is a problem recognized in Recommender Systems research.

2.2 Preferences in Decision Theory

Assumptions that the decision maker can accurately state (and indeed bound) which levels within an attribute are acceptable versus unacceptable is a fundamental to a self-explicated approach [7]. Decision-makers (DM) often use a conjunctive evaluation of available alternatives in which all the alternatives that possess at least one attribute with unacceptable values are rejected from further consideration. Product search and filtering mechanism offered online adhere to that approach, and filter out all products that do not fully fulfil stated requirements. However, previous research indicates that decision makers tend to fail to fully adhere to the self-explicated approach. Klein [8] found that decision makers often fail to reject alternatives with attribute levels which they themselves had previously described as unacceptable, and showed that significant numbers of participants can choose an alternative described with at least one attribute level they initially indicated as completely unacceptable. Preference relaxation mechanisms may assist in alleviating this problem. Further, a decision aid supporting preference relaxation can be seamlessly integrated with the existing online shopping websites to improve consumer decisions. The rigidity of typical preference elicitation (filtering) mechanisms is a well-established problem [9] that can potentially lead to the elimination of all available products from consideration. Over-specification of consumer requirements leading to an empty result sets motivated research on similarity based-retrieval [10] and query (preference) relaxation [11]. The process of filtering involves the application of filtering rules (or restriction on attributes) to the items in the set to be filtered [11,9]. Consumer preferences are the key input for alternative pre-filtration as only alternatives that fully satisfy all provided preferences are presented to the user as a result to his query. Mirzadeh and Ricci proposed a mechanisms for preference relaxation for failing queries (producing an empty result set) [11]. However, they do not investigate the impact of the extent of relaxation on decision maker behaviour, and their method is applicable primarily to failing queries.

Our research differs from these approaches. First, we primarily focus on reduction of type I error by extending the preferences provided by a consumer (which, however, can lead to discovering alternatives that may lead to providing preference on additional attributes). Second, many of these approaches require prior knowledge or history of user interactions and preference models, which are not required in our approach. We argue that the decision aid proposed in this paper can increase the average quality of result sets presented to a user after filtration, and positively impact decision making.

3 The Decision Aid

You intend to buy a car priced between €7000 and €8000 with reasonable mileage (25000 to 75000 km). Would you be willing to pay slightly more (€8100) for a car with mileage lower than you expected (11000 km)? The ability to locate cars with such attribute values which, albeit out of the boundary ranges specified, may provide consumers with a better awareness of possible choices. The method proposed here enables consumers to consider products that would ordinarily be eliminated early in the selection process by falling outside rigid preferences. In the subsections below we discuss our approach in more detail and contrast it with common simple preference relaxation methods.

3.1 Edge Sets

Typically, preferences on numerical attributes are expressed using value ranges. As such, we allow the decision maker to specify his/her attribute value range preference for an i-th attribute as $d = (d_L, d_U)$ where d_L (d_U) indicates the lowest (highest) acceptable value for a given attribute. We now introduce softening variables e_U (upper) and e_L (lower), and a relaxation factor δ (where $e_i = \delta * d_i$), which enhance the filtering rule (value range) built based on attribute value preference p causing the filtering rule to be less restrictive. The alternatives that satisfy the less strict preference $d^* = (d_L - e_L, d_U + e_U)$ remain in the set and can be

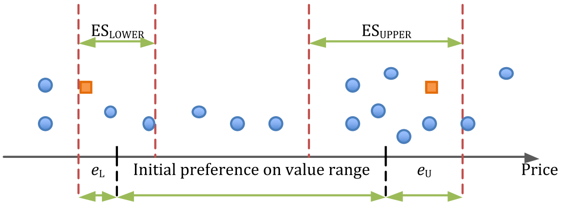


Fig. 1. An example of an Edge Set for a user price preference

considered by the DM. However, this approach, commonly referred to as Simple Preference Relaxation (SR), can significantly increase the number of alternatives presented to the user, resulting in information overload and increasing decision effort [12]. In order to prevent these negative effects we use approach based on a concept of Edge Set (ES). We conceptualize an Edge Set as a set of alternatives that fall into a value range based on the initial consumer value preference for a given attribute (see Fig. 1). For every preference value range two edge sets can be constructed (lower and upper), respectively: $ES_{LOWER} = (d_L - e_L, d_L + e_L)$ and $ES_{UPPER} = (d_U - e_U, d_U + e_U)$. We explain this concept using price range preference p_{PRICE} =(€3000, €4000). For example, assuming softening variables e_U =€200 and e_L = €150 (5% of respective preference interval boundaries' values) we can construct ES_{LOWER} =(€3000 - €150, €3000 + €150) resulting in ES_{LOWER} =(€2850, €3150) and ES_{UPPER} =(€4000 - €200, €4000 + €200) resulting in ES_{UPPER} =(€3800, €4200). Thus, ES_{LOWER} will contain cars that fall into the (€2850, €3150) price range.

3.2 Information Filtering Using Edge Sets

The inclusion of all alternatives satisfying the relaxed criteria would ordinarily increase the number of items presented to the DM, contributing to information overload. To address this issue we incorporate a selection mechanism into our relaxation method that includes only some of those cases (see Algorithm 1). First, we create edge sets (ES) based on relaxed preferences (e.g. $ES_{LOWER} = (d_L - e_L, d_L + e_L)$ for a lower preference boundary) using a selected δ (e.g. 0.05). Second, for every ES we identify the subset of all non-dominated alternatives (also referred to as the skyline [13]) that are part of this set. An item is non-dominated if no other item is better for any preference on attribute without being worse for at least one preference on other attributes [14]. If a non-dominated item is a member of an edge set and it does not satisfy the non-relaxed initial DM preferences (is not a member of $ResultSet_{NR}$) it is added to the set of *Suggestions*, as it may be found valuable. We define two methods for inclusion of *Suggestions* in the result set presented to a consumer. First, we propose to add suggestions to an initial result set constructed using a non-relaxed (NR) query. This method, further referred to as SBR_{ADD} (Soft Boundary Preference

```
Input: $Products$, $Preferences$, $\delta$, $Method$
Output: $ResultSet_{SBR}$
$SKYLINE \longleftarrow$ findSkyline($Products$);
$ResultSet_{NR} \longleftarrow$ filter($Products$, $Preferences$);
$PREF_{RELAXED} \longleftarrow$ relaxPreferences($Preferences$, $\delta$);
$SUGGESTIONS \longleftarrow \emptyset$;
$EdgeSet \longleftarrow$ filter($Products$,$PREF_{RELAXED}$) ;
foreach $Product\ p \in EdgeSet$ do
    if $p \in SKYLINE$ and $p \notin ResultSet_{NR}$ then
        $SUGGESTIONS \longleftarrow SUGGESTIONS \oplus p$;
    end
end
if $Method = ADD$ then
    $ResultSet_{SBR} \longleftarrow ResultSet_{NR} \oplus SUGGESTIONS$ ;
end
if $Method = REPLACE$ then
    $LowUtilSet \longleftarrow$ findLowUtil($ResultSet_{NR} \cap EdgeSet$, $|SUGGESTIONS|$);
    $ResultSet_{SBR} \longleftarrow ResultSet_{NR} \ominus LowUtilSet$;
    $ResultSet_{SBR} \longleftarrow ResultSet_{SBR} \oplus SUGGESTIONS$ ;
end
```

Algorithm 1. The Soft Boundary Preference Relaxation Mechanism

Relaxation with addition), may lead to increases in the size of result sets. To address this drawback and to prevent an increase in cognitive load we propose an alternative method. Instead of simple addition to the set, the method would replace dominated, low-utility items from a non-relaxed result set ($ResultSet_{NR}$) that belong to the EdgeSet, with high-utility alternatives. We refer to this method as SBR_{REP} (Soft Boundary Preference Relaxation with replacement). With this approach, the total size of the set is kept constant, and the alternatives with lowest utility according to current preference model (in this study we use the WADD model) are substituted with items from the skyline. We further refer to to these two mechanisms as SBR (Soft Boundary Preference Relaxation).

As indicated earlier, our method assumes variables e_U (upper) and e_L (lower), and a relaxation factor δ, which relax the value preference p. Selecting an appropriate value of δ is not trivial, as it resembles *closeness* (similarity of values) and can differ among consumers [15]. However, some studies [16,**?**] report that the maximum relaxation value δ_{max} should not be greater than $(3 - \sqrt{5})/2$, that is 0.382. Thus, the relaxation factor δ should be selected from the interval $[0, 0.382]$ to satisfy the concept of closeness [17]. Although Mirzadeh and Ricci [11] report that relaxation parameters are attribute-dependent and should be tuned according to consumer sensitivity to changes in that feature, in our study we implemented the former simpler relaxation approach to explore potential effects in the first instance, with a view towards possible expansion of parameters in future work. Although our approach is applicable to all types of attributes, in this study we investigate the methods that use numerical attributes as, commensurate with the literature [11], relaxation of binary and nominal constraints is trivial, as they are typically discarded during the relaxation process.

4 Hypotheses

Many dependent variables have been proposed as good indicators of the impact of decision aids on DM performance [18,19,1]. In our study we concentrate on three common measures, that is: decision quality, decision effort and diversity of a set of considered alternatives.

Previous studies [18,14] assess decision quality as a match between actual DM's choice from a set of alternatives and the "ideal selection" (a *non-dominated* alternative [14]). Hostler et al. [18] and Häubl and Trifts [14] have used such conceptualization of decision quality as a measure of decision performance. Our Soft Boundary Preference Relaxation method leads to better decisions by facilitating the consideration of a larger number of high quality alternatives by DMs. Consequently, compared to non-relaxing methods, we propose:

H1.1: Simple Preference Relaxation increases decision quality.

Similarly, our method should allow consumers to locate products that more closely match their preferences further improving decision quality:

H1.2: Soft Boundary Preference Relaxation increases decision quality.

The level of effort required to make a decision is another common decision performance indicator [1]. Ideally, the better support offered by a decision aid, the lower the cognitive effort required by a DM to make a decision. Effort is directly related to the amount of information that needs to be considered by a DM [20,12]. Intuitively, preference relaxation mechanisms increase effort by relaxing rigid requirements, and therefore incorporating more alternatives for consideration by a DM. We expect that our method will not lead to a significant increase in decision-making effort due to an increased number of products included for consideration. Compared to non-relaxing methods, we propose:

H2.1: Simple Preference Relaxation increases decision-making effort.

H2.2: Soft Boundary Preference Relaxation does not increase decision-making effort.

Selection of a product is considered context dependent, as the relative value of an option depends not only on the characteristics of that option, but also upon characteristics of other options in the choice set [21]. According to behavioral decision theory [1,22] the existence of such context impacts the perceived quality of available products. Indeed, Tversky [22] pointed out that in such contexts, DMs tend to adjust their initial preferences based on available choices, in contrast to maximizing pre-computed preferences. Further, the diversity of an RS is important in Recommender Systems research [23]. Consequently, we argue that preference relaxation mechanisms will increase the diversity of a result set.

H3: Soft Boundary Preference Relaxation increases result set diversity.

5 Evaluation

5.1 Dataset

The dataset consisted of 2650 used car advertisements collected from the most popular website in Ireland (http://carzone.ie/, a member of Autotrader media group). Additional attributes for used cars in the set not present in advertisements, such as reliability, were automatically generated using standard information retrieval methods based on product reviews collected from car review websites (e.g. whatcar.com). Generated attributes were classified as benefit-type and given scores ranging from 0 to 5 to resemble star ratings (e.g. 5 points for *maintenanceCost* describes the relatively lowest maintenance cost).

5.2 Method

Our experimental design was based on a *leave-one-out* (LOV) [10] approach in which we temporarily removed each alternative from the dataset and used its description as a DM preference. Based on user studies on importance of attributes in the used cars domain [24], and consistent with bounded rationality we chose 6 most popular attributes for our experiments. To best resemble user behaviour the preferences in our simulations were constructed similarly to filtering interfaces of the popular websites, where value preference intervals were selected to simulate possible user entries. Using the LOV approach, every used car advert in the set was temporarily removed from the set and its values were used to create preference values (based on available preference intervals). For example, a car at €3500 would be represented as a user search query with preference for price at (€3000-€4000). Simulations were run for combinations of 1 to 6 stated preferences and for relaxation factors 0.05, 0.1, 0.2, 0.3 and 0.382 (δ_{max}). Thus, for every set of parameters a maximum of 2650 *non-failing relaxed* queries were issued and relevant result sets were constructed for all four investigated methods: non-relaxed (NR), Simple Preference Relaxation (SR), Soft Boundary Preference Relaxation with Addition (SBR_{ADD}), and with Replacement (SBR_{REP}). Particular characteristics of these constructed result sets (see the next section) were assessed and compared to evaluate the methods under investigation.

5.3 Indicators

In our study a number of indicators were used to evaluate the four methods: non-relaxing (NR), Standard Preference Relaxation (SR), Soft Boundary Preference Relaxation with addition (SBR_{ADD}), and with replacement (SBR_{REP}).

Decision quality is a common indicator of performance. Häubl et al. [14] showed that the share of considered products that are non-dominated indicates the quality of a set of products considered by a consumer, which positively impacts decision quality. Conversely, we measured decision quality using a share of non-dominated alternatives present in the result set. Further, we note that decision quality is directly related to fulfilling particular DMs criteria for product selection (preferences) that can be measured by the utility of selected alternatives [4]. The

Table 1. Average: utility (AvgUtil), share of non-dominated alternatives in the result set (%ND), and result set size ($|RS|$) for relaxed (SR), non-relaxed (NR), SBR with addition (SBR_{ADD}) and replacement (SBR_{REP}) for number of stated preferences N

	N	1	2	3	4	5	6	Avg
AvgUtil	NR	0.3414	0.2792	0.2197	0.1851	0.1638	0.1498	0.1879
	SR	0.4533	0.4263	0.4123	0.3981	0.3869	0.3784	0.3975
	SBR_{ADD}	0.4892	0.4460	0.4029	0.3721	0.3509	0.3360	0.3730
	SBR_{REP}	0.5056	0.4541	0.4064	0.3735	0.3513	0.3357	0.3747
%ND	NR	12.25%	17.26%	16.62%	15.65%	14.91%	14.36%	15.58%
	SR	12.10%	18.87%	22.54%	24.18%	25.05%	25.57%	23.92%
	SBR_{ADD}	21.16%	43.77%	54.77%	59.48%	61.87%	63.27%	58.65%
	SBR_{REP}	25.37%	54.27%	65.46%	69.60%	71.47%	72.46%	68.65%
$\|RS\|$	NR	673.31	201.97	78.46	39.82	23.97	16.13	51.10
	SR	1057.82	494.97	291.61	213.15	175.39	154.08	229.73
	SBR_{ADD}	712.17	231.31	99.63	56.93	38.86	29.64	68.94
	SBR_{REP}	673.33	205.92	84.14	46.20	30.68	23.02	57.54
Diversity	NR	0.1560	0.1212	0.0913	0.0839	0.0754	0.0694	0.0837
	SR	0.2503	0.2134	0.1735	0.1661	0.1585	0.1537	0.1669
	SBR_{ADD}	0.1963	0.1812	0.1526	0.1463	0.1375	0.1311	0.1455
	SBR_{REP}	0.1945	0.1838	0.1542	0.1468	0.1379	0.1316	0.1463

higher the average utility of alternatives presented for choice, the more suitable options can be considered. Thus, we propose to measure decision quality using the average utility of a result set (where $AvgUtil \in [0, 1]$).

Information overload is an important factor that increases decision making effort and leads to changes in strategies employed by decision makers when selecting a product [20]. Following [21], we propose to measure decision making effort by the number of alternatives presented for consideration by a DM (that is, the size of a result set).

Vahidov [25] has indicated the importance of result set diversity in decision making. In our study we use a common conceptualization of normalized diversity ($diversity(ResultSet) \in [0, 1]$) that is inversely proportional to similarity, following the relation presented by Smyth and McClave [5]. We compute similarity using a law proposed by Shepard [26] stating that perceived similarity of items is related to their distance via an exponential function $sim(A, B) = e^{-distance(A,B)}$.

5.4 Results

We used related samples non-parametric tests to compare the average share of nondominated alternatives in the RS for queries using the preference relaxing mechanisms discussed in this study, with no preference relaxation. Results show that on average, RS constructed using relaxation contained significantly more non-dominated alternatives than the result sets constructed using no relaxation. In particular, we observed on average 23.92% (SR) of non-dominated alternatives in contrast to only 15.58% in case of non-relaxing methods (NR)

Table 2. Average: utility (AvgUtil), share of non-dominated alternatives in the result set (%ND), and result set size ($|RS|$) for relaxed (SR), non-relaxed (NR), SBR with addition (SBR_{ADD}) and replacement (SBR_{REP}) for different values of δ

	δ	0.05	0.1	0.2	0.3	0.382	Avg
AvgUtil	NR	0.2107	0.2009	0.1878	0.1758	0.1702	0.1879
	SR	0.3537	0.3752	0.3964	0.4171	0.4310	0.3975
	SBR_{ADD}	0.3327	0.3430	0.3623	0.3939	0.4204	0.3730
	SBR_{REP}	0.3369	0.3455	0.3637	0.3949	0.4206	0.3747
%ND	NR	17.47%	16.65%	15.57%	14.57%	14.12%	15.58%
	SR	24.65%	24.50%	24.07%	23.13%	23.49%	23.92%
	SBR_{ADD}	39.70%	46.05%	58.12%	69.16%	74.95%	58.65%
	SBR_{REP}	46.09%	54.65%	69.73%	80.92%	85.87%	68.65%
$\|RS\|$	NR	57.51	54.85	51.28	48.00	46.49	51.30
	SR	113.04	142.64	211.53	286.60	359.07	229.73
	SBR_{ADD}	63.34	63.59	67.22	71.50	77.06	68.94
	SBR_{REP}	58.33	56.38	55.50	56.76	60.48	57.54
Diversity	NR	0.0939	0.0895	0.0837	0.0783	0.0759	0.0837
	SR	0.1393	0.1497	0.1728	0.1808	0.1848	0.1669
	SBR_{ADD}	0.1182	0.1211	0.1445	0.1636	0.1718	0.1455
	SBR_{REP}	0.1169	0.1208	0.1461	0.1647	0.1743	0.1463

(see Table 1). Similar results were obtained for average utility of alternatives in a RS ($AvgUtil_{NR} = 0.1879$ and $AvgUtil_{SR} = 0.3957$)). These differences were statistically significant (p<0.001) thus confirming the hypothesis H1.1. Similarly, our results indicate that the use of the SBR mechanism improves the share of non-donimated alternatives in a result set in contrast to both non-relaxing (NR) and simple relaxation (SR) methods. We observed 58.65% (SBR_{ADD}), and 68.65% (SBR_{REP}) of non-dominated alternatives in contrast to 23.92% (SR) and 18.79% (NR). These differences were statistically significant ($p < 0.001$). Although the average utility of alternatives in a RS was similar to all preference relaxing methods with 0.3975 (SR), 0.3730 (SBR_{ADD}), and 0.3747(SBR_{REP}) the extent of improvement in the average share of non-dominated alternatives in a RS provides evidence for accepting H1.2.

The second group of hypotheses relates to the decision-making effort measured by a number of items from which DM has to select. For the methods investigated, we observed on average 229.73 (SR), 68.94 (SBR_{ADD}), and 57.54 (SBR_{REP}) items in the result set in contrast to only 51.10 items on average in a result set for non-relaxed queries (NR). These differences are statistically significant (p<0.001), confirming H2.1 and indicating rejection of H2.2. Finally, results indicate that the diversity of sets of alternatives generated using preference relaxation methods are more diverse than when no relaxation is used. In particular, we observed an average diversity of 0.1455 (SBR_{ADD}), 0.1463 (SBR_{REP}, and 0.1699 (SR) in contrast to 0.0837 for a non-relaxing method (NR) (see Table 2). These differences were statistically significant ($p < 0.001$), confirming H3. Results of the study are summarized in Table 3.

Table 3. The summary of results

Hypothesis		Result
H1.1	Simple Preference Relaxation increases decision quality	supported
H1.2	Soft Boundary Preference Relaxation increases decision quality	supported
H2.1	Simple Preference Relaxation increases decision-making effort	supported
H2.2	SBR does not increase decision-making effort	not supported
H3	SBR increases result set diversity.	supported

5.5 Discussion

Our study highlights the benefits of preference relaxation from a decision making perspective. First, we showed that preference relaxation methods lead to construction of result sets with a higher average utility ('usefulness') and a greater share of non-dominated alternatives. Furthermore, we demonstrated the positive impact of our method on the diversity of alternatives in result sets, which, according to [25], may lead to higher DM satisfaction. In addition, we showed that standard Preference Relaxation (SR) induces very significant increase in size of a result set, leading to an unacceptable increase in the decision-making effort. We proposed two variants of our method (SBR) that addresses this disadvantage. We demonstrate that our methods outperform the SR method and minimize the additional decision-making effort. In particular, for a low number of explicated preferences ($N < 3$), the difference in the size of a result set for SBR_{REP} and non-relaxing method (NR) is not statistically significant (see Table 1). Furthermore, when comparing SBR_{REP} and NR method, we observed 12.6% increase in the average size of a result set (57.54 and 51.10 respectively). However we found a large (340,6%) increase in the share of non-dominated alternatives (68,65% and 15.58% respectively). Moreover, we note that for low values of relaxation factor (e.g $\delta = 0.05$) we observed only 1.4% increase in the average size of the result set between SBR_{REP} (58.33 items) and NR (57.51 items) (see Table 2). On the other hand, results indicate a 163,8% increase in the share of non-dominated items (from 17.47% for NR to 46.09% for SBR_{REP}) and 59.9%

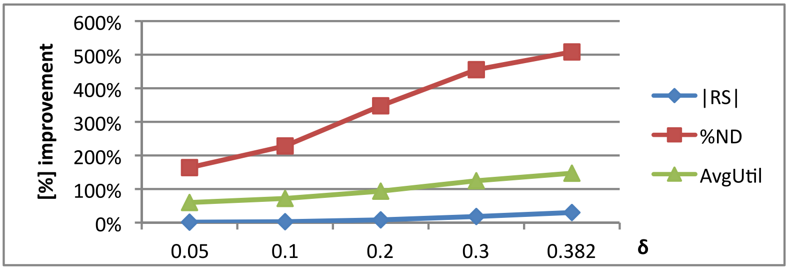


Fig. 2. Relative improvement in the size of a RS ($|RS|$), the share of non-dominated alternatives (%ND) and the average utility of alternatives in a set (AvgUtil) for SBR_{REP} compared with no relaxation (NR) for different values of δ

increase in average utility (from 0.2107 for NR to 0.3369 for SBR_{REP}). As such, we highlight the strong positive impact of SBR_{REP} on our decision-making indicators, with minimum negative impact on effort compared with other relaxation methods (see Fig. 2), and show that, overall, our method outperforms standard preference relaxation mechanisms.

6 Conclusions

This paper investigated the impact of preference relaxation on decision performance measures. We argued that during the process of filtering of the initial, very large set of products, consumers eliminate alternatives they could later consider, by providing inaccurate preferences for attributes and attribute values. In this paper we introduced a model for a decision aid based on preference relaxation that can limit the potentially negative effects of the dynamic preferences of consumers, addressing the limitations of existing methods. Moreover, we discussed the results of our experiments showing potential positive effect of preference relaxation on consumer decisions. The e-commerce application of our method may be highly beneficial to providers of online shopping services: diverse result sets may lead to more consumer satisfaction and potentially higher customer retention [25]. Moreover, increased average quality of the alternatives considered by a decision maker would reduce decision-making effort. This would have direct relevance to online consumers, as well as having value to e-commerce providers.

Acknowledgements. The work presented in this paper has been funded in part by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-2).

References

1. Payne, J., Bettman, J., Johnson, E.: The Adaptive Decision Maker. Cambridge University Press, Cambridge (1993)
2. Viappiani, P., Pu, P., Faltings, B.: Preference-based search with adaptive recommendations. AI Communications 21(2-3), 155–175 (2008)
3. Resnick, P., Varian, H.R.: Recommender systems. Communications of the ACM 40(3), 56–58 (1997)
4. Bridge, D., Ricci, F.: Supporting product selection with query editing recommendations. In: Proceedings of the 2007 ACM Conference on Recommender Systems. ACM, New York (2007)
5. Smyth, B., McClave, P.: Similarity vs. diversity. In: Aha, D.W., Watson, I. (eds.) ICCBR 2001. LNCS (LNAI), vol. 2080, p. 347. Springer, Heidelberg (2001)
6. Cao, H., Chen, E., Yang, J., Xiong, H.: Enhancing recommender systems under volatile userinterest drifts. In: CIKM '09: Proceeding of the 18th ACM Conference on Information and Knowledge Management, pp. 1257–1266. ACM, New York (2009)
7. Klenosky, D.B., Perkins, W.S.: Deriving attribute utilities from consideration sets: An alternative to self-explicated utilities. Advances in Consumer Research 19(1), 657–663 (1992)
8. Klein, N.M.: Assessing unacceptable attribute levels in conjoint-analysis. Advances in Consumer Research 14, 154–158 (1987)

9. Chaudhuri, S.: Generalization and a framework for query modification. In: Proceedings: 6th International Conference on Data Engineering, pp. 138–145. IEEE, Computer Soc. Press, Los Alamitos (1990)
10. McSherry, D.: Retrieval failure and recovery in recommender systems. In: 15th Artificial Intelligence and Cognitive Science Conference (AICS 2004), Castlebar, Ireland, pp. 319–338. Springer, Heidelberg (2004)
11. Mirzadeh, N., Ricci, F.: Cooperative query rewriting for decision making support and recommender systems. Applied Artificial Intelligence 21(10), 895–932 (2007)
12. Turetken, O., Sharda, R.: Development of a fisheye-based information search processing aid (fispa) for managing information overload in the web environment. Decision Support Systems 37(3), 415–434 (2004)
13. Borzsonyi, S., Kossmann, D., Stocker, K., IEEE Computer Society, I.C.S.I.C.S.: The skyline operator. In: 17th International Conference on Data Engineering, Heidelberg, Germany, pp. 421–430. IEEE Computer Society Press, Los Alamitos (2001)
14. Haubl, G., Trifts, V.: Consumer decision making in online shopping environments: The effects of interactive decision aids. Marketing Science 19(1), 4–21 (2000)
15. Motro, A.: Flex: A tolerant and cooperative user interface to databases. IEEE Transactions on Knowledge and Data Engineering 2(2), 231–246 (1990)
16. Bosc, P., Hadjali, A., Pivert, O.: Incremental controlled relaxation of failing flexible queries. Journal of Intelligent Information Systems 33(3), 261–283 (2009)
17. Ali, A.H., Dubois, D., Prade, H.: Qualitative reasoning based on fuzzy relative orders of magnitude 11(1), 9–23 (2003)
18. Hostler, R.E., Yoon, V.Y., Guimaraes, T.: Assessing the impact of internet agent on end users' performance. Decision Support Systems 41(1), 313–323 (2005)
19. Parra, J.F., Ruiz, S.: Consideration sets in online shopping environments: the effects of search tool and information load. Electronic Commerce Research and Applications 8(5), 252–262 (2009)
20. Eppler, M.J., Mengis, J.: The concept of information overload: A review of literature from organization science, accounting, marketing, mis, and related disciplines. Information Society 20(5), 325–344 (2004)
21. Payne, J.W., Bettman, J.R., Schkade, D.A.: Measuring constructed preferences: Towards a building code. Journal of Risk and Uncertainty 19(1-3), 243–270 (1999)
22. Tversky, A.: Contrasting rational and psychological principles in choice. In: Zeckhauser, R.J., Keeney, R.L., Sebenius, J.K. (eds.) Wise choices: decisions, games, and negotiations. Harvard Business Press, Boston (1996)
23. Bridge, D., Ferguson, A.: Diverse product recommendations using an expressive language for case retrieval. In: Craw, S., Preece, A. (eds.) ECCBR 2002. LNCS (LNAI), vol. 2416, pp. 43–57. Springer, Heidelberg (2002)
24. Dabrowski, M., Acton, T.: Improving consumer decision making through preference relaxation. In: IADIS Information Systems, March 18-20 (2010)
25. Vahidov, R., Ji, F.: A diversity-based method for infrequent purchase decision support in e-commerce. Electronic Commerce Research and Applications 4(2), 143–158 (2005)
26. Shepard, R.: Toward a universal law of generalization for psychological science. Science 237(4820), 1317–1323 (1987)

Detecting Leaders to Alleviate Latency in Recommender Systems

Ilham Esslimani, Armelle Brun, and Anne Boyer

KIWI Team, LORIA, Nancy University
615 rue du Jardin Botanique, 54600 Villers-Lès-Nancy, France
{ilham.esslimani,armelle.brun,anne.boyer}@loria.fr

Abstract. The exponential increasing of information on the Web and information retrieval systems engendered a heightened need for content personalization. Recommender systems are widely used for this purpose. Collaborative Filtering (CF) is the most popular recommendation technique. However, CF systems are very dependent on the availability of ratings to model relationships between users and generate accurate predictions. Thus, no recommendation can be computed for newly incorporated items. This paper proposes an original way to alleviate the latency problem by harnessing behavioral leaders in the context of a behavioral network. In this network, users are linked when they have a similar navigational behavior. We present an algorithm that aims at detecting behavioral leaders based on their connectivity and their potentiality of prediction. These leaders represent the entry nodes that the recommender system targets so as to predict the preferences of their neighbors about new items. This approach is evaluated in terms of precision using a real usage dataset. The results of the experimentation show that our approach not only solves the latency problem, it also leads to a precision higher than standard CF.

Keywords: recommender systems, usage analysis, behavioral networks, behavioral leaders, preference propagation.

1 Introduction

With the heightened development of the Web, of e-commerce applications and the exponential increasing of available information, the diffusion of personalized content is required. In this context, recommender systems are widely used for this purpose. Based on the observation of users' behavior, recommender systems predict the future preferences of users about a collection of items. Collaborative Filtering (CF) is one recommendation technique which consists in identifying similarities between users in order to generate recommendations of items to an active user [1].

CF and recommender systems are widely implemented in many application areas thanks to their reliability for personalization. Nevertheless, some research questions subsist and strangle recommender systems performance. Some of these questions concern the new item cold-start problem or latency problem [3]. Indeed, when a new item is introduced in the system, the preferences (the ratings) relating to this item are not available because no user has assigned a rating yet. Thus, recommender systems based on ratings, as CF systems, are unable to recommend new items to users.

F. Buccafurri and G. Semeraro (Eds.): EC-Web 2010, LNBIP 61, pp. 229–240, 2010.

In this paper, we propose an original approach to address the latency problem. We suggest to harness behavioral leaders in the context of recommender systems and behavioral networks [13]. In these networks, users are connected when they have a similar navigational behavior. Thus, we propose an algorithm that aims at detecting leaders. This algorithm relies on the identification of potential behavioral leaders based on their high connectivity in the behavioral network. Then, so as to detect actual leaders, leader preferences are propagated to their neighbors through the network and these propagated preferences are evaluated in terms of precision. The higher the precision ratio are, the more reliable the leaders are. These leaders have a prominent role in the behavioral network thanks to their important potentiality of prediction. They represent the entry nodes that the recommender system targets so as to predict user preferences about new items.

The remainder of this paper is organized as follows. In the second section related works regarding the latency problem and detection of leaders are presented. The third section focuses on the description of our algorithm for the detection of behavioral leaders. The fourth section is dedicated to the presentation of the experimentations we conducted so as to evaluate the performance of our approach. Finally, we summarize our research and conclude with some possible future directions.

2 Related Works

In this paper, our research focuses on the detection of leaders in the context of recommender systems. With the objective of alleviating latency problem, the issue regarding this research is to find a reliable method for detecting leaders. In the following subsections, as related works, we present some research studies examining the latency problem in the frame of recommender systems. We also focus on studies investigating the problem of detecting leaders.

2.1 Latency Problem

Recommender systems based on CF are confronted with the new item cold start problem, called also the latency problem. When a new item is introduced in the system, it cannot be used in collaborative recommendations as users' evaluation about this item are not yet available in the system. Let us notice that in general users do not devote much time to express regularly their preferences about new items [27].

To alleviate latency problem, content-based filtering and ontology-based filtering have been suggested as solutions. We present here an overview of these studies.

Content-based filtering has been widely used in several works as [20] and [6]. This technique relies on the analysis of the content of items so as to generate recommendations. Thus, a new item is recommended according to the similarity of its content with the other items. For example, a user who has already seen items about "genetics" will receive recommendations of new items related to this subject.

However, the problem of content-based filtering is the limitation of the recommendation diversity. Moreover, recommendations are usually overspecialized as the recommended items are always similar and identical to those already appreciated by the user.

Thus, the other items with a different content are neglected and are never integrated in recommendation lists suggested to this user.

To overcome this problem, content-based filtering is often combined with CF in the frame of hybrid recommenders [4,14] based notably on probabilistic models [24,26].

Furthermore, ontology-based filtering has been also developed and suggested as a solution to the latency problem. For example, this filtering technique has been used in Quickstep-Foxtrot (recommending research papers) [22] and Entree (recommending restaurants) [7]. Ontologies are used to automatically construct knowledge bases, then learning techniques are applied to classify items and generate user profiles.

The problem regarding such systems is the requirement of the availability of an ontology.

2.2 Detection of Leaders and Influencers

Leadership and influence propagation have been subject of many studies in the area of marketing, social science and social network analysis [15]. Researchers tend to understand how communities start, what are their properties, how they evolve, what are the roles of their members and how influencers and opinion leaders can be detected through these communities. Katz and Lazarsfeld [11] defined opinion leaders as ”the individuals who were likely to influence other persons in their immediate environment“.

The earliest studies of influence and leadership focused on the analysis of the propagation of medical and technological innovations [9]. More recently, [28] also examined this question and proposed diffusion models of innovations in networks.

In the area of marketing (viral marketing), influence propagation is often linked to the word-of-mouth phenomenon and its effects on the success of new products [10]. The most important challenge in marketing is how to find a small segment of the population (influencers or leaders) that can influence the other segments by their positive or negative opinions regarding products and services [29]. Keller and Berry [18] confirm the importance of influencers as they guide the decisions of a community and predict market trends. According to their study, “one American in ten tells the other nine how to vote, where to eat and what to buy”.

With the development of Internet, leaders and influencers do not use only traditional word-of-mouth, they can propagate their opinions based on interactive exchanges through blogs, forums, wikis and various social network platforms. Indeed, nowadays, social networks become the most important medium for propagating information, innovation and opinions.

Several recent studies have been interested in analyzing interactions and influences between entities and examining the impact of leaders in social networks. [19] study approximation algorithms for influence maximization in co-authorship network. [2] show how to identify active and non active influential bloggers that can lead trends and affect group interests in the blogosphere. [15] propose a pattern mining approach to discover leaders and to evaluate their influence in social networks. Actions such as tagging, rating, buying and blogging are considered in frequent pattern discovery. [15] consider in fact that in a social network, a leader can guide the trends of performing actions. Thus, friends are tempted to perform the same actions than the ones the leader performed.

Other studies investigated the role of network structure on the propagation of information and opinions. Some of them [5] [23] emphasize the role of highly connected nodes in a social network, called also hubs, in information dissemination and evolution of collaboration in this network. [21] confirm that highly connected nodes have an important influence on their neighbors. Keller and Berry [18] show also that users who influence others, have relatively large numbers of social links.

To our knowledge, in the context of recommender systems and CF, the detection of leaders and influencers has been examined in few studies. [8] present a recommendation system that selects opinion leaders by category to solve the new user problem in CF. To detect opinion leaders, this system uses a fuzzy inference system exploiting a marketing method called RFM (Recency, Frequency, Monetary). [25] define several metrics to measure the influence of users in rating based recommender systems. They propose a metric that measures the influence by removing some user's ratings while computing predictions and observing the effect of this removal on the recommendation results. If the difference is high, the user is detected as influential.

In this paper, by considering social networks approaches, we aim at detecting leaders among users so as to propagate their preferences about new items to the others with the objective of alleviating latency problem in recommender systems. The following section describes our approach.

3 Leader Detection and Preference Propagation in Behavioral Networks

Standard recommender systems and CF require a significant amount of rating data so as to evaluate similarities between users and compute recommendations. When an item is new, no or few ratings about this item is available yet. Therefore, the system cannot incorporate it in any list of recommendations. To alleviate this problem of latency, we propose to identify leaders through a behavioral network. Then, leader appreciations are propagated in this behavioral network so as to predict the preferences of neighbors about new items and eventually recommend these items to users.

The following subsections describe behavioral networks modeling and present the algorithm we propose to detect behavioral leaders.

3.1 Construction of the Behavioral Network

As presented in [13], a behavioral network is constructed based on behavioral information and users are linked as they share similar navigational patterns. This approach exploits behavioral data with the objective of assessing similarities between users. Behavioral data refers to usage traces that capture navigational activities and interactions of users on a given website.

We consider that two users u_a and u_b, who share common sequential patterns are similar. The longer the sequence of a common pattern is, the more the users are similar. Therefore, our goal is to identify for every pair of users (u_a, u_b), the maximum length $L_{Kmax}(u_a, u_b)$ of a navigational pattern among their navigational common patterns.

Then, the similarity of navigation between two users is computed by using formula (1) that takes into account the following parameters:

- Common patterns between the active user u_a and the neighbor user u_b.
- The maximum length of common patterns between the active user u_a and the neighbor user u_b. For example, if u_a and u_b have the common patterns $< i_1 i_5 >$ and $< i_3 i_2 i_9 >$, the maximum length of their common patterns is 3 corresponding to $< i_3 i_2 i_9 >$. 3 represents the number of items occuring in this navigational pattern.
- The maximum length of sessions of user u_a and the neighbor user u_b.

This formula computes, for each pair of users u_a and u_b the similarity of navigation $SimNav_{(u_a,u_b)}$ as the ratio of the maximum length of a common frequent pattern $L_{Kmax}(u_a,u_b)$ and the minimum of maximum sizes of u_a and u_b sessions denoted $SessMax_{(u_a)}$ and $SessMax_{(u_b)}$. Let us note that the common frequent pattern is intra-session.

$$SimNav_{(u_a,u_b)} = \frac{L_{Kmax}(u_a,u_b)}{\min(SessMax_{(u_a)}, SessMax_{(u_b)})} \tag{1}$$

We note that the similarity value is normalized between 0 and 1. This metric emphasizes the importance of the longest frequent patterns to evaluate similarities of users. The higher the length of a sequential pattern is, the more the users are similar.

Once navigational similarities are evaluated, we build the behavioral network by using a graph $G = (N, E)$ where nodes N represent users, edges E represent the links between users and the navigational similarities are the weights of the edges.

3.2 Detecting Leaders in a Behavioral Network

With the objective of alleviating the latency problem in recommender systems, we propose to detect behavioral leaders in the constructed behavioral network. Unlike CF, if ratings about new items are assigned by only one leader, our recommender system incorporate them in recommendations. The information from leaders about these items is sufficient. We propose in fact an algorithm that aims at detecting leaders so as to predict the preferences of other users in the network about new items.

In social networks, the detection of leaders relies on the analysis of the social links through the network. Here, we emphasize the role of behavioral links to identify behavioral leaders in a network.

According to [5] [23] [21] [18] mentioned in section 2.2, we define a behavioral leader as a user who is not only highly connected in the behavioral network, he has also a high potentiality of predicting the future preferences of other users. We assume in fact that a behavioral leader can propagate his preferences in the network. We propose to propagate appreciations with an attenuation factor. This factor is directly related to the similarity between users (the weights of the links). Indeed, when users are very similar, there is a high probability that they share the same appreciations about items.

In addition, in recommender systems the items recommended to an active user are those highly appreciated by his neighbors [12]. Thus, similarly, we assume here that the items that a behavioral leader can propagate (recommend), are the items he prefers. Since our model relies on usage traces, the estimation of user appreciations (“like” or “dislike” an item) is required. To distinguish preferred items from non preferred ones, we take into account two implicit parameters: frequencies of visiting an item and duration of visiting an item that we can compute based on extracted information from Web server log files [12].

Algorithm 1 represents the algorithm we propose for detecting behavioral leaders. This algorithm uses as input the graph $G = (N, E)$ modeling the behavioral network where nodes N represent users and edges E are the links between them. Our algorithm includes two main steps. Let us notice that each step considers a distinct set of items denoted I_{tr} and I_{ts}. I_{tr} refers to the items used (at the training step) to assess behavioral similarities and construct the behavioral network and I_{ts} represents the set of new items considered to validate the actual behavioral leaders (the test step). Obviously, there is no item in common between these two sets.

At the first step of the algorithm (function “SelectPotentialLeaders”), for each node u_a in the graph G, the connectivity (centrality degree) is computed as the number of links (neighbors) incident upon u_a. Then, TopN potential leaders U_{PL} are selected based on their high connectivity in the behavioral network.

At the second step of the algorithm (function “DetectLeaders”), for each potential leader $u_{pl} \in U_{PL}$, their preferred items are identified $I_{prf}(u_{pl}) \subset I_{ts}$. Then, as presented in formula (3), potential leader appreciations $apr(u_{pl}, i_j)$ about items i_j ($i_j \in I_{prf}(u_{pl})$) are propagated to their direct neighbors such as a propagated appreciation, denoted $papr(u_{pl}, i_j)$, from a leader u_{pl} to the neighbor node u_a about an item i_j is weighted by the coefficient $\alpha_{(u_a,u_{pl})}$. The weights α range from 0 to 1 according to the similarity between u_{pl} and u_a computed by formula (1).

Once appreciations are propagated to a neighbor u_a, they are evaluated in terms of precision using formula (4). This precision is calculated as the ratio between N_{rl} representing the number of recommended items that are relevant for u_a (that are really appreciated by him) and N_r representing the number of all recommended items (relevant and irrelevant), as described in Table 1 [17]. Then, as presented in formula (5), for each potential behavioral leader we evaluate the precision $P(u_{pl})$. This precision is calculated as the average of precisions computed over all his neighbors u_a. We note that m represents in formula (5) the number of u_{pl} neighbors.

Precision ratios highlight finally the actual behavioral leaders over the network. The higher the precision ratio is, the more reliable the behavioral leader is.

So, when the recommender system needs to generate recommendations about new items, the behavioral leaders detected by our algorithm are considered. Indeed, the recommender system recommends the new items to these leaders, as they represent the entry nodes in the behavioral network. Then, in case of positive preferences, these leaders push their preferences about these new items to their neighbors based on formula (3).

Table 1. Categorization of items based on the intersection between recommendation lists and real preferences

	Selected
Relevant	N_{rl}
Irrelevant	N_{ri}
Total	N_r

Algorithm 1. Detection of behavioral leaders

```
function SELECTPOTENTIALLEADERS
    for each node $u_a$ over the graph G do
        Evaluate “Centrality Degree ” $D_{(u_a)}$        ▷ denoted $|\Gamma_{(u_a)}|$
            $$D_{(u_a)} = |\Gamma_{(u_a)}| \qquad (2)$$
    end for
    Sorting Degrees $D$ of all nodes $N$ in a descending order
    return TopN potential behavioral leaders $U_{PL}$ with high centrality degrees
end function

function DETECTLEADERS
    for each potential behavioral leader $u_{pl} \in U_{PL}$ do
        Select preferred items $I_{prf}(u_{pl}) \subset I_{ts}$
        Select neighbor nodes
        for each selected neighbor $u_a$ do
            for each item $i_j \in I_{prf}(u_{pl})$ do
                Propagate appreciations $apr(u_{pl}, i_j)$ to $u_a$ such as:
                    $$papr(u_a, i_j) = \alpha_{(u_a, u_{pl})} * apr(u_{pl}, i_j) \qquad (3)$$
                Evaluate precision of each $papr(u_a, i_j)$ ▷ $papr(u_a, i_j)$ is relevant or not for $u_a$
            end for
            Evaluate precision of all propagated appreciations to $u_a$
                $$p = \frac{N_{rl}}{N_r} \qquad (4)$$
        end for
        Evaluate precision of the potential leader $u_{pl}$ as the average of precisions $p$ computed over all his neighbors
            $$P(u_{pl}) = \frac{\sum_{u_a=1}^{m} p}{m} \qquad (5)$$
    end for
    return TopN actual behavioral leaders $U_L$ with the best ratios of precision
end function
```

4 Experimentation

4.1 Dataset

So as to evaluate the performance of our approach, we use a real usage dataset extracted from the Intranet of Credit Agricole Banking Group, in particular the usage data relating to the Department of Strategies and Technology Watch. All the users are members of the Group and can access numerous resources such as: news, articles, faq, special reports, etc.

Thus, we exploit the usage data that represents the navigational activities of users on the Intranet. This data has been stored in Web server log files. It contains mainly information about user-ids, session-ids and time of starting and ending sessions. The studied dataset is related to 748 users and 3856 resources. It has been collected during 24 months. This dataset has been split into 80% and 20% corresponding respectively to training and test datasets.

So as to evaluate the quality of propagation of potential leader appreciations through the network, we extracted leaders preferences about items from the test set I_{ts}. As mentioned in section 3.2, we considered only positive appreciations of these leaders (the items that they like). Besides, the weights α are used as the attenuation factor at the propagation step.

4.2 Evaluation

The most commonly used evaluation metrics in the experimentation of recommender systems are accuracy and precision of predictions. The evaluation of precision can rely on statistical or decision support measures [17]. When it is statistical, an accuracy metric evaluates the deviation between predicted ratings and the real ratings that are actually assigned by users to items. When it is a measure of decision support, it evaluates the relevance of a set of recommendations by computing the proportion of items in a recommendation list that the user consider actually useful and relevant. Precision is widely used as a measure of decision support. It evaluates if a selected item is relevant [3]. A selected item is a recommended item contained in the test set. Let us notice that here binary preferences are considered to distinguish relevant items from non relevant ones. These preferences are estimated from usage traces based on frequency and duration of visiting items.

4.3 Results

This section aims at examining the effectiveness of our approach for detecting leaders in a behavioral network. Thus, in this experimentation, we evaluate the precision of propagated appreciations regarding each potential leader based on formulas (4) and (5).

Figures 1 and 2 present the distributions of the number of potential behavioral leaders according to the precision when we take respectively into account 10% and 20% of TopN potential behavioral leaders at the propagation step. Let us notice that for about 53% of TopN10 behavioral leaders and 49% of TopN20 behavioral leaders, precision cannot be evaluated due to one of the following reasons:

- The items recommended by a potential behavioral leader have not been viewed by their neighbors in the test set. Thus, we cannot examine if this potential leader is actual or not.
- The potential behavioral leader has no positive appreciations (in the test set I_{ts}). So, he cannot propagate his preferences to the neighbors.

We note that in the results presented here, these behavioral leaders are not considered.

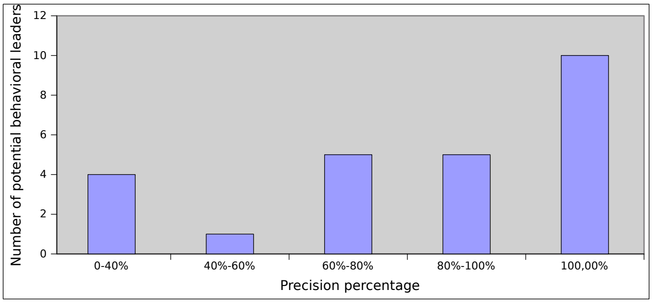


Fig. 1. Distribution of TopN10 potential behavioral leaders according to precision percentage

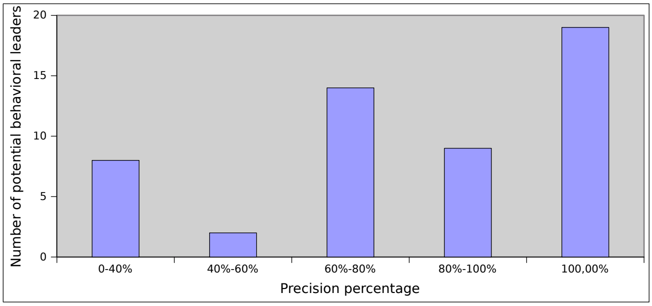


Fig. 2. Distribution of TopN20 potential behavioral leaders according to precision percentage

Now, by observing the results in Figure 1 and 2, we can see that precision distributions have a similar evolution for TopN10 and TopN20. TopN10 and TopN20 correspond respectively to about 53 and 101 potential behavioral leaders among all the users in the studied dataset. When TopN10 behavioral leaders are harnessed, we observe that 80% of these leaders have more than 60% of precision, 60% have a precision higher than 80% and 40% reach 100%.

Regarding TopN20 behavioral leaders, we can similarly see that about 80% leaders propagate accurate recommendations as the corresponding precision is greater than 60%, 53% have a precision greater than 80% and 37% reach 100% of accuracy.

So, when using either TopN10 or TopN20, an important proportion of potential behavioral leaders have a high precision of propagated appreciations. We consider that the leaders that reached a precision higher than 80%, represent the prominent nodes among all the nodes in the behavioral network. They can in fact predict accurately the preferences of the other users.

Moreover, in this experimentation we compare the performance of our approach to the standard CF [16] in terms of precision. Based on the "k Nearest Neighbors" approach, standard CF exploits the "Pearson Coefficient" to evaluate similarities between users. Then, neighbor users are involved in prediction generation.

Table 2. Precision average of Leaders based recommendations compared to the Standard CF

Recommendation Model	R_1	R_2
Leaders based recommendations	77%	76%
Standard CF	51%	43%

Table 2 presents the precision averages corresponding to our approach and to the standard CF. These precisions have been computed over the same pairs of $< user, item >$ when using two sets R_1 and R_2. These sets correspond respectively to the pairs of $< user, item >$ considered at the propagation step by TopN10 and TopN20 leaders.

By observing the results of Table 2, we can see that, on the set of items recommended by the leaders, our approach outperforms the standard CF as a greater accuracy is reached. Indeed, when we consider the sets R_1 and R_2, the precision is about 77%. At the opposite, the standard CF is less accurate as the precision averages reach only 51% and 43% when R_1 and R_2 are respectively considered. Thus, these results confirm the effectiveness of behavioral leaders regarding the recommendation of relevant items to the other users.

So, overall, the results presented here show the interest of our approach to detect reliable behavioral leaders in behavioral networks. These leaders have in fact an important potentiality of prediction as a high accuracy is reached by most of them. Thus, they represent the entry nodes in the behavioral network as they can predict efficiently the preferences of the other users about new items.

Nevertheless, considering the predictions generated by these leaders, our approach is confronted with the problem of coverage. Thus, the challenge will be to find a trade-off between improving accuracy of predictions and enhancing the coverage.

5 Conclusion and Future Work

In this paper we presented an original approach that aims at detecting leaders within the framework of recommender systems so as to alleviate latency problem. These leaders are harnessed in the context of a behavioral network. In this network, users are connected when they have similar navigational behaviors. The detection of leaders relies on their high connectivity in this behavioral network and their potentiality of prediction.

This approach is evaluated in terms of precision using a real usage dataset. The experimentation highlights the importance of considering TopN behavioral leaders through the network so as to predict the preferences of their neighbors about new items. Besides, the results show that our approach not only solves the latency problem, it also improves the performance of the recommender system. Indeed, a high accuracy is reached comparing to the standard CF when we consider the set of items recommended by the leaders.

As a future work, we plan to experiment additional datasets so as to validate the generalization of our approach. Moreover, we plan to solve the problem of coverage in the frame of our approach. Besides, it would be interesting to investigate other methods for detecting leaders and analyze the performance of the recommender system regarding the recommendation of new items.

Acknowledgment

We would like to thank M. Jean Philippe Blanchard, from the Credit Agricole Banking Group, for providing the dataset experimented in this paper.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art. IEEE transactions on knowledge and data engineering 17(6), 734–749 (2005)
2. Agarwal, N., Liu, H., Tang, L., Yu, P.S.: Identifying the influential bloggers in a community. In: Proceedings of the International Conference on Web Search and Web Data Mining (WSDM'08), pp. 207–218. ACM, New York (2008)
3. Anand, S., Mobasher, B.: Intelligent techniques for web personalization. In: Mobasher, B., Anand, S.S. (eds.) ITWP 2003. LNCS (LNAI), vol. 3169, pp. 1–36. Springer, Heidelberg (2005)
4. Balabanović, M., Shoham, Y.: Fab: content-based, collaborative recommendation. ACM Commun. 40(3), 66–72 (1997)
5. Barabasi, A.L., Jeong, H., Neda, Z., Ravasz, E., Schubert, A., Vicsek, T.: Evolution of the social network of scientific collaboration. Physica A 311(3-4), 590–614 (2002)
6. Billsus, D., Pazzani, M.: User modeling for adaptive news access. User-Modeling and User-Adapted Interaction 10(2-3), 147–180 (2000)
7. Burke, R.: Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction 12(4), 331–370 (2002)
8. Cheon, H., Lee, H.: Opinion Leader Based Filtering. In: Fox, E.A., Neuhold, E.J., Premsmit, P., Wuwongse, V. (eds.) ICADL 2005. LNCS, vol. 3815, pp. 352–359. Springer, Heidelberg (2005)
9. Coleman, J., Menzel, H., Katz, E.: Medical Innovations: A Diffusion Study. Bobbs-Merrill Co. (1966)
10. Domingos, P., Richardson, M.: Mining the network value of customers. In: KDD '01: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 57–66. ACM, New York (2001)
11. Elihu, K., Lazarsfeld, P.F.: Personal Influence; the Part Played by People in the Flow of Mass Communications. Free Press, New York (1955)
12. Esslimani, I., Brun, A., Boyer, A.: A collaborative filtering approach combining clustering and navigational based correlations. In: Proceedings of the 5th International Conference on Web Information Systems and Technologies (WEBIST), Lisbon, Portugal (2009)
13. Esslimani, I., Brun, A., Boyer, A.: From social networks to behavioral networks in recommender systems. In: Proceedings of The 2009 International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 143–148. IEEE Computer society, Los Alamitos (2009)
14. Good, N., Schafer, J.B., Konstan, J.A., Borchers, A., Sarwar, B., Herlocker, J., Riedl, J.: Combining collaborative filtering with personal agents for better recommendations. In: Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence (AAAI'99/IAAI'99), Menlo Park, CA, USA, pp. 439–446. American Association for Artificial Intelligence (1999)

15. Goyal, A., Bonchi, F., Lakshmanan, L.V.: Discovering leaders from community actions. In: Proceeding of the 17th ACM Conference on Information and Knowledge Management (CIKM'08), pp. 499–508. ACM, New York (2008)
16. Herlocker, J., Konstan, J., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99), pp. 230–237. ACM, New York (1999)
17. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. 22(1), 5–53 (2004)
18. Keller, E., Berry, J.: The influentials. Simon and Schuster Ed. (2003)
19. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03), pp. 137–146. ACM, New York (2003)
20. Lang, K.: Newsweeder: Learning to filter netnews. In: Proceedings of the 12th International Conference on Machine Learning (ICML'95), pp. 331–339 (1995)
21. Malcolm, G.: The Tipping Point: How Little Things Can Make a Big Difference. Little Brown, New York (2000)
22. Middleton, S.E., Shadbolt, N.R., Roure, D.D.: Ontological user profiling in recommender systems. ACM Trans. Inf. Syst. 22(1), 54–88 (2004)
23. Newman, M.: The structure and function of complex networks. SIAM Review 45, 167–256 (2003)
24. Popescul, A., Ungar, L., Pennock, D.M., Lawrence, S.: Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence (UAI'01), pp. 437–444. Morgan Kaufmann Publishers Inc., San Francisco (2001)
25. Rashid, A., Karypis, G., Riedl, J.: Influence in ratings-based recommender systems: An algorithm-independent approach
26. Schein, A., Popescul, A., Ungar, L.H., Pennock, D.M.: Methods and metrics for cold-start recommendations. In: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'02), pp. 253–260. ACM, New York (2002)
27. Sollenborn, M., Funk, P.: Category-based filtering and user stereotype cases to reduce the latency problem in recommender systems. In: Craw, S., Preece, A.D. (eds.) ECCBR 2002. LNCS (LNAI), vol. 2416, pp. 395–420. Springer, Heidelberg (2002)
28. Valente, T.W.: Network models of the diffusion of innovations. Hampton Press (1995)
29. Watts, D.J., Dodds, P.S.: Influentials, networks, and public opinion formation. Journal of Consumer Research 34(4), 441–458 (2007)

Electronic Markets, a Look Behind the Curtains: How Can Semantic Matchmaking and Negotiation Boost E-Commerce?

Tommaso Di Noia[1] and Azzurra Ragone[1,2]

[1] Politecnico di Bari – Via Orabona, 4, 70125 Bari, Italy
t.dinoia@poliba.it
[2] University of Trento – Via Sommarive, 14, 38100 Povo (Trento), Italy
ragone@disi.unitn.it

Abstract. In this paper[1] we present an overview of different semantic-based approaches to matchmaking and negotiation in electronic markets, showing how semantics can lead to a new generation of EC systems. We will introduce and briefly review different solutions to solve the two problems in the context of search/retrieval, multiattribute auctions, advertising, just to cite a few, showing the added-value provided by these techniques in so lively environments. The presentation range from strictly semantic-based approaches, to those combining logic languages with utility theory, to most recent ones relying on Semantic Web technologies and Linked Data datasets.

1 Electronic Markets

Automating commerce has enhanced the efficiency of online markets (a.k.a *electronic marketplaces* or *e-marketplaces*[2]) through reduction of transaction costs, improved matching of buyers and sellers and broadening the scope of trading relationships. An e-marketplace can be defined as *an interaction mechanism where the participants establish deals (trade) to exchange goods and services* using a standard currency [37]. Establish the scope of online marketplaces is not easy, as some would defines the entire Web as a giant marketplace where buyers and sellers find each other and trade using different rules and in different time. In this work we refer to sites or services providing a well-scoped environment where buyers and sellers can be matched and, subsequently, trade and, perhaps, ending the transaction with an exchange. The role of a marketplace is to support any or all phases in the lifecycle of a transaction. A commercial transaction, electronic or not, may be defined by three phases [16]:

1. **Connection** (*Discover*): the process of search and discovery of a counterpart to start a transaction.

[1] This paper is an extended abstract of the talk given by the authors at Ec-Web 2010.

[2] From now on, we refer to terms marketplace, e-marketplace and online market as synonyms.

F. Buccafurri and G. Semeraro (Eds.): EC-Web 2010, LNBIP 61, pp. 241–252, 2010.

2. **Deal** (*Negotiate*): the process of negotiating the terms of a deal.
3. **Exchange** (*Execute*): the execution of the terms of an agreed transaction.

Clearly, these three phases may be iterated or interleaved. The possibility of automation at each of these levels offers challenging opportunities for new models of economic interaction. Obviously, automation requires a clear definition of the problem and objectives, consequently, referring to the mechanism design literature, we define a marketplace as a *mechanism*, implemented by a mediator, and a set of agents who participate in the mechanism [33]. The mechanism is defined by a set of rules defining the permissible actions (*protocol*) and outcomes (*space of possible deals*), as a functions of agent actions (*strategies*). Usually, agents participating in the mechanism are self-interested, autonomous and rational. In this setting there are at least two design problems: firstly, design the market mechanism, i.e. define the protocol; secondly, design the agents that will interact w.r.t. the mechanism.

In this work we focus on the first two stages of a commercial transaction, namely discovery and negotiation. The automation of these phases is a challenging problem, with a plethora of different proposed solutions, but without a definitive one. In particular, we concentrate on knowledge-based approaches, aimed at taking advantage of the formal semantics of expressive logic-languages and related inferences to increase the effectiveness of both stages.

2 Discovery in E-Markets

Marketplaces should support discovery to the extent of enabling users to look among all the opportunities available at a site [37]. Traditional discovery services are electronic catalogues, keyword-based or hierarchical search facilities. More recent proposals refer to techniques borrowed from resource retrieval in the *Semantic Web* to perform matchmaking services between richer descriptions of goods/services offered and demanded. Indeed, Semantic-Based Resource retrieval addresses the problem of finding best matches to a request among available resources, where both the request and the resources are described adopting a shared interpretation of the knowledge domain the resource belongs to i.e., an ontology. The problem of semantic-based resource retrieval arises in several scenarios. Among them, personnel recruitment and job assignment, dating agencies, but also generic electronic marketplaces, web-services discovery and composition, resource matching in the Grid. All these scenarios share a common purpose: given a request, find among available descriptions those best fulfilling it, or “at worse”, when nothing better exists, those that fulfil at least some of the requirements. A challenge for B2C e-marketplaces is to match resources in the e-marketplace to potential buyer’s interests, but also to present available goods in an appealing manner, facilitating exploration and selection of product characteristics [9]. As pointed out by [35], selecting a product to buy in e-marketplaces is usually quite a frustrating experience: finding products best fitting users needs and/or financial capabilities often requires too much effort and time, spent browsing web sites or taxonomies in a web site. Especially when the searched product is not a perfectly defined item, users may have a vague idea of what they are actually looking for, being unaware of all the characteristics of the product. Searching for a product or service often requires domain knowledge that users do not have, so that many potential buyers tend to prefer traditional sales channels, such as physical stores where shop assistants can help the customer to make the right choice and answer to users requests or doubts.

A central issue in e-commerce is hence to support the user in the searching process of the products or services: converting site visitors to buyers in e-commerce environments is a recognized challenging subject [20].

The promise of the Semantic Web is to make information available on the web machine-understandable. By means of formal ontologies [14], modeled using Semantic Web languages such as RDF(S) [34] or OWL [19], the knowledge on specific domain can be modeled and exploited in order to make explicit the implicit knowledge, and reason on it thanks to the formal semantics provided by the representation language.

2.1 The Need for Semantics in the Matchmaking Process

Semantic web technologies open extremely interesting new scenarios, including: formalization of annotated descriptions that are machine understandable and interoperable, without being biased by usual drawbacks of natural language expressions; the possibility to reason on descriptions and infer new knowledge; the validity of the Open World Assumption[3] (OWA), overcoming limits of structured-data models. Furthermore, there are several issues that should not be underestimated: the annotation effort is considerable; computational complexity is often demanding also for simple reasoning tasks; interaction with semantic-based systems is often cumbersome and requires skills that most end users do not have –and are not willing to learn. The effort of annotation should be rewarded with inferences smarter than purely deductive services such as classification and satisfiability, which, although extremely useful show their limits in (real) approximate searches. Exact, or full, matches are usually rare [10] –and maybe these ones are not what a user really wants– and the true matchmaking process is aimed at providing one or more best available matches to be explored, thus leveraging further interaction. In this perspective also *missing* and *conflicting* information in the description of the resource (offer/supply) can be taken into account [7,11]. This can be aimed at better specifying the request, or modifying it, but also at initiating a negotiation/transaction process. We stress this point, as we believe that, as in textual information retrieval and in contrast with classic structured-data retrieval, the notion of relevance is central and must be taken into proper account. Obviously, the notion of resources relevance w.r.t a request calls for the definition of a ranking function, defining a partial or total

[3] The absence of a characteristic in the description of a resource to be retrieved should not be interpreted as a constraint of absence. Instead it should be considered as a characteristic that could be either refined later or left open if it is irrelevant for the user searching for the resource.

order of resources sorted w.r.t. the request, but also determine in a semantic-based way, which are the missing and/or conflicting information, in order to provide an explanation of results. In recent years Description Logics (DLs) [1] have been investigated by both the academic and industrial world as a formalism for Knowledge Representation. The interest in Description Logics shown by the two communities is more evident if we think that OWL, the standard language for ontology modeling in the Semantic Web, can be seen as an XML-based syntax for a particular DL. Modeling the information domain trough the formalism of a DL allows one to employ reasoning services provided by DLs to perform a knowledge-based search. Knowledge domain is modeled by means of formal ontologies, which resource descriptions refer to. The need for a common, shared, ontology is usually the main objection towards logic-based approaches to matchmaking. Nevertheless, it should be considered that even when requests and resources are expressed in heterogeneous forms, integration techniques [6] can be employed to make heterogeneous descriptions comparable.

We now briefly highlight the limitations of non-semantics-based approaches. First of all, we note that using standard relational database techniques to model a resource retrieval framework, there is a need to completely align the attributes of the offered and requested resources descriptions, in order to evaluate a match. If requests and offers are simple names or strings, the only possible match would be identity, resulting in an all-or-nothing approach to the retrieval process. Vague query answering, proposed by [18], was an initial effort to overcome limitations of relational databases, with the aid of weights attributed to several search variables. Vector-based techniques taken by classical Information Retrieval can be used, too, thus reverting the search for a matching request to similarity between weighted vectors of stemmed terms, as initially proposed in the COINS matchmaker [15] or in LARKS [36]. Such a formalization for resource descriptions makes matches only probabilistic, because descriptions lack of a document structure, causing strange situations to ensue.

Taxonomies are very useful to browse classes of items. Each node in a taxonomy can represent a set of items sharing a common characteristic. But, once this initial set of items has been found, it is not possible to use the taxonomy to refine the query.

Besides the above mentioned limitations, all these approaches lack of the possibility to deal with the semantics of the descriptions – both the user request and resources descriptions; a very useful feature in the search process. In taxonomy-based approaches a very basic semantic search (IS-A relation between category in the tree) is presented, but it results very weak. We believe that especially in e-marketplaces, the "meaning" of the terms rather than the terms themselves is very important. As a way of example, if a user is looking for a `safe car`, then a car endowed with `ABS system` and `airbags` would be a good choice. In order to catch these semantic correlation, ontologies would help our user in the search process. An ontology allows to relate terms with each other and give a formal model to the knowledge of the marketplace domain, and consequently express that a `safe car` is a `car` `endowed_with` an `ABS_system` and `endowed_with` `airbags`.

Exploiting the formal semantics of the language used to build an ontology, logic based inference processes can be performed, successfully dealing also with incomplete information (Open World Assumption). Based on such inference services an efficient retrieval process can be carried out.

Nevertheless, using standard deductive inference services only exact matches can be identified. Neither logical ranking nor explanation services on resources discarded during the search process are available, as the reasoning engine behaves as a boolean oracle.

We present how to overcome such limitations and how a semantic-enabled marketplace can provide valued-added services in terms of explanations to users requests, ranking of offers and request refinement, and that use of such systems can be made easy and immediate. In [8] we show how, using some non-monotonic inferences, namely Concept Abduction and Concept Contraction, it is possible to cope with incomplete and conflicting information, while providing logical explanations of results. In [31] a novel approach to matchmaking is presented, which mixes in a formal and principled way Datalog, fuzzy sets and utility theory, in order to determine most promising matches between prospective counterparts, *i.e.* what we name bilateral matchmaking. In [32] we take into account also fuzzy constraints to express preferences like *I would like a fast car.* In [27] we concentrate on *closed* marketplaces, *e.g.* electronic barter trade systems. Here, the focus is on how to find most promising matches, in a many-to-many matchmaking process, between bids (supplies/demands), taking into account not only the price and quantities as in classical barter trade systems, but also a semantic similarity among bid descriptions while keeping exchanges balanced.

3 Multi-attribute Negotiation

As in the discovery phase, also in the negotiation one automation is a challenging problem. Several recent research efforts have been focused on automated negotiation in various contexts, not only e-marketplaces, but also resource allocation settings, supply chain management and, in general terms, e-business processes. Negotiation mechanisms usually model resource and task allocation problems where issues to negotiate on are well established and defined in advance, e.g some online auctions. Many other negotiation mechanisms instead model e-marketplaces of undifferentiated products (commodities) where the only issues to negotiate on are *price* or *quantity*. Nevertheless there are a number of frameworks where agents have to reach an agreement on a product (car, house, etc.) or service (travel booking, wedding service, etc.) that can be described by many issues amenable to negotiation, and such issues may be not necessarily all established in advance. Moreover Buyer (Requester) and Seller (Provider) may be not necessarily interested in the same set of issues and may have different preferences on bundles of interrelated issues [21]. Obviously, if issues are not fixed there is the problem to express what agents "want" or "prefer". For instance, considering a car scenario, how to express a request for *a red sport car with GPS system and endowed with security features* or, conversely an offer for *a Lamborghini with satellite alarm, airbag and four year guarantee*? Is there any negotiation space? Can an agreement —in an automated way— be reached? Trying to answer to this and other questions, in this work we briefly outline several frameworks for multi-issue negotiation, with issues expressed and related to each other through an ontology. Indeed, when a potential buyer browses a car e-marketplace, she looks for a car fulfilling her needs and/or wishes, so not only the price is important, but also warranty or delivery time, as well as look, model, comfort and so on. In such domains it is harder to model not only the negotiation process, but also the request/offer descriptions, as well as finding the best suitable agreement [28]. Furthermore, preferences can refer to (1) bundle of issues, e.g *Sports car with navigator pack* where both the meaning of *sport car* and *navigator pack* are in the ontology; or preferences can be (2)*conditional* ones – when issues are inter-dependent i.e. the selection of one issue depends on the selection made for other issues – e.g. *I would like a car with leather seats if its color is black*. In such cases some kind of logical theory (ontology), able to let users express their needs/offers, could surely help [22]. Also, when descriptions refer to complex needs, we should take into account **preferences**, distinguishing them from hard mandatory constraints –**strict requirements**–, e.g., *I would like a black station wagon, preferably with GPS system*[4]. The possibility to handle some of the above mentioned issues in some electronic facility may help not only in the discovery/matchmaking stage of a transaction process, thus selecting most promising counterparts to initiate a negotiation, but also in the actual negotiation stage. In [28] we show how it is possible to express user's preferences through the help of a logic language and how to model a multi-attribute utility function on logic formulas [26]. We point out that in our frameworks we do not leave aside the analysis of all economic properties that a negotiation mechanism has to satisfy, such as efficiency, equilibrium, individual rationality of the agents participating in the mechanism; as well as the computational properties. A first attempt to model very simple negotiation scenarios with logic is proposed in [23], eventually in [28] we exploit propositional logic with concrete domains to model a one-shot bilateral negotiation. Then we introduce two negotiation mechanisms with partial [24] and incomplete information [25] using two different DLs-based negotiation protocols. In [29,30] we explore multilateral negotiation, presenting an auction-based mechanism exploiting DLs to elicit and represent non-additive preferences.

4 Semantic Web and E-Commerce

The approaches outlined in the previous sections offer to users smart inference services, but at cost of having the domain knowledge modeled through an ontology, that should be constantly updated as the domain evolves in time. In lively

[4] Strict requirements, in contrast with preferences, are constraints the buyer and the seller want to be necessarily satisfied to accept the final agreement, while preferences are issues they may accept to negotiate on.

and open environments to overcome such limitations the Linked Open Data initiative [2] is gaining momentum. The idea behind Linked Open Data is to allow users linking data and simplifying the publication of new interconnected data on the Web. It proposes a new method of exposing, connecting and sharing data through deferenceable URIs on the Web. The goal is to extend the Web by publishing various open datasets as RDF triples on the Web and by setting RDF links between data items from different data sources.

DBpedia [3] is one of the main dataset of the Linked Open Data graph. It is the machine-understandable equivalent of Wikipedia project. It is possible to ask queries against DBpedia (through its SPARQL endpoint http://dbpedia.org/sparql), and link other data sets on the web to DBpedia data. Currently the DBpedia dataset (version 3.5.1[5]) contains almost three million and half resources, including more than three hundred thousand persons, over four hundred thousand places, thousands of films, companies, music albums, etc.. All this information is stored in RDF triples. The whole knowledge base consists of over one billion triples. DBpedia labels and abstracts of resources are stored in 92 different languages. The graph is highly connected to other RDF dataset of the Linked Open Data cloud. DBpedia has many strong points over existing knowledge bases: it is spread over many domains; being based on Wikipedia, it represents a real community agreement; it follows the changing in Wikipedia, so it is continuously updated; it is multilingual. Moreover DBpedia has a central role in the Linked Open Data community effort: it is one of the central interlinking-hubs of the emerging Web of Data, inducing data providers to link their RDF datasets to DBpedia.

From an inference point of view, it is worth noticing that information within the DBpedia knowledge base is classified with respect and OWL ontology containing about 250 classes and 1200 properties. This gives a more formal perspective to the whole dataset.

4.1 Semantic Tag Retrieval for Online Advertising as a Matchmaking Problem

One of the main problem in online advertising is to display ads which are relevant and appropriate w.r.t. what the user is looking for. Often search engines fail to reach this goal as they do not consider semantics attached to keywords. Web advertising relies on sophisticated statistical analysis on plain and structured text. Furthermore, these techniques often do not take into account semantic relations among keywords, displaying ads that are sometime not relevant w.r.t. what the user is looking for or the text of the web page where the ad is placed (e.g. ads about a *zoo* for a page talking about *Tiger Woods*). The simplicity of this approach has several drawbacks: considering only a lexicographic approach and discarding the semantics of phrases does not allow to face problems such as synonymy, polysemy, homonymy, context analysis, nor to discover particular relations as hyponymy and hyperonymy. If we consider only a string-based

[5] http://wiki.dbpedia.org/Downloads351

analysis, we can not *semantically expand* both queries and ads. As a result, if two objects (e.g., an *ad* and a *query*, or an *ad* and a *web page*) use different collections of words to represent similar topics, they will be assigned to different clusters and will not match, although the meaning conveyed by both objects is related.

Moreover, advertisers lose a lot of potentially interested users because they do not exploit all the possible combinations of (semantically related) keywords and phrases that could be used by the users in a query to a search engine. Currently, the process of ads generation is quite tedious: one of the most boring task is the selection of keywords and bid phrases that activate a given ad. This means that a large slice of clients/customers is usually neglected because of the difficulty required by the creation of a successful marketing campaign.

The next step in computational advertising is finding a new technology able

(i) to improve the relevance/appropriateness of the ads in order to better capture the user's attention;

(ii) to ease the process of ad creation allowing the advertiser to establish powerful campaigns in a simplified way.

In the last years, several works [5,12,4] have been proposed to overcome the above mentioned problems. Many of them identified a possible solution in the adoption of external sources of information. By using a well structured information source as external domain knowledge (a taxonomy or an ontology), the proposed solutions tend to classify an ad, a query or a web page, in a set of resources belonging to this external knowledge, which are linked to other resources according to precise relationships. This allows to add semantics to objects traditionally analysed just on a syntactic/textual base.

Due to its wide knowledge coverage and its regular update, DBpedia seems to be the most suitable choice in such environments. The main idea of the approach presented in [17], is the following: keywords can be mapped to corresponding DBpedia resources. After this mapping, we are able to associate a well defined semantics to keywords and we can enrich the "meaning" of the keywords by exploiting the ontological nature of DBpedia. We associate a set of semantically related resources to each keyword mapped to a DBpedia resource.

The problem we address is twofold: (i) in the ad selection process in a sponsored search we need to find ads whose related keywords are semantically related to the query; (ii) in the keyword selection during the creation of the advertising campaign, the advertiser needs to add new tags whose meaning (semantics) is related to the one of the keywords she originally selected. The system described in [17] (*Not Only Tag – NOT*)[6] aims to overcome the above issues by enriching a keyword/tag with semantic information coming from DBpedia. The system makes possible to discover the *meaning* of a query in order to show to the user those ads that are in the same context (and meaning) of the query itself. Moreover, *NOT* is really useful also to help advertisers preparing their campaigns.

[6] A working prototype of *Not Only Tag* (*NOT*) is available at `http://sisinflab.poliba.it/not-only-tag`

If the advertiser is willing to promote a web site on *Luxury cars*, she could be suggested that also *BMW M5* and *Jaguar FX* are suitable keywords for the campaign, with (possibly) a different semantic similarity degree.

5 Discussion

We believe that now Semantic Web technologies have reached a degree of maturity such that they can really boost e-commerce, providing added value to both actors involved in the commercial transaction: customers and sellers.

Since the advent of the Internet, electronic commerce has witnessed a steady growth, both in terms of exchanges and of protocols/approaches supporting the various means and stages of electronic transactions. Nevertheless, as e-commerce becomes more and more pervasive in everyday life new challenges and issues arise. In particular, within all those scenarios where we deal with unstructured or semistructured information, new techniques and technologies are needed in order to represent and manipulate such informative contents. On the other hand, well-known frameworks based on simple attribute/value models are often not sufficient to the informative richness needed for effectively and efficiently deciding on the outcome of a commercial transaction. Using knowledge representation techniques and Semantic Web technologies, it is possible to perform an effective semantic-based matchmaking process (discovery), in order to find most promising candidates, and to design logic-based negotiation mechanisms exploiting the semantics of annotations and enhancing bid expressiveness. The leading thread of this work has been in fact the exploitation of knowledge representation techniques to annotate resources, model preferences and good/service descriptions and —more important— automatically reason about them, using both classical and non-monotonic inference services in innovative ways. Using a semantic-based language has been shown useful to model not only user's preferences, but, thanks to ontology modeling, also relations among issues.

The main points against semantic-based technologies and the Semantic Web vision has always been: *Who will annotate resources? Who will create and maintain ontologies and semantic-datasets on the Web?*. The answer to the first question emerged from Web 2.0. The current social web showed that people/companies are willing to tag pages and resources as well as to share their tags since they see a benefit in the annotation mechanism. Retrieval and clustering of web resources is much easier if they expose a summary of their content as a set of user-defined keywords. On the other side, the Linked Open Data initiative provided an actual answer to the problem of community-aware ontologies creation and maintenance. RDF datasets in the Linked Open Data cloud covers many knowledge domains and are tightly connected with each other. They are publicly available online and maintained by the community. One of the most relevant dataset is surely DBpedia exposing in a structured RDF format all the information available on its "unstructured sister" Wikipedia. A very interesting initiative linking e-commerce and the Semantic Web is the GoodRelations [13] ontology. This is a OWL DL ontology tailored at semantically describing products on the Web. It covers the representational needs of typical business scenarios for commodity products and services. In the next future e-marketplaces, we see product and service descriptions structured with respect to ontologies such as GoodRelations and whose informative content exploits data coming from the Linked Open Data cloud.

In the e-commerce scenario an important role is played by computational advertising, a field where semantic technologies are not yet fully exploited. However, as we have briefly outlined, the potential of the Semantic Web could also enhance this field making it more and more powerful, helping advertisers to perform very effective campaigns that show ads to potential interested customers in a more efficient way.

Acknowledgements

The authors are very grateful to all the people at Information Systems Laboratory (SisInf Lab) at Politecnico di Bari, Umberto Straccia and Michael Wellman. Part of the results in this paper are a joint work with them. The authors acknowledge partial support of Strategic Apulia projects PS_121, PS_092 and PS_025.

References

1. Baader, F., Calvanese, D., Mc Guinness, D., Nardi, D., Patel-Schneider, P.: The Description Logic Handbook. Cambridge University Press, Cambridge (2002)
2. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. International Journal on Semantic Web and Information Systems 5(3), 1–22 (2009)
3. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: Dbpedia - a crystallization point for the web of data. In: Web Semantics: Science, Services and Agents on the World Wide Web (July 2009)
4. Broder, A.Z., Ciccolo, P., Gabrilovich, E., Josifovski, V., Metzler, D., Riedel, L., Yuan, J.: Online expansion of rare queries for sponsored search. In: Proceedings of the 18th International Conference on World Wide Web (WWW 2009), pp. 511–520 (2009)
5. Broder, A.Z., Fontoura, M., Josifovski, V., Riedel, L.: A semantic approach to contextual advertising. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2007), pp. 559–566 (2007)
6. Calì, A., Calvanese, D., Giacomo, G.D., Lenzerini, M.: Data integration under integrity constraints. Information Systems 29(2), 147–163 (2004)
7. Colucci, S., Coppi, S., Di Noia, T., Di Sciascio, E., Donini, F.M., Pinto, A., Ragone, A.: Semantic-based resource retrieval using non-standard inference services in description logics. In: Proceedings of the 13th Italian Symposium on Advanced Database Systems– Sistemi Evoluti per Basi di Dati (SEBD-2005), pp. 232–239 (2005)
8. Colucci, S., Di Noia, T., Di Sciascio, E., Donini, F.M., Ragone, A.: Knowledge Elicitation for Query Refinement in a Semantic-Enabled E-Marketplace. In: Kishino, F., Kitamura, Y., Kato, H., Nagata, N. (eds.) ICEC 2005. LNCS, vol. 3711, pp. 685–691. Springer, Heidelberg (2005)
9. Colucci, S., Di Noia, T., Di Sciascio, E., Donini, F.M., Ragone, A., Rizzi, R.: A semantic-based fully visual application for matchmaking and query refinement in b2c e-marketplaces. In: Proceedings of the 8th International conference on Electronic Commerce, ICEC'06, pp. 174–184. ACM Press, New York (2006)

10. Colucci, S., Di Noia, T., Pinto, A., Ragone, A., Ruta, M., Tinelli, E.: A nonmonotonic approach to semantic matchmaking and request refinement in e-marketplaces. International Journal of Electronic Commerce 12(2), 127–154 (2007), http://mesharpe.metapress.com/link.asp?id=l77n007v82t13r10
11. Colucci, S., Di Noia, T., Pinto, A., Ragone, A., Ruta, M., Tinelli, E.: A nonmonotonic approach to semantic matchmaking and request refinement in e-marketplaces. International Journal of Electronic Commerce 12(2), 127–154 (2007)
12. Gabrilovich, E., Broder, A.Z., Fontoura, M., Joshi, A., Josifovski, V., Riedel, L., Zhang, T.: Classifying search queries using the web as a source of knowledge. TWEB 3(2) (2009)
13. Hepp, M.: GoodRelations: An ontology for describing products and services offers on the web. In: Gangemi, A., Euzenat, J. (eds.) EKAW 2008. LNCS (LNAI), vol. 5268, pp. 329–346. Springer, Heidelberg (2008)
14. Horrocks, I.: Owl: A description logic based ontology language. In: Proceedings of 21st International Conference on Logic Programming, pp. 1–4 (2005)
15. Kuokka, D., Harada, L.: Integrating Information Via Matchmaking. Journal of Intelligent Information Systems 6(2) (1996)
16. MacKie-Mason, J., Wellman, M.: Automated markets and trading agents. In: Tesfatsion, L., Judd, K.L. (eds.) Handbook of Computational Economics, Agent-Based Computational Economics, vol. 2. North-Holland, Amsterdam (2006)
17. Mirizzi, R., Ragone, A., Di Noia, T., Di Sciascio, E.: Semantic tags generation and retrieval for online advertising. Technical report (2010)
18. Motro, A.: VAGUE: A User Interface to Relational Databases that Permits Vague Queries. ACM Trans. Office Inf. Syst. 6(3), 187–214 (1988)
19. OWL. Web Ontology Language (2004), http://www.w3.org/TR/owl-features/
20. Pu, P.H.Z., Kumar, P.: Evaluating example-based search tools. In: Proceedings of 5th ACM Conference on Electronic Commerce, pp. 208–217 (2004)
21. Ragone, A.: Logic as a power tool to model negotiation mechanisms in the semantic web era. In: Proceedings of KnowledgeWeb PhD Symposium @ European Semantic Web Conference ESWC 2007, CEUR Workshop Proceddings, vol. 275, pp. 46–51 (2007)
22. Ragone, A.: Owl-dl as a power tool to model negotiation mechanisms with incomplete information. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 941–945. Springer, Heidelberg (2007)
23. Ragone, A., Di Noia, T., Di Sciascio, E., Donini, F.: A logic-based framework to compute pareto agreements in one-shot bilateral negotiation. In: Proc. of ECAI'06, pp. 230–234 (2006)
24. Ragone, A., Di Noia, T., Di Sciascio, E., Donini, F.M.: Alternating-offers protocol for multi-issue bilateral negotiation in semantic-enabled marketplaces. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 395–408. Springer, Heidelberg (2007)
25. Ragone, A., Di Noia, T., Di Sciascio, E., Donini, F.M.: Description logics for multi-issue bilateral negotiation with incomplete information. In: Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence (AAAI-07), pp. 477–482 (2007)

26. Ragone, A., Di Noia, T., Di Sciascio, E., Donini, F.M.: Extending propositional logic with concrete domains in multi-issue bilateral negotiation. In: Baldoni, M., Son, T.C., van Riemsdijk, M.B., Winikoff, M. (eds.) DALT 2007. LNCS (LNAI), vol. 4897, pp. 211–226. Springer, Heidelberg (2008)
27. Ragone, A., Di Noia, T., Di Sciascio, E., Donini, F.M.: Increasing bid expressiveness for effective and balanced e-barter trading. In: Baldoni, M., Son, T.C., van Riemsdijk, M.B., Winikoff, M. (eds.) DALT 2008. LNCS (LNAI), vol. 5397, pp. 128–142. Springer, Heidelberg (2009)
28. Ragone, A., Di Noia, T., Di Sciascio, E., Donini, F.M.: Logic-based automated multi-issue bilateral negotiation in peer-to-peer e-marketplaces. Autonomous Agents and Multi-Agent Systems Journal (JAAMAS) 16(3), 249–270 (2008)
29. Ragone, A., Di Noia, T., Di Sciascio, E., Donini, F.M., Wellman, M.: Computing utility from weighted description logic preference formulas. In: Baldoni, M., Bentahar, J., van Riemsdijk, M.B., Lloyd, J. (eds.) DALT 2009. LNCS, vol. 5948, pp. 158–173. Springer, Heidelberg (2010)
30. Ragone, A., Di Noia, T., Donini, F.M., Di Sciascio, E., Wellman, M.: Weighted description logics preference formulas for multiattribute negotiation. In: Godo, L., Pugliese, A. (eds.) SUM 2009. LNCS (LNAI), vol. 5785, pp. 193–205. Springer, Heidelberg (2009)
31. Ragone, A., Straccia, U., Di Noia, T., Di Sciascio, E., Donini, F.M.: Fuzzy matchmaking in e-marketplaces of peer entities using datalog. Fuzzy Sets and Systems 160(2), 251–268 (2009)
32. Ragone, A., Straccia, U., Di Sciascio, T.E., Donini, F.M.: Towards a fuzzy logic for automated multi-issue negotiation. In: Hartmann, S., Kern-Isberner, G. (eds.) FoIKS 2008. LNCS, vol. 4932, pp. 381–396. Springer, Heidelberg (2008)
33. Rosenschein, J., Zlotkin, G.: Rules of Encounter. MIT Press, Cambridge (1994)
34. Guha, D.R.V.: RDF Vocabulary Description Language 1.0: RDF Schema (2004), http://www.w3.org/TR/rdf-schema
35. Sacco, G.M.: The intelligent e-store: Easy interactive product selection and comparison. In: Proceedings of the 7th IEEE International Conference on E-Commerce Technologies, pp. 240–248 (2005)
36. Sycara, K., Widoff, S., Klusch, M., Lu, J.: LARKS: Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace. Autonomous agents and multi-agent systems 5, 173–203 (2002)
37. Wellman, M.: Online marketplaces. In: Singh, M.P. (ed.) Practical Handbook of Internet Computing. CRC Press, Boca Raton (2004)

Quantile Matrix Factorization for Collaborative Filtering

Alexandros Karatzoglou[1] and Markus Weimer[2]

[1] Telefonica Research
Barcelona, Spain
alexk@tid.es
[2] Yahoo! Labs
Santa Clara, USA
weimer@acm.org

Abstract. Matrix Factorization-based algorithms are among the state-of-the-art in Collaborative Filtering methods. In many of these models, a least squares loss functional is implicitly or explicitly minimized and thus the resulting estimates correspond to the conditional mean of the potential rating a user might give to an item. However they do not provide any information on the uncertainty and the confidence of the Recommendation. We introduce a novel Matrix Factorization algorithm that estimates the conditional quantiles of the ratings. Experimental results demonstrate that the introduced model performs well and can potentially be a very useful tool in Recommender Engines by providing a direct measure of the quality of the prediction.

1 Introduction

Recent research on Recommender System algorithm is focused on Collaborative Filtering methods that compute the predictions for a user-item pair based on past ratings of this and other users. Factor models and more precisely Matrix Factorization approaches have been introduced with great success in this area, see e.g. [1] for an example using data from the Netflix Prize Competition.

The vast majority of the state-of-the-art Matrix Factorization techniques are based on least squares regression-like methods in order to build a model capable of predicting the rating for a given user-item pair. While many systems provide impressive performance as expressed in empirical evaluation measures such as the root mean squared error, they cannot address additional questions about the recommendation posed by both website owners and users; such as the how certain the system is about the quality of a prediction, e.g. in the form of a confidence interval.

From the perspective of the website owner, such information would be valuable in order to provide the users with the right “mix” of recommendations containing items the user will like with a high confidence as well as more adventurous recommendations. In more technical terms, recommender engines could be set to a conservative mode giving priority in the list of recommendations to items that exhibit a narrower confidence interval (i.e. we are fairly sure about our prediction) or could be set to a more adventurous mode where recommendations are given purely based on the conditional mean, median or the 1st quantile etc.

F. Buccafurri and G. Semeraro (Eds.): EC-Web 2010, LNBIP 61, pp. 253–264, 2010.

Users, on the other hand, may well be interested in using this information in order to narrow down their item search by first eliminating all the items they surely won't like. Please note that "surely" and "won't like" in the last sentence are two distinct pieces of information to be gathered from the recommender system. In fact, these two dimensions of a recommendation can be presented to the user in an intuitively understandable two dimensional field of recommendations similar to the one presented in [2].

Standard regression methods typically aim at estimating $y|x$ by finding the conditional mean. Quantile regression methods [3], [3], [4] aim at estimating the τ quantile of the conditional distribution of $y|x$ where e.g. the 5^{th} quantile corresponds to the median. Note that while squared error loss based regression techniques are sensitive to noise and outliers, the estimation of the conditional median is more robust to both outliers and noise. User rating data and collaborative data in general are known to contain noise and outliers [5].

In least squares regression the desired estimate of $y|x$ is given by a conditional mean. In certain occasions one wants to obtain a good estimate that satisfies the property that a proportion, τ, of $y|x$, will be below the estimate (for $\tau = 0.5$ this is an estimate of the median). This type of regression is known under the term quantile regression. Assume that the conditional quantile is given by the function $f(\tau|x)$ then for example for $\tau = 0.9$, $f(0.9|x)$ is the 90th percentile of the distribution of y conditional on the values of x i.e. 90% of the values of y are less than or equal to the value $f(0.9|x)$. Quantile regression has been used in many areas such as in monitoring the growth of infants given their age and gender, in ecology, in quality control and risk management where a banker might want to estimate with a high certainty a lower bound of the value of a set of financial products.

Contributions. In this paper, we introduce a novel algorithm based on Matrix Factorization for computing conditional quantile estimates on Collaborative Filtering data. This algorithm provides for both a robust estimate of the conditional median of a user item combination and for any other quantile that can consequently be used to present the confidence of the provided recommendation. We present the following contributions to the field of factor models for collaborative filtering:

- A novel model for collaborative filtering based on quantile regression.
- An empirical analysis of the behavior of this system on real data.
- A simple way to integrate "external data" on the users such as demographic information and the movies such as genre into the matrix factorization model.

To the best of our knowledge, we are presenting the first matrix factorization algorithm for quantile estimation.

Organization of this Paper. The remainder of this paper is organized as follows: Section 2 introduces regularized matrix factorization as well as the related work in Section 1.1. Section 3 presents the main contribution of this paper, a quantile regression model built upon the matrix factorization framework. This model is studied empirically in Section 4 before the paper ends with conclusions in Section 5.

1.1 Related Work

Factor models and more specifically matrix factorization methods have been successfully introduced to Collaborative Filtering and form the core of many successful recommender system algorithms. The basic idea is to estimate vectors $U_i \in \mathbb{R}^d$ for each user i and $M_j \in \mathbb{R}^d$ for every item j of the data set so that their inner product minimizes an explicit [6] or implicit loss function [7].

Many of the most popular matrix factorization algorithms including SVD are based upon minimizing the least squares loss function (see e.g. [8,9]) where the sum of the squared errors between $F_{i,j}$ and $Y_{i,j}$ is used as loss function. A notable exception is the Maximum Margin Matrix Factorization approach presented in [6] which uses a multi-class hinge loss in conjunction with an L_2 regularizer to compute a model that introduces a large margin of separation, and thus improved generalization, performance.

In *matrix factorization* the observations are viewed as a sparse matrix Y where Y_{ij} indicates the rating user i gave to item j. Matrix factorization approaches then fit this matrix Y with a dense approximation F. This approximation is modeled as a matrix product between a matrix $U \in R^{n\times d}$ of user factors and a matrix $M \in R^{m\times d}$ of item factors such that $F = UM^T$.

Directly minimizing the error of F with respect to Y is prone to overfitting and thus regularization is required. Limiting the rank of the approximation by restricting d leads to a SVD of F, which is known as Latent Semantic Indexing in Information Retrieval. Note that this approach ignores the sparsity of the input data and instead models Y as a dense matrix with missing entries being assumed to be 0, thereby introducing a bias against unobserved ratings.

An alternative is proposed in [10] by penalizing the estimate only on observed values. While finding the factors directly now becomes a nonconvex problem, it is possible to use semidefinite programming to solve the arising optimization problem for hundreds, at most, thousands of terms, thereby dramatically limiting the applicability of their method. An alternative is to introduce a matrix norm, which can be decomposed into the sum of Frobenius norms [11,6,12]. It can be shown that the latter is a proper matrix norm on F. Together with a multiclass version of the hinge loss function that induces a margin, [6] introduced Maximum Margin Matrix Factorization (MMMF) for Collaborative Filtering. We follow their approach in this paper. Similar ideas were also suggested by [8,13,1] mainly in the context of the Netflix Prize.

2 Regularized Matrix Factorization

2.1 Model

In Matrix Factorization methods for Collaborative Filtering, the known data is interpreted as a sparse matrix $Y \in \mathbb{R}^{n\times m}$ where $Y_{i,j}$ contains the rating of item j by user i, if such a rating is known. The predicted rating $F_{i,j}$ of item j by user i is modeled as a linear combination of *item factors* $M_{j*} \in \mathbb{R}^d$ and *user factors* $U_{i*} \in \mathbb{R}^d$:

$$F_{ij} = \langle U_{i*}, M_{*j} \rangle \tag{1}$$

where U_{i*} is the factor vector for user i and M_{*j} the factor vector for item j.

Let $U \in \mathbb{R}^{n\times d}$ denote the matrix of all user factor vectors and $M \in \mathbb{R}^{m\times d}$ the matrix of all item factor vectors. We can then express this prediction rule as a matrix product:

$$F = UM' \tag{2}$$

When learning a matrix factorization model, the aim is to estimate U and M in such a way that the model predictions F minimize a loss L on the training set Y.

$$U, M := argmin_{U,M} L(F = UM', Y) \tag{3}$$

However, optimizing this will typically yield poor predictive performance due to overfitting. Thus, a *regularization function* $\Omega(F)$ is added for capacity control and thus overfitting prevention. This leaves us with the following objective function:

$$U, M := argmin_{U,M} L(F = UM', Y) + \lambda\Omega(F) \tag{4}$$

Here, λ is a constant that is used to control the trade-off between the regularization and the performance of the system on the known training data. Typically (e.g. in [6]), the regularizer $\Omega(F)$ is chosen to be the sum of the Frobenius (L_2) norms of the matrices U and M.

3 Quantile Matrix Factorization

In analogy to [6] we define the loss:

$$L(F, Y) := \frac{1}{\|S\|_1} \sum_{i,j} S_{ij} l(F_{ij}, Y_{ij}) \tag{5}$$

where $l : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}$ is a pointwise loss function penalizing the distance between estimate and observation while by $S \in \{0; 1\}^{n\times m}$ we denote a binary matrix with nonzero entries S_{ij} indicating whenever Y_{ij} is observed.

We now seek to find matrices $M \in \mathbb{R}^{m\times d}$ and $U \in \mathbb{R}^{n\times d}$ minimizing the following objective function:

$$U, M := argmin_{U,M} \frac{1}{\|S\|_1} \sum_{i,j} S_{i,j} l(F_{ij}, Y_{ij}) + \lambda\Omega(F) \tag{6}$$

Given the factors U, M which constitute our model we have a choice of ways to ensure that the model complexity does not grow without bound. A simple option is [12] to use the penalty

$$\Omega[U, M] := \frac{1}{2}\left[\|U\|^2_{\text{Frob}} + \|M\|^2_{\text{Frob}}\right]. \tag{7}$$

Indeed, the latter is a good approximation of the penalty we will be using. The main difference being that we will scale the degree of regularization with the amount of data similar to [1]:

$$\Omega[U, M] := \frac{1}{2}\left[\sum_i n_i \|U_i\|^2 + \sum_j m_j \|M_j\|^2\right] \tag{8}$$

Here U_i and M_j denote the respective parameter vectors associated with user i and item j. Moreover, n_i and m_j are scaling factors which depend on the number of reviews by user i and for item j respectively.

Note here that the loss function is only computed on the non-zero elements of the matrix Y as zeros are treated as missing values and not ratings. In order to now build a model to estimate quantiles on collaborative data and ultimately provide more transparent and more flexible recommendations, we introduce and adapt a new loss function to matrix factorization, namely the so-called quantile loss.

3.1 Quantile Regression

Definition 1 (Quantile). *Let $y \in \mathbb{R}$ be a random variable and $\tau \in (0,1)$. The τ quantile of y μ_τ is given by the infimum over μ for which $Pr\{y \leq \mu\} = \tau$. The conditional Quantile $\mu_\tau(x)$ for a pair of random variables $(x,y) \in \mathbb{X} \times \mathbb{R}$ is defined as the function $\mu_\tau : \mathbb{X} \to \mathbb{R}$ for which μ_τ is the infimum over μ so that $Pr\{y \leq \mu | x\} = \tau$.*

The loss function for quantile regression was derived in [3] based on the observation that minimizing $l(f) = |f - y|$ would force half of the estimates of f to lie over y and half below. This error measure corresponds to the absolute error loss functions which yields a conditional median as a solution at optimality. Essentially due to the symmetry of this error measure there will be about as many data points with negative residuals as with positive.

This leads to the natural observation that by minimizing an asymmetrically weighted absolute error measure and thus giving different weights to points below and above the estimate one can retrieve the quantiles. Effectively by tilting the loss function (figure 1) in a suitable fashion, one can get estimates for any quantile. The loss function used for quantile regression is also known as the pinball loss and is given by:

$$L(F_{ij}, Y_{ij}, \tau) = \begin{cases} \tau(F_{ij} - Y_{ij}) & F_{ij} \geq Y_{ij} \\ (\tau - 1)(F_{ij} - Y_{ij}) & F_{ij} < Y_{ij} \end{cases} \tag{9}$$

where $\tau \in (0,1)$ is the quantile to be obtained. The optimization procedure given below requires the derivative of the loss with respect to F, which can be computed as:

$$\partial_F L(F_{ij}, Y_{ij}) = \begin{cases} \tau & F_{ij} \geq Y_{ij} \\ \tau - 1 & F_{ij} < Y_{ij} \end{cases} \tag{10}$$

The loss and the derivative can be calculated for each individual user-item-rating triple and thus many simple optimization methods can be used. An illustration of the loss function is given in 1.

Note that when optimizing the pinball loss for Matrix Factorization the prediction function of the model $F_{ij} = \langle U_{i*}, M_{j*} \rangle$ does not return a prediction of the rating that user i might give to item j but instead returns an estimate of the conditional quantile of that prediction for the value of τ used during the optimization process. We can thus use two Quantile Matrix factorization models one for e.g. $\tau = 0.25$ and one for $\tau = 0.75$ and define a confidence interval for the ratings (in this case the interquartile range) for each user-item rating combination.

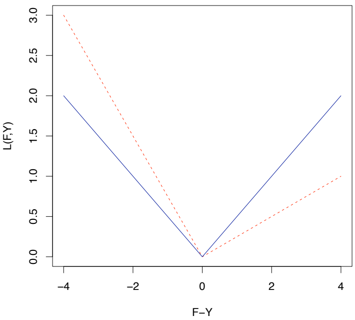


Fig. 1. Plot of the pinball loss for $\tau = 0.5$ blue line and for $\tau = 0.25$ red dotted line. Notice the higher loss incurred for negative $F - Y$ when $\tau = 0.25$.

3.2 Optimization

The loss function introduced above decomposes per element of F and Y. Thus, we can employ an online optimization technique similar to the one mentioned in [8]. This method is an adaptation of the simple Stochastic Gradient Descent algorithm to the matrix factorization framework. To this end we compute the element-wise gradient of the objective function (6) with respect to F.

Notation. As the loss decomposes per element, we can define the following, more compact notation: Let f and y denote two corresponding element entries in F and Y and u and m be the corresponding rows of U and M. Then, we can formulate:

$$\partial_f E(f, y) = \partial_f L(f, y) + \frac{\lambda}{2}\partial_f \left(\|u\|_2^2 + \|m\|_2^2\right) \tag{11}$$

The partial gradients with respect to u and m can then be written as:

$$\partial_u E = \partial_f L(f, y)m + \lambda u \tag{12}$$

$$\partial_m E = \partial_f L(f, y)u + \lambda m \tag{13}$$

This, together with a learning rate η, allows us to define the following *update rules* for an iterative optimization procedure:

$$u^{t+1} = u^t - \eta \partial_u E \tag{14}$$

$$m^{t+1} = m^t - \eta \partial_m E \tag{15}$$

This algorithm scales linearly to the number of nonzero entries in the matrix Y, and the dimensionality of d of the feature matrices that is $O(\|S\|_1 d)$ it can thus be used on very large datasets.

3.3 Feature Extensions

In Recommender Systems often additional information on the user or the items is available either in the form of demographic information on the users or of genre information for movies. In [14], the integration of features is proposed by defining a kernel between rows and columns that integrates features. Another way of introducing features is to use them as a prior for the factors, as studied in [15].

Here, we present an integration of features to the matrix factorization framework by adding several linear learning models. Assuming that the d_u user features are contained on the rows of $X^U \in \mathbb{R}^{n \times d_u}$ where n the number of users the prediction model then becomes:

$$F_{ij} = \langle U_{i*}, M_{j*} \rangle + \frac{1}{\sqrt{n_i}} \left\langle X^U_{i*}, W^U_{*j} \right\rangle \tag{16}$$

where $W^U \in \mathbb{R}^{d_u \times m}$ is a parameter matrix where m the number of movies. The same idea can be applied for the item features:

$$F_{ij} = \langle U_{i*}, M_{j*} \rangle + \frac{1}{\sqrt{n_i}} \left\langle X^U_{i*}, W^U_{*j} \right\rangle + \frac{1}{\sqrt{m_j}} \left\langle X^M_{j*}, W^M_{i*} \right\rangle \tag{17}$$

with $X^M \in \mathbb{R}^{m \times d_m}$ and $W^M \in \mathbb{R}^{n \times d_m}$ and W^U_{*j} denoting the jth column of W^U and n_i the number of items rated by user i and m_j the number of ratings for item j. We discount the influence of the external features in proportion to the square root of the number of items a user has rated. The model thus gives more weight to external features for users that have few ratings. The same principal is also applied to the items. The features can be integrated easily in the optimization procedure, i.e. in each iteration an additional update of the corresponding columns in W^U and rows in W^M is performed. The newly introduced parameter matrices are also regularized using the Frobenius L_2.

4 Experiments

Quantile Factorization models computes only quantiles and thus cannot be compared directly to standard Matrix Factorization approaches that minimize the RMSE. Moreover it is not the aim of the model to provide an exact rating prediction but instaed to provide a measure of quality for a prediction.

Section 4.1 describes the evaluation procedure, including the choice of data sets, evaluation measure and parameter tuning conducted. Section 4.2 contains the results obtained for the new model with this procedure.

4.1 Evaluation Setup

All experiments where conducted on the same data sets using the train-test splits, the evaluations measure and the number of factors d described in the following paragraphs.

Data. For the experiments, we used the well known data sets EachMovie and MovieLens. Table 2 shows the descriptive statistics for these data sets. Please note that we did not perform any preprocessing of the data such as mean removal and normalization. Detailed information on proper preprocessing and main effect removal for collaborative filtering data can be found in [16] and [17].

Both data sets originate from movie recommender systems and thus contain ratings of movies by users. The ratings in the MovieLens are given on a five star scale, while

Table 1. Data set statistics

Data set	Users	Movies	Ratings
EachMovie	61265	1623	2811717
MovieLens	6040	3900	1000209

EachMovie uses six rating levels. In both cases, more stars indicate a higher rating. For the EachMovie data set, we randomly extracted 10 ratings from the known ratings of each user to form the *test set* for this user. The remaining known ratings were used as the training set. For the Movielens data set, we used the 5 train-test splits provided by the GroupLens[1] research lab.

Fixed Number of Factors. We tuned the dimensionality factor and regularization parameter for good performance. In all experiments the number of factors d was fixed to 20 for EachMovie and 15 for MovieLens. In our experiments we also use a single regularization parameter λ.

Model Evaluation. Computing the true conditional quantile is impossible and most methods only approximate it using density estimation. Moreover, we are not aware of any other method for quantile estimation on collaborative filtering data. We thus conduct an empirical evaluation of the performance of the model by computing the portion of user-item ratings that fall over the estimated conditional quantile value F_{ij} which should be close to the quantile τ for which we are optimizing. We also explore some properties of the quantiles and the interquantile ranges.

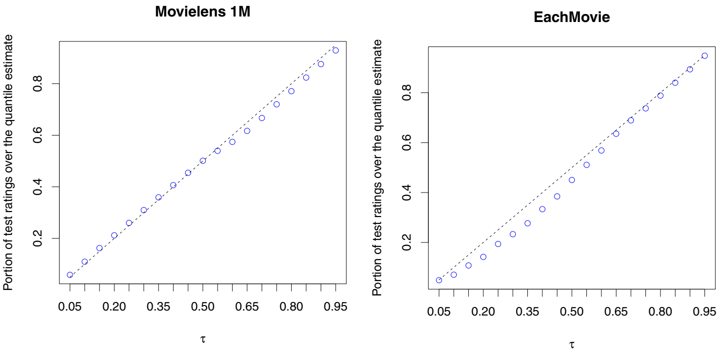
4.2 Results

Model Validation. Our aim in this set of experiments is to validate the model by computing the portion of the test user ratings that fall over the computed conditional quantiles. To this end, we set the value of τ and train the model for each training set in the data. During evaluation we count the number of user-movie ratings that fall above the estimated quantile in the test set and compute the proportion these represent in the test set as mentioned this should be optimally in the order of τ. We then report the mean of these proportions over all the different train - test splits. Finally, we repeat the procedure for different τ values.

We performed model selection on one test train split for $\tau = 0.5$ for each dataset and used the computed values of λ for the procedure. Note that the standard deviation of these estimates over the different train-test sets is of the order of 0.005 and thus not reported.

Figure 4.2 contains the results of this experiment on the EachMovie and the MovieLens data. We observe very good performance of the model: the quantiles the system was trained for almost exactly carry over to the test set. We can thus deduce that the estimated conditional quantile estimates are very close to the true values and that the system exhibits a strong generalization performance.

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



(a) (b)

Fig. 2. Scatterplots of the proportion of user ratings that fall over the quantile estimates in the test set for different values of the pinball loss quantile parameter τ both for the Eachmovie dataset 2(b) and the Movielens dataset 2(a). The dotted line is the $y = x$ line.

Table 2. Results obtained form a model without and with external Features on the Movielens dataset. For $\tau = 0.10$ the model with features is statistically significantly better. For the other τ there is no statistically significant difference though the model with feature tends to perform slightly better.

	Plain	Features
$\tau = 0.10$	0.1122	**0.1096**
$\tau = 0.25$	0.2609	0.2601
$\tau = 0.50$	0.5011	0.5011
$\tau = 0.75$	0.7192	0.7201
$\tau = 0.90$	0.8759	0.8761

Features. We also compare the performance of the system with external features for the users and the movies provided in the Movielens dataset to the plain factorization model. The Movielens dataset contains demographic information on the users and genre information on the movies. Using 17 we include this information into the model. We repeat the experimental procedure which we used to generate figure 2 with and without features. Table 2 contains the results for different values of τ.

The result show that including the features does only marginally improve the performance of the model. In fact this observation seems to confirm the notion that in the presence of enough rating data external features do not provide significant additional performance benefits to collaborative models [18].

Quantile Properties. In this set of experiments we demonstrate the behavior of the quantiles given the amount of movies a users has seen. We calculate the 2.5 and the 7.5 conditional quantiles by training models for $\tau = 0.25$ and $\tau = 0.75$ on the MovieLens training sets. We then compute the conditional inter-quantile range that is the difference between the 2.5 and the 7.5 quantile on the test set. We split the users in five groups of users each containing about 190 users so that the first group contains users who have rated $1 - 21$ movies the second $22 - 39$ etc.

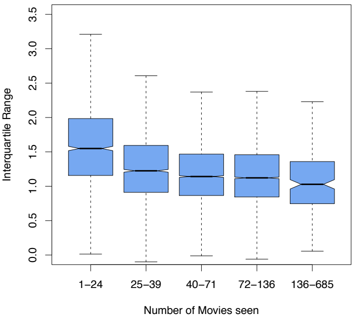


Fig. 3. Boxplot of the Interquartile Range given by groups of users grouped by numbers of movies seen on the MovieLens dataset. The upper and lower limits of the boxes represent the first and third quartiles of the interquartile ranges. The median for each group is represented by the horizontal bar in the middle of each box. If the notches of two plots do not overlap this is strong evidence that the two medians differ.

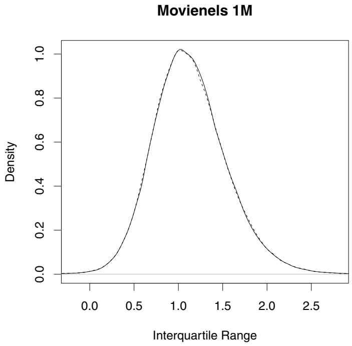


Fig. 4. Density plot of the distribution of the Interquartile range for the MovieLens dataset. The dashed line is the distribution of the interquartile range for a run without user and movie features We observe almost no significant difference in the two distributions Wilcox test p-value 0.194.

In Figure 3, we plot a notched boxplot of the interquartile ranges for different user groups grouped by the number of seen movies. We observe that the interquartile range is generally smaller and thus narrower for users who rated more movies. This behavior is to be expected since given more data, the model is able to identify “better” estimates, yielding somewhat narrower interquartile ranges. Moreover one would expect that users who have rated a large number of movies have developed a more consistent rating behavior.

Figure 4 shows the distribution of the width of the interquartile ranges. The interquartile range could provide for a good measure for the confidence of the rating prediction since by definition the “true” rating will be within this range with a 50% probability. We also observe that a large portion of the calculated ranges are below 1, indicating that the system is rather certain about these predictions. On the same figure we also plot the distribution of the interquartile ranges for a model with features, notice that there is no significant difference between the two.

5 Conclusions

We introduced a novel Quantile Matrix Factorization model. The model is able to estimate the confidence of the system when computing rating predictions in a Collaborative Filtering setting. To the best of our knowledge, this is the first system able to perform quantile and thus confidence prediction in Recommender Systems. Recommender Systems stand to benefit from this additional information by providing users with more information on the quality of the recommendation and giving practitioners another option in configuring their recommender engines. Experimental evaluation of the system demonstrated that the model provides an excellent estimate of the true conditional quantile and exhibits properties that warrants investigating its utility to the end-users in future research.

References

1. Bell, R., Koren, Y., Volinsky, C.: The bellkor solution to the neflix prize. Technical report, AT&T Labs (2007)
2. Ries, S.: Extending bayesian trust models regarding context-dependence and user friendly representation. In: Proc. of the 2009 ACM Symposium on Applied Computing. ACM, New York (2009)
3. Koenker, R., Hallock, K.: Quantile regression. Journal of Economic Perspectives 15(4), 143–156 (2001)
4. Takeuchi, I., Le, Q.V., Sears, T.D., Smola, A.J.: Nonparametric quantile regression. Journal of Machine Learning Research 7, 1231–1264 (2006)
5. Amatriain, X., Pujol, J.M., Oliver, N.: I like it... i like it not: Evaluating user rating noise in recommender systems. In: Houben, G.-J., McCalla, G., Pianesi, F., Zancanaro, M. (eds.) UMAP 2009. LNCS, vol. 5535, pp. 247–258. Springer, Heidelberg (2009)
6. Srebro, N., Rennie, J., Jaakkola, T.: Maximum-margin matrix factorization. In: Saul, L.K., Weiss, Y., Bottou, L. (eds.) Advances in Neural Information Processing Systems 17 NIPS. MIT Press, Cambridge (2005)
7. Hoffman, T.: Latent semantic models for collaborative filtering. ACM Transactions on Information Systems (TOIS) 22(1), 89–115 (2004)
8. Takacs, G., Pilaszy, I., Nemeth, B., Tikk, D.: Scalable collaborative filtering approaches for large recommender systems. Journal of Machine Learning Research 10, 623–656 (2009)
9. Koren, Y.: Collaborative filtering with temporal dynamics. In: Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD). ACM Press, New York (2009)
10. Srebro, N., Jaakkola, T.: Weighted low-rank approximations. In: Proceedings of the 20th International Conference on Machine Learning ICML, pp. 720–727. AAAI Press, Menlo Park (2003)
11. Rennie, J., Srebro, N.: Fast maximum margin matrix factorization for collaborative prediction. In: Proc. of the 22nd International Conference on Machine Learning ICML (2005)
12. Srebro, N., Shraibman, A.: Rank, trace-norm and max-norm. In: Auer, P., Meir, R. (eds.) COLT 2005. LNCS (LNAI), vol. 3559, pp. 545–560. Springer, Heidelberg (2005)
13. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: Advances in Neural Information Processing Systems 20 NIPS. MIT Press, Cambridge (2008)
14. Abernethy, J., Bach, F., Evgeniou, T., Vert, J.P.: A new approach to collaborative filtering: Operator estimation with spectral regularization. Journal of Machine Learning Research 10, 803–826 (2009)

15. Agarwall, D., Chen, B.C.: Regression-based latend factor models. In: Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD). ACM Press, New York (2009)
16. Bell, R., Koren, Y.: Improved neighborhood based collaborative filtering. In: The Netflix-KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition. ACM Press, New York (2007)
17. Potter, G.: Putting the collaborator back into collaborative filtering. In: The 2nd-Netflix-KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition. ACM Press, New York (2008)
18. Pilaszy, I., Tikk, D.: Recommending new movies: Even a few ratings are more valuable then metadata. In: Proceedings of the 3rd ACM International Conference on Recommender Systems (RecSys). ACM Press, New York (2009)

Partial Ranking of Products for Recommendation Systems

Sébastien Hémon, Thomas Largillier, and Sylvain Peyronnet

LRI; INRIA; Univ. Paris-Sud XI; 91405 Orsay, France

Abstract. A recommendation system (or recommender) is an algorithm whose goal is to recommend products to potential users. To achieve its task, it uses information about some user preferences.

We present recommenders that use information about the preferences of only a very small subset of users (called a committee) on a very small set of products called the witness products set. The main interest of our approach compared to previous ones is that it needs substantially less data for ensuring a very good quality of recommendation.

1 Introduction and Related Work

Recommendation systems aim to select products for a particular user from a list shared by all (available products for instance) according to known previous preferences of users. There are essentially two ways to recommend products. One consists in learning the preferences of a particular user based on the products (s)he liked before and recommending products similar to these ones (content based approach, see [1]). The other approach consists in recommending products to someone by choosing products that have been liked by users that seem to have the same preferences as the person to recommend (collaborative filtering, see [2]).

We propose a recommender that follows the collaborative filtering paradigm. Our algorithm recommends to a particular user a product that will be in the set of its favorite products with high probability. Such recommendations are done according to a categorization of users into equivalence classes with respect to the relation “having the same preferred products”. Our algorithm is original since it produces a good recommendation with high probability without knowing the exact rating of each product by each user.

Most of the collaborative filtering work relies on an analysis of users' preferences that take their values in a continuous space (see [3,2,4]). In these papers, recommenders are based on the assumption that users can be categorized in classes that strongly differ one from each other. In [5], Kleinberg *et al* use mixture models to make a good recommender. In [6], the authors made a first attempt to design recommenders that are not based on specific assumptions about the internal behavior of a committee (a small set of users that rate a lot of products). Meanwhile, preferences need to be binary values. They are interpreted as "good" or "bad". This assumptions is clearly a drawback since the behavior of customers do not generally obey to such separate agreements. Moreover, the

F. Buccafurri and G. Semeraro (Eds.): EC-Web 2010, LNBIP 61, pp. 265–277, 2010.

need for the committee to evaluate all products is required. Several papers agree to consider such a need to be hardly reliable [3,4,2,7,6].

To the contrary, the essence of our method is to make user evaluate very few products with small discrete rating scale and consider that a committee would never be able to evaluate a large number of products. So our method does not use a bound on the number of products and ask for a committee to rank only a very small number of these products. It means that we do not need a rating of products, a strong assumption used in [4,8,9]). While [4] asks for a strong gap between user classes (to be said, orthogonality), we weaken this assumption by asking for user classes to be *sufficiently* different.

The structure of the paper is the following. We present in section 2 our framework. Section 3 sets the condition on products and users for the recommendation system to work. Section 4 presents our recommenders. Last, section 5 shows the effectiveness of our approach through a user satisfaction experiment.

2 Framework and Principle of Our Method

2.1 Our Framework : Modeling of Users and Products

The goal of a recommendation system is to provide users with "good" products. In this paper, we consider that users belong to a set $\mathcal{U} = \{u_1 \cdots u_m\}$ of m distinct users and that products come from $\mathcal{P} = \{p_1 \cdots p_n\}$ a set of n distinct products. We also suppose that we are given, even implicitly, a function $f : \mathcal{U} \times \mathcal{P} \longrightarrow \mathbb{R}$ that gives for every couple of user/product a *utility*; f is then a so-called utility function. We can now define a recommendation system as:

Definition 1. *A recommendation system is a function $\mathcal{R} : \mathcal{U} \longrightarrow \mathcal{P}_r$, where $\mathcal{P}_r = \{X \subseteq 2^{\mathcal{P}}, |X| = r\}$. Thus, for each user u_i, $\mathcal{R}(u_i)$ is a set of r products.*

r is a fixed parameter ($r = 5$ in our experiments). Let $\mathcal{F}_r(u_i)$denotes the r favorite products (according to f) of user u_i, we have the following definition.

Definition 2. *A good recommendation occurs when we have $\mathcal{F}_r(u_i) \cap \mathcal{R}(u_i) \neq \emptyset$*

Our goal is to obtain an algorithm that gave good recommendations.

For convenience, we summarize the utilities in a $m \times n$ matrix M_f such that $M_f(i,j) = f(u_i, p_j)$. We denote respectively by $M_f(i,\star)$ and $M_f(\star,i)$ the i^{th} row and j^{th} column of the matrix M_f.

In order to design recommenders, we are interested in top values of a given row $M_f(i,\star)$. These values corresponds to the favorite products of user u_i. These top values are given by an injective function $rank : \mathcal{U} \times \mathcal{P} \longrightarrow [n]$, where $[n]$ denotes the set $\{1, \cdots, n\}$ for $n \in \mathbb{N}$. $rank(u_i, p_j)$ is the index of p_j in the sorted (according to the values of f) list of products for user u_i. For instance, $rank(u_1, p_2) = 3$ means that p_2 is the third preferred product of user u_1.

Using the function $rank$ we can define the notion of $r - equivalence$ for users.

Definition 3. *Two users u_i and u_j are said to be r-equivalent if and only if*

$$\forall p \in \mathcal{P} \quad rank(u_i, p) \leq r \Longleftrightarrow rank(u_j, p) \leq r$$

Intuitively, u_i and u_j are r-equivalent if they have the same r preferred products (but not necessarily with the same order of preference).

We also define a function $index : [n] \times \mathcal{U} \longrightarrow [n]$. $index(*, u_i)$ that indicates the permutation that sort products according to their rank for u_i. We then define an equivalence relation $\equiv_{ps}$ between users. It is called the *product sorting relation.* $\widetilde{\mathcal{U}}$ denotes the quotient space of $\mathcal{U}$ by $\equiv_{ps}$.

Definition 4. *Two users u_k and u_l are equivalent w.r.t. the relation $\equiv_{ps}$ iff*

$$\forall i \leq n \quad index(i, u_k) = index(i, u_l).$$

In this case we write $u_k \equiv_{ps} u_l$.

If necessary, we use a $m \times n$ matrix S, called the sort table, such that $S(i, j) = index(j, u_i)$. This will be only for the sake of clarity in the notations.

We are interested in good recommendations, so we need a notion of equivalence between users that considers only the favorite products of a given user.

Definition 5. *Let $r < n$, two users u_k and u_l are r-equivalent (that is have the same r favorite products) if and only if :*

$$\forall i \leq r \quad \exists j \leq r \quad index(i, u_l) = index(j, u_k)$$

When this happens, we write $u_k \equiv_r u_l$. Moreover this is an equivalence relation.

This relation is important for the rest of the paper since our goal is to deal with only a small numbers of products (here r) in order to give good recommendation. $\widehat{\mathcal{U}}$ denotes the quotient space of $\mathcal{U}$ by $\equiv_r$.

2.2 Principle of Our Method

We follow the modeling we just defined above. We want our recommendation system to output a good recommendation of r products, where r is a very small integer (typically $r = 5$ in our experiments). It is often admitted that, to get a good recommendation, users follow some kind of behavior which can be viewed as arbitrary distinct classes. In our method we made this natural and formal using the notion of product sorting equivalence and r-equivalence. Thus, we do not admit that users behave the same, exactly or modulo some randomized perturbations, but only tell that there is a model that, given a ranking of products for each user, naturally sort users into equivalence classes. In the following, we will consider cases where both quotient sets $\widehat{\mathcal{U}}$ and $\widetilde{\mathcal{U}}$ have small cardinality in regards to both m and n (numbers of users and products).

All of our work was done according to two assumptions. The first is that we have access to a few users who will rank a set of witness products and give their r favorite products. These users are known as “the committee”. The second assumption is that we are authorized to ask every people about these witness products. This two hypothesis allow us to say that we use partial information in order to make its recommendation. The main issue is then to understand what should be the size of the committee, and how many witness products we need in order to be able to provide a good recommendation to users.

Our recommender is described in Fig.1. We first choose a committee and ask to each member of the committee to sort some products according to its preferences. Then we choose a set of witness products and ask all users to sort those products. Then we can cluster, with high confidence, users into equivalence classes according to their products sorting. The given clustering will likely attribute at least one member of the committee to each equivalence class. This peculiar user will be used to make recommendations to members of its class.

In the following we note by C the committee (thus $C \subseteq \mathcal{U}$) and by W the witness products ($W \subseteq \mathcal{P}$). As said previously, $\widehat{\mathcal{U}}$ and $\widetilde{\mathcal{U}}$ are the quotient sets of $\mathcal{U}$ with respect to, respectively, $\equiv_r$ and $\equiv_{ps}$. Classes of $\widetilde{\mathcal{U}}$ (resp. $\widehat{\mathcal{U}}$) are denoted by $\widetilde{u_i}$ (resp. $\widehat{u_i}$) for i ranging from 1 to $|\widetilde{\mathcal{U}}|$ (resp. $|\widehat{\mathcal{U}}|$). For the sake of clarity, we use θ as a notation for the cardinal of $\widetilde{\mathcal{U}}$ and θ_i for the cardinal of class $\widetilde{u_i}$.

The next section is devoted to the calculus of $|C|$ and $|W|$ in order to make a good recommendation. Our goal is to find W and C such that the following holds: let $u_i \in C$ and $u_j \in \mathcal{U}$. If $u_i \equiv_{ps} u_j$ for products from W then $u_i \equiv_r u_j$ on $\mathcal{U}$ with high probability. Note that it implies that with high probability we can have a good recommendation for u_j by giving him the favorite of u_i.

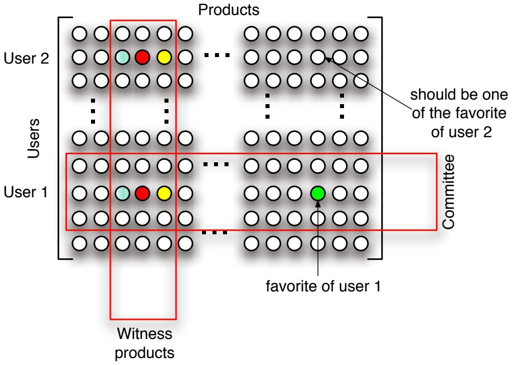


Fig. 1. Principle of our method

3 Recommendation under Partial Information

We now address the problem of the size of both the committee and the set of witness products in order to provide good recommendations with high probability. We first consider the problem of the size of the witness products set.

3.1 Cardinality of the Witness Products Set

We consider that the relation between products is preserved, e.g. if a user prefers product p_i to product p_j, he will provide a higher utility value for p_i than for p_j. This only means that users always sort products the same way. This may not be true that users evaluate utilities the same way, regardless of the proposed products. Sorting information may look weaker than utility in order to perform recommendations, but it is clearly more robust : context or even mood can change the actual value of utility. A users does not have the will to be rigorous but it will certainly keep its preferences. Thus, this is one of the assumption on which we build our algorithm.

We now look at bounds for $|W|$, assuming that θ is the number of classes of users, regarding the relation *having same product sorting* ($\equiv_{ps}$). We start by giving a combinatorial result (proof in appendix) which will ensure bounds for our recommendation system to work:

Proposition 1 (sub-permutation). *Let τ be a permutation over $[n]$ and W be a subset of $[n]$ of cardinality $|W|$. There exists a unique sub-permutation $\tau|_W$ which is the one-to-one mapping from W to $[|W|]$ that satisfies :*

$$\forall k, l \in W \quad \tau|_W(k) < \tau|_W(l) \Leftrightarrow \tau(k) < \tau(l) \tag{1}$$

Lemma 1. *Let $\{\tau_i ; i \leq \theta\}$ be a family of θ distinct permutations over $[n]$. To distinguish all τ_i between them with their sub-permutations, using a single $W \subset [n]$ of cardinality $|W|$, we need at most $|W| = \inf\{n; 2(\theta - 1)\}$ elements.*

Proof. The proof is in the appendix.

Observe that the proof of the lemma defines an algorithm which computes the set W. Its time complexity is $\theta \cdot n$, which is just the input size. This algorithm is optimal among deterministic techniques. We will prove that we can get good recommendations with high probability when taking less than $2(\theta - 1)$ products.

We now rephrase the two last results in term of recommendation :

Proposition 2. *Let $W \subset \mathcal{P}$ be of cardinality $|W|$ and $\widetilde{\mathcal{U}} = \{\widetilde{u_1}, \cdots, \widetilde{u_\theta}\}$ be the quotient set of $\mathcal{U}$, w.r.t.* having the same product sorting *relationship, where $\theta \leq \lceil n/2 \rceil$. Suppose we are given $M_f(l_i, \star)$ for at least one user $u_{l_i} \in \mathcal{T}_i$ for all $i \leq \theta$ (or, which is enough, an ordering of all products by user u_{l_i}). Then we need at least $|W|$ products in the witness products set in order to make good recommendation to all users where $|W|$ is such that $(|W|)! \geq \theta$ and $|W| \leq 2(\theta - 1)$.*

Proof. We prove the proposition in two steps.

Lower Bound: It is sufficient to note that when extracting the ordering of $|W|$ products for a user u_j, the total number of possible different ordering is exactly $(|W|)!$. Thus, it becomes clear that if θ is the number of different classes of users among $\mathcal{U}$, we obviously need $|W|$ such that $(|W|)! \geq \theta$ in order to distinguish all different classes and, thus, making a *good recommendation* to u_j.

Upper Bound : From $M_f(l_i, \star)_{i \leq \theta}$, we can compute a family of functions $index(*, u_{l_i})$. For a given l_i, the function $index(*, u_{l_i})$ can be seen as a permutation over $[n]$.

We can now use proposition 1. We need a subset of $[n]$ of cardinality no more than $2(\theta - 1)$ to distinguish each of these θ permutations with their associated sub-permutations. This means that we need $|W| \leq 2(\theta - 1)$ products to make every functions $index(*, u_{l_i})$ one different of each others.

Finding exactly the $|W|$ products that fit with the lower bound may be a very difficult task. If we choose these products uniformly at random they will be unlikely to ensure to distinguish between classes. Then, it would certainly be easier to choose more products for the witness products set in order to be able to have enough information. The upper bound means that it is always possible to make good recommendations, under condition that every class of user is known via one of its representant, with at most $2(\theta-1)$ known products per user. Thus, we add to our recommender a specific algorithm to select products that every people should evaluate.

Although we have these lower and upper bounds, we feel concerned with finding the number of products needed in W in order to achieve a fast but good recommendation. Proposition 3 below gives the solution.

Proposition 3. *Let $W \subseteq \mathcal{P}$ a set of products and $\widetilde{\mathcal{U}}$ be as before. Suppose we know for all $i \leq \theta$, $M_f(l_i, \star)$ for at least one user u_{l_i} such that $u_{l_i} \in \widetilde{u}_i$. Then if we pick uniformly at random $|W| = \lceil\sqrt{\theta}\rceil$ elements from $\mathcal{P}$, we can correctly sort, with high probability, every user in its class (e.g. equivalence class w.r.t. the $\equiv_{ps}$ relation) by looking the ordering of those $|W|$ products for each user.*

Proof. Let Dst be the event *every $M_f(l_i,\star)|_W$ is distinct*, when taking $|W|$ elements from $\mathcal{P}$ with uniform distribution. This event can be observed as distinguishing each class from $\widetilde{\mathcal{U}}$ with $|W|$ products. Hence, this means exactly that, taking one user from each class as a witness of his class, we do not want two distinct of these users to be represented asame (i.e. with the same ordering over these $|W|$ products). The probability of the event Dst is given by :

$$\Pr[Dst] = \prod_{i=0}^{\theta-1} \frac{(|W|)! - i}{((|W|)!)} \quad \Leftrightarrow \quad \ln\left(\Pr[Dst]\right) = \sum_{i=0}^{\theta-1} \ln\left(1 - \frac{i}{(|W|)!}\right)$$

Which leads to, using power series :

$$\ln\left(\Pr[Dst]\right) = \sum_{i=0}^{\theta-1} \sum_{j\geq 1} \frac{(-1)^j}{j} \left(\frac{i}{(|W|)!}\right)^j = \frac{-1}{(|W|!)} \sum_{i<\theta} i \quad + o(\tfrac{1}{(|W|)!})$$

Finally, we get: $\ln\left(\Pr[Dst]\right) = \frac{\theta-\theta^2}{2(|W|)!} + o(\frac{1}{(|W|)!})$, which leads to the approximation: $\Pr[Dst] \approx e^{-\frac{(|W|)^4}{2(|W|)!}}$. This is close to 1, thus the proposition holds.

From now on, we know how much products we have to choose in our witness products set in order to achieve good recommendation with high probability. We now address the problem of the number of users that must be in the committee.

3.2 Cardinality of the Committee

In order to cluster users into classes that depend on the product sorting equivalence, we need a committee that will evaluate all products of the set W.

This committee is a subset C of $\mathcal{U}$ whose members must be chosen in a way or an other. After choosing how we sample users for being in the committee, we must evaluate how much of these users we need. As in the case of witness products, it is easier to have people chosen uniformly at random. We should then study the behavior of a recommendation system that pick users for its committee with uniform distribution. We thus define the notion of good (e.g. representative) committee.

Definition 6. *A committee C is representative of $\mathcal{U}$ if and only if :*

$$\forall \widetilde{u}_i \in \widetilde{\mathcal{U}}, \quad \exists u \in C \text{ such that } u \in \widetilde{u}_i$$

Our goal is then to pick enough users in order to have a representative committee, that is obtaining at least one member of each class of users (w.r.t. the relation *having same product sorting*) with high probability.

We then have to evaluate the probability of getting one member of each of such class when asking $|C|$ users from $\mathcal{U}$ at random with uniform distribution, where this last event will be written *GC* for *Good Committee.* It is given via the probability of the opposite event (i.e. not having a representative committee) :

$$\Pr[\neg GC] = \sum_{i=1}^{\theta} \left(1 - \frac{\theta_i}{m}\right)^{|C|}$$

We recall that θ_i stands for the cardinal of class $\widetilde{u}_i$ and $m = |\mathcal{U}|$.

We make some reasonable assumptions about θ_i and θ. It seems natural to assume that $\theta \ll m$, that is the number of classes, is small w.r.t. the number of users. We also suppose that every class of users contains a non negligible number of users. Formally, these facts can be expressed as $\theta = O(1)$ and $\theta_i = \Theta(m)$ with associated multiplicative constant $q_i < 1$ for all $i \leq \theta$. That is $\theta_i = q_i \cdot m$. In the following we consider, without loss of generality, that $q_1 < q_2 < \cdots < q_\theta$ so that the size of the $\widetilde{u}_i$ is increasing with i.

These assumptions are reasonable since, in practice, we are not interested in providing good recommendations to users that belong to not large enough classes. Providing users in small classes with good recommendations will increase massively the size of the committee only to satisfy few more users.

We can now compute the probability $\Pr[\neg GC]$, written ε from now on :

$$\varepsilon \approx \sum_{i=1}^{\theta} (1 - q_i)^{|C|} \leq \theta \, (1 - q_1)^{|C|} = \theta e^{|C| \ln(1 - q_1)}$$

This can be equivalently written as :

$$\ln\left(\frac{\varepsilon}{\theta}\right) \leq |C| \ln(1 - q_i) \Longleftrightarrow |C| \geq \frac{1}{q_1} \ln\left(\frac{\theta}{\varepsilon}\right)$$

Observe that, under our assumptions, we have $q_1 = \Theta(1)$. It leads to :

$$|C| \geq q\theta \ln\left(\frac{\theta}{\varepsilon}\right) = O(\theta \ln(\theta/\varepsilon))$$

We can now summarize this result into the following proposition :

Proposition 4. *Suppose that $\theta = O(1)$ and $\theta_i = \Theta(m)$ with associated multiplicative constant $0 < q_i < 1$ for all $i \leq \theta$. Then a committee C is representative of $\mathcal{U}$ with probability $(1 - \varepsilon)$ if and only if $|C| = O(\theta \ln(\theta/\varepsilon))$*

Note that, here, θ is a constant, so $|C| = O(\ln(1/\varepsilon))$. It means that the size of the committee only depends on the targeted precision of the recommender.

4 Recommenders

In this section, we use the results of the previous sections to design a family of recommendation systems. Each of these recommendation systems depends on how we collect the information needed for initializing the algorithm.

We present here two of these algorithms, probably the most naturals that can be constructed using our framework. First we present two different ways to collect the informations that are needed by our algorithm. Please note that θ, the number of equivalence classes of users w.r.t. the relation "having same product sorting", is a parameter of the two initialization process. The natural question is then how to choose the value of this parameter in order to make the algorithm usable ? In practice θ is a constant known via users' polls. But if we consider the more general case where $\theta = O(\ln m)$, our algorithm is still efficient.

The first initialization process is used if one has only access to values of the utility function for users from the committee C.

Initialization case 1

- Set $C = \{u_{h_i}; i \leq |C|\}$ by picking $|C| = q\theta \ln\left(\frac{\theta}{\varepsilon}\right) = \Theta(\theta \ln(\frac{\theta}{\varepsilon})$ users from $\mathcal{U}$.
- Set W by picking $|W| = \max(8\ ;\ \lceil\sqrt{\theta}\rceil)$ products from $\mathcal{P}$.
- For all $u_k \in C$, Extract $\mathcal{F}_r(u_k)$ from $\mathcal{P}$.
- For all $u_k \in C$, build the family of functions $index|_W(\star, u_k)$. This family is represented as a restriction to C and W of the matrix S defined in subsection 2.1. We denote this matrix as $S_{C,W}$

The second initialization process is used when the user of the recommender can ask to committee members their r favorite products in P without rating all products in P. In our practical experiment, we use this initialization method.

Initialization case 2

- Set $C = \{u_{h_i}; i \leq |C|\}$ by picking $|C| = q\theta \ln\left(\frac{\theta}{\varepsilon}\right) = \Theta(\theta \ln(\frac{\theta}{\varepsilon})$ users from $\mathcal{U}$.
- Set W by picking $|W| = \max(8\ ;\ \lceil\sqrt{\theta}\rceil)$ products from $\mathcal{P}$.
- Ask each user from C about his r favorite products. Build a $|C| \times r$ table containing this information.
- Ask every member of the committee to sort products of W. Use this to build the family of functions $index|_W(\star, u_k)$. This family is represented as a restriction to C and W of the matrix S defined in subsection 2.1. We denote this matrix as $S_{C,W}$.

Note that $S_{X,Y}$ will denotes the restriction of S to users from X and products from Y. Moreover, $S(i, j) = index(j, u_i)$. We now give the algorithm that uses either initialization case 1 or 2.

```
Algorithm
Input : $u_i$ from $\mathcal{U}$ represented by its sorting on products from $W$. This can be seen as a vector $V_i = \big(index|_W(1, u_i), \ldots, index|_W(|W|, u_i)\big)$.
Output : The $r$-recommended products for $u_i$.
Behavior :
  1. Compute $S_{C,W} \cdot V_i := {}^t w(i)$
  2. Set $J = \{j \leq |C| : w(i)_j = \max_{l \leq |W|}\{w(i)_l\}\}$ where $w(i) = (w(i)_1 \cdots w(i)_{|W|})$
  3. Take $j_0 \in J$ at random uniformly
  4. Outputs the $r$ favorite products of committee user $u_{j_0}$ corresponding to ${j_0}^{th}$ row of matrix $S_{C,W}$ (This is $\mathcal{F}_r(u_{j_0})$).
```

For this algorithm, the following theorem holds :

Theorem 1. *The above algorithm gives good recommendation to a given user with probability at least* $1 - (\varepsilon + \eta)$*, where* $\eta = 1 - e^{-\frac{|W|^4}{2(|W|)!}} = o(1)$ *and* ε *is such that* $|C| = q\theta \ln\left(\frac{\theta}{\varepsilon}\right)$*. Its time complexity is* $O(\theta\sqrt{\theta}\ln\left(\frac{\theta}{\varepsilon}\right))$*.*

Proof. We have at least one user from each of the θ classes with probability $(1-\varepsilon)$ according to proposition 4. Let user u_i be the input of our algorithm. As there exists $j \leq \theta$ so that user $u_i \in \widetilde{u_j}$, we get that user u_i has a member of its class in C with the same probability $(1-\varepsilon)$. In this case, let us define $ref(i) = C \cap \widetilde{u_j}$. By proposition 3, since $|W| = \lceil\sqrt{\theta}\rceil$, two users from different classes have different corresponding rows in $S_{C,W}$ with probability $e^{-\frac{|W|^4}{2(|W|)!}} = 1 - \eta$.

Hence, there is a probability at least $1 - (\varepsilon + \eta)$ that the committee contains at least one user from each class (i.e. the committee is representative) and that its members have same representative in the matrix $S_{C,W}$ computed so that $S_{C,W}(k, \star) = V_i$ if and only if user $u_{h_k} \in ref(i)$. Thus, picking any of the user in $ref(i)$ will provide someone who share same product ordering as user u_i so that recommending his top r favorite products is a good recommendation for user u_i. It remains now to show that the set J is exactly $ref(i)$. For this we use the following lemma which is a direct application of the Cauchy-Schwarz inequality.

Lemma 2. *Let* τ *and* σ *be permutations over* $[N]$*,* $N \in \mathbb{N}^\star$*. We have that* $\sum_{i=1}^{N} \tau(i) \cdot \sigma(i)$ *is maximum if, and only if,* $\tau = \sigma$*.*

Hence, since every row l in $S_{\mathcal{U},W}$ can be seen as the restriction of the *index* function to the set W of witness products, this corresponds to enumerating every image of a sub-permutation. Thus, the vector ${}^t w(i)$ consists of the scalar product given by the previous lemma, so that $w(i)_k$ reaches its maximum if, and only if, $S_{C,W}(k, \star) = V_i$, permitting us to conclude.

The time complexity is in $O(|C| \cdot |W|) = O(\theta\sqrt{\theta}\ln\left(\frac{\theta}{\varepsilon}\right))$. Computational complexity for finding maximum coordinates of list $w(i)$ and taking j_0 at random among corresponding indexes can be neglected because of the O.

Table 1. Percentage of recommendations regarding the number of good and unknown recommended products

unknown	0		1		2		3		4		5	
good	Rec	Ran	Rec	Ran	Rec	Ran	Rec	Ran	Rec	Ran	Rec	Ran
0	0%	7%	1.1%	5.2%	1.3%	1.6%	4.7%	8%	22.2%	23.7%	100%	100%
1	4.3%	20.7%	5.7%	22.4%	16%	31.1%	16.3%	37.1%	77.8%	76.3%	0	0
2	19.1%	17.2%	14.8%	24.1%	20%	39.3%	**79%**	**54.9%**	0	0	0	0
3	23.4%	24.1%	33%	31%	**62.7%**	**28%**	0	0	0	0	0	0
4	19.1%	13.8%	**45.4%**	**17.3%**	0	0	0	0	0	0	0	0
5	34.1%	17.2%	0	0	0	0	0	0	0	0	0	0

With the previous theorem, we have a recommendation system for recommending to one user a set of r products. We now see what happens if we use this algorithm for making recommendations to all users of $\mathcal{U}$.

Corollary 1. *When adding a loop to the beginning of the algorithm in order to make recommendation for every user in $\mathcal{U}$, the time complexity of the algorithm is $O(m+n)$ when using initialization case 1 and $O(m)$ when using case 2, assuming that we both have $|C| = O(1)$ and $|W| = O(1)$. Thus, the fact that* every user gets a good recommendation *happens with probability $1-(\varepsilon+\eta)$.*

5 Experiments

In order to validate the effectiveness of our approach we decided to make an experiment with actual products and users.We selected from a set of 4400 movies, 160 of them uniformly at random. We then extract uniformly at random from this data set 9 movies (our witness products set).

Our methodology was then the following. From the people that volunteer to appear in the committee, we extract (at random) 20 of them. It was then asked, through a web site, to the committee members to sort the 9 movies and then to choose their 5 favorite movies in the data set (these 5 movies were called the selection). We then asked as many people as possible to use our recommendation engine to see if it is effective. 270 people have used it so far, the experiment is ongoing so the presented results are only partial but still trustworthy. The protocol was the following, first a user is asked to sort the 9 witness movies and then we offer him two recommendations. The first one is provided by the recommendation system presented in this paper and the second recommendation is simply composed of 5 movies chosen uniformly at random in the data set. The users are then asked two questions for each recommendation, how many films do they like in the recommendation and how many films do they actually know in the recommendation.

Fig.2. It shows the percentage of recommendations that contains at least a given number of good recommended products (e.g. products liked by user). It first gives the percentage of good recommendations (according to the definition 2). Our method outperforms the random recommendation since we achieve a percentage greater than 95% while the random recommendation only achieves

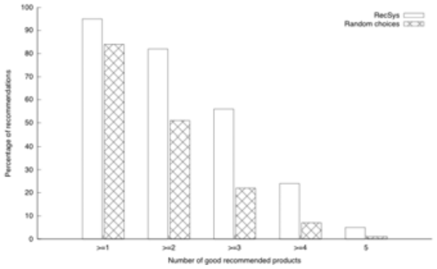


Fig. 2. Percentage of recommendations containing at least x good products

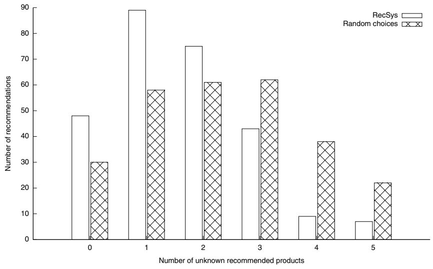


Fig. 3. Number of recommendations with x unknown products

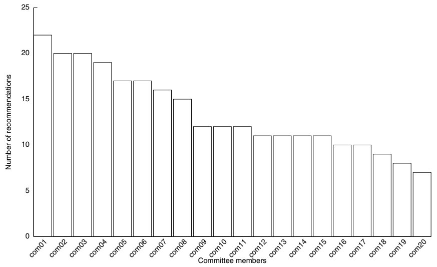


Fig. 4. Number of recommendations made by each committee member

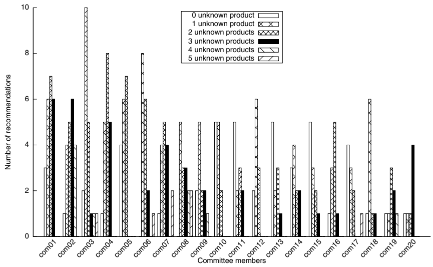


Fig. 5. Number of recommendations with x unknown products made by each committee member

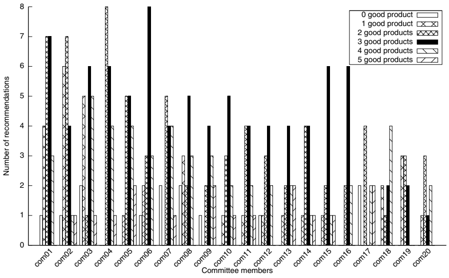


Fig. 6. Number of good recommendations made by each committee member

less than 85%. Moreover, we can see that our recommender also provides higher order good recommendations. On the other hand, random choices' effectiveness drops quickly and it often fails to provide users with more than 2 good products.

Fig. 3. It shows the number of recommendations w.r.t. the number of unknown products recommended. A too large number (i.e. 4 or 5) of unknown products seems to indicate a poor quality of recommendation since we are here dealing with well known movies: if one does not know items recommended too him, it is likely that they are in fact movies he did not want to see. It is also important to

note that if this number is too high it will decrease the user's confidence in the recommendation. We can clearly see on Fig.3 that this number decrease much faster with our algorithm than with the random choices.

Table 1. We compare the "quality" of the recommendations made by both techniques. Meaning we want to compare the number of good recommended products w.r.t. the number of unknown products in the recommendation. The columns are indexed by the number nu of unknown products in the recommendations and the rows are indexed by the proportion of recommendations with ng good products. We see that when $(ng, nu) \in \{(2,3),(3,2),(4,1)\}$ our algorithm outperforms the random choices. These cases are interesting because they concern good recommendations where $(ng + nu = 5) \wedge (ng > 1) \wedge (nu < 5)$, thus the user is confident in the recommendation (he liked all the movies he knows in the recommendation) and will probably consult the unknown products.

Figs.4, 5 and 6. These figures consider each committee member separately. We see in Fig.4 that most recommendations are given by only a few users, but that there are no users that make zero recommendation. We also see that the percentage of good products and unknown products in each committee member is approximately the same for every member and does not depend on the number of recommendations.

Experiments show the effectiveness of our approach: it is possible to provide users with good recommendations with high probability but low complexity.

6 Conclusion

In this paper we have presented a new recommendation system based on weaker assumptions than previous ones. Our recommender is as efficient in terms of time complexity and probability of having a good recommendation as other recommendation systems. A user satisfaction experiment supports these results.

References

1. Allen, R.B.: User model: theory methods and practice. International Journal of Man-Machine Studies, 511–543 (1990)
2. Goldberg, D., Nichols, D., Oki, B.M., Terry, D.: Using collaborative filtering to weave an information tapestry. Communications of the ACM, 61–70 (1992)
3. Awerbuch, B., Azar, Y., Lotker, Z., Patt-Shamir, B., Tuttle, M.: Collaborate with strangers to find own preferences. In: Proceedings of the Seventeenth Annual ACM Symposium on Parallelism in Algorithms and Architectures, pp. 263–269 (2005)
4. Drineas, P., Kerenidis, I., Raghavan, P.: Competitive recommendation systems. In: Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing, pp. 82–90 (2002)
5. Kleinberg, J., Sandler, M.: Using mixture models for collaborative filtering. Journal of Computer and System Sciences 74, 49–69 (2008); Learning Theory (2004)
6. Awerbuch, B., Patt-Shamir, B., Peleg, D., Tuttle, M.: Improved recommendation systems. In: Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1174–1183 (2005)

7. Resnick, P., Varian, H.R.: Recommender systems. Communications of the ACM, 56–58 (1997)
8. Mahoney, M.W., Maggioni, M., Drineas, P.: Tensor-cur decompositions for tensor-based data. In: Eliassi-Rad, T., Ungar, L.H., Craven, M., Gunopulos, D. (eds.) KDD 2006, pp. 327–336. ACM, New York (2006)
9. Kleinberg, J., Sandler, M.: Convergent algorithms for collaborative filtering. In: EC '03: Proceedings of the 4th ACM Conference on Electronic Commerce, pp. 1–10. ACM, New York (2003)

Appendix

Proof. (of proposition 1).
The existence is straightforward, so it remains to prove the unicity. Let $[n]$ and W be such fixed sets and suppose that there exists two different sub-permutation τ_1 and τ_2 of τ that satisfy the propriety (1). We then consider the smallest $i \in [|W|]$ such that $\tau_1(k) = \tau_2(l) = i$, with $k \neq l$ elements from W. It follows that $\tau_1(l) > \tau_1(k)$ and $\tau_2(k) > \tau_2(l)$: otherwise, $\tau_1(l)$ or $\tau_2(k)$ would be lower than i making k and l equal. But then, we have by (1) that $\tau(l) > \tau(k)$ and $\tau(k) > \tau(l)$, a contradiction. In any case, we are lead to $\tau_1 = \tau_2$ showing that the function is unique when W is fixed.

Proof. (of lemma 1).
By induction on the number θ of distinct permutations. The result stands for $\theta = 1$. As this is obvious for $\theta \geq n$, we will assume $\theta < n$.

Let us consider that $W \subset [n]$ is a set of cardinality at most $2(\theta - 1)$ satisfying $\forall i, j \ \leq \theta \ \ \tau_i|_W \neq \tau_j|_W$. Recall that is means $\tau_i(a) < \tau_i(b)$ while $\tau_j(a) > \tau_j(b)$ for some a, b belonging to W, a and b being case-specific for each $(i; j)$ with $i \neq j$. We then compare $\tau_{\theta+1}|_W$ to the $\tau_i|_W$'s. If they are all different, then we are done. Else, there exists some $i \leq \theta$ so that $\tau_i|_W \neq \tau_{\theta+1}|_W$. Moreover, such an i is unique among $[\theta]$, otherwise it would contradict the assumption that W allows to distinguish between $\tau_1, \ldots, \tau_\theta$. As all the $\theta+1$ permutations are distinct, there exists a, b such that $\tau_i(a) < \tau_i(b)$ while $\tau_{\theta+1}(a) > \tau_{\theta+1}(b)$. Let $W' = W \cup \{a; b\}$. It must be the case that $\tau_i|_{W'} \neq \tau_{\theta+1}|_{W'}$. Finally, $\tau_{\theta+1}|_W \neq \tau_j|_W \Rightarrow \tau_{\theta+1}|_{W'} \neq \tau_j|_{W'}$ so that W' permits to distinguish every of the $\theta + 1$ permutations and we have that $|W'| \leq 2(\theta - 1) + 2 = 2\theta$, and the result follows.

Author Index