Combining Collaborative Filtering and Text Similarity for Expert Profile Recommendations in Social Websites

Alexandre Spaeth and Michel C. Desmarais

École Polytechnique de Montréal,
C.P. 6079, succ. Centre-ville
Montréal (Québec) H3C 3A7, Canada
{alexandre.spaeth,michel.desmarais}@polymtl.ca

Abstract. People-to-people recommendation differ from item recommendations in a number of ways, one of which is that individuals add information to their profile which is often critical in determining a good match. The most critical information can be in the form of free text or personal tags. We explore text-mining techniques to improve classical collaborative filtering methods for a site aimed at matching people who are looking for expert advice on a specific topic. We compare results from a LSA-based text similarity analysis, a simple user-user collaborative filter, and a combination of both methods used to recommend people to meet for a knowledge-sharing website. Evaluations show that LSA similarity has a better precision at low recall rates, whereas collaborative filters have a better precision at higher recall rates. A combination of both can outperform the results of the simpler algorithms.

Keywords: Social Recommender Systems, Text Mining, People Recommendation, Content-based Recommender, Collaborative Filtering

1 Introduction

Recommender systems try to predict, among many items, the ones that a particular user might like, according to the user's preferences [8]. Those preferences are usually based on explicit ratings (item votes) or implicit data (browsing behaviour or buying habits). Those systems are widely used in e-commerce and their applications greatly increase the chance for the user to like the proposed item, and therefore, the chances of the user to buy it. A great number of methods to compute the best recommendations have been proposed, including SVD decomposition [9], using tags [7], or including content-based information [15] for example.

The application of those techniques to people-to-people recommendation has become an important topic in the last few years [2, 10, 13, 14]. The particularity of people-to-people recommendation is that the users and the items represent the same entity. Moreover, the sites that can generate people-to-people recommendations generally allow users to add free text and personal tags to describe

themselves, and to describe people they would like to meet or interact with. These particularities allow for the combination of techniques in a way that classical recommender systems cannot use [3].

The topic of expertise recommendation has been recently tackled. Often, the goal is to recommend experts to other experts, for example authors to co-authors [16], or teachers to teachers [1, 4]. Sometimes, it's also used in a learning environment context to recommend more advanced peers to users [12].

In this paper, we use latent semantic analysis to compute similarity between users and combine this approach with collaborative filtering. The experiments, conducted with a knowledge-based meeting website, show an improvement in recall and precision for the prediction of who will meet whom.

The rest of the paper is organized as follows. Next section presents the data. Section 3 presents the different algorithms we used and combined. Section 4 presents the evaluation framework and the results.

2 Data presentation

Social websites have plenty of data concerning the users. We used such data from an expertise requests and offers website [5]. The goal of this website is to facilitate the meeting between people, based on their respective expertise and needs. Users who are looking for help on a particular topic will post a query in the hope that someone with the expertise on this topic will answer the query. But the ultimate goal is that the two will actually meet face to face instead of posting answers or get involved in some other kind of electronic exchange.

The first step for a new user is to fill his profile. The member can then browse other profiles to find a matching expertise or use the search engine to look for interesting profiles. Figure 1 shows the browsing page. Finally, after completing his profile, he can also browse among our recommendations, as displayed in figure 2.

Each user's profile is composed of a small essay, geographical information, and a list of expertise offers and demands. Each expertise has a short free text description and some custom tags attached. The distribution of the number of tags per expertise is shown in figure 4. Each user is responsible for his own profile and for providing a correct description and tagging. It is possible to enter free tags but an auto-completion system provides suggestions and ensures that tags are correctly spelt.

Besides the data explicitly provided by the users, we also have access to the browsing behaviour of each user, including viewing events, messages and previous meetings. The distribution of the number of events per user is presented in figure 3.

The statistics for the data are gathered in table 1.

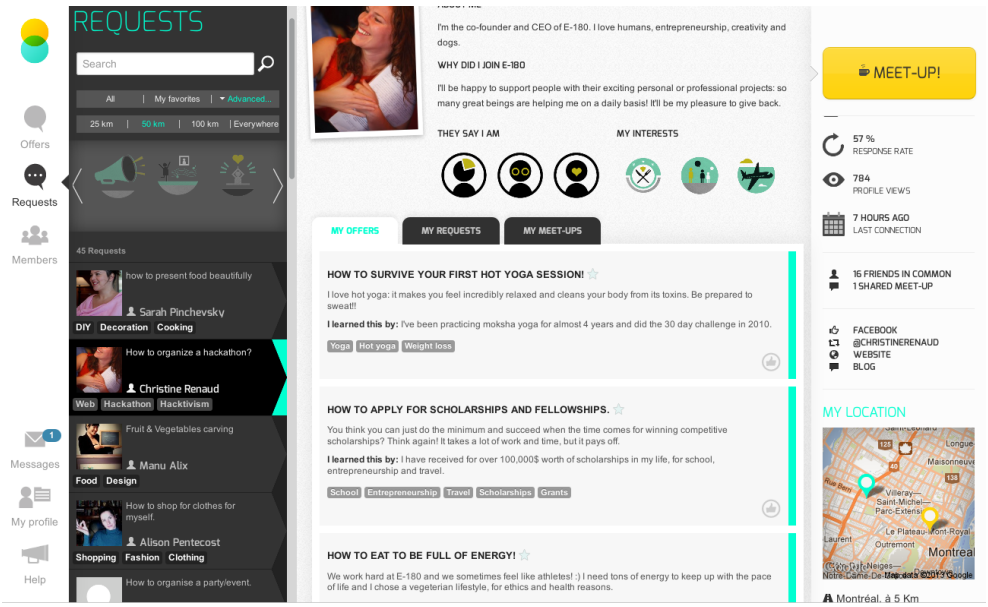

Fig. 1. Profile browsing page

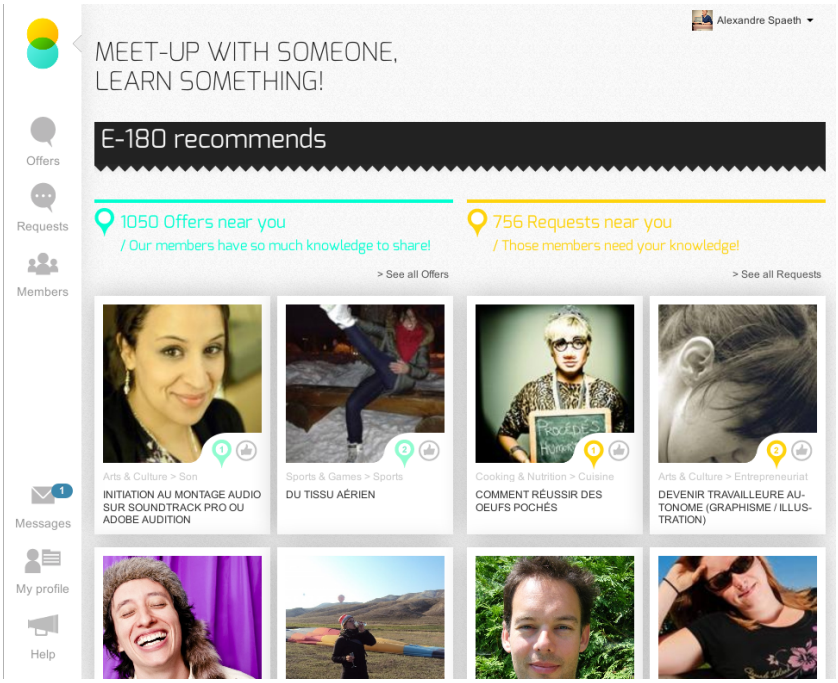

Fig. 2. Recommendations page

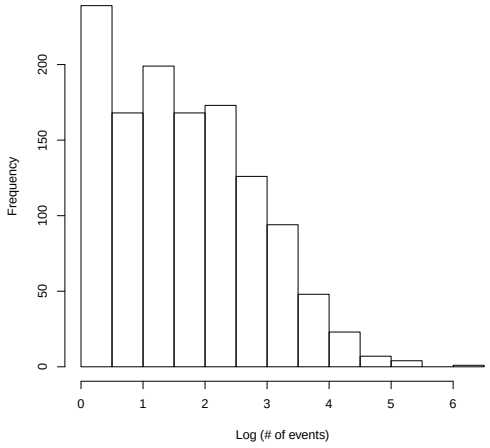

Fig. 3. Distribution of the number of events per user. Events include views, messages and meetings.

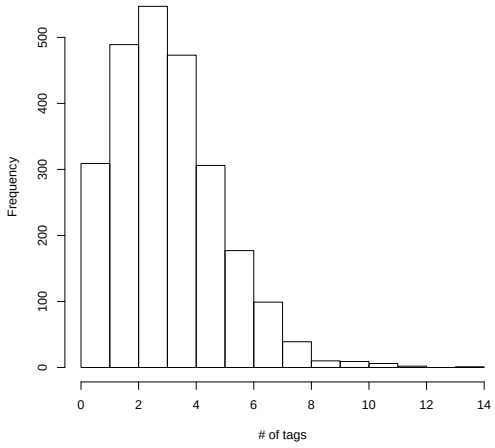

Fig. 4. Distribution of the number of tags per expertise description.

Table 1. Statistics about users' behaviour and expertise

Users	
Users with activity	1 655
Activities	
Browsing profiles	25 142
Messages	7 086
Meetings	557
Total activity (Fig 3)	32 785
Expertise	
Total of expertise offers	1 432
Users with an offer	793
Total of expertise demands	1 044
Users with a demand	651
Mean of number of words per expertise description	5.78
Mean of number of tags attached per expertise (Fig 4)	3.49
Mean of number of expertise per tag	9.85

3 Recommendations

Users become member of the website and use it for two reasons: to fill a need for a specific expertise, or by curiosity. People who are on the website to fill a need will meet people with an offer matching their queries. We would expect that classical information retrieval techniques should yield appropriate suggestions to those users. But for people without queries or offers, and also to address the need of the mere curious, other strategies are required and collaborative filtering is a good candidate.

3.1 Text similarity

Expertise queries are much like information retrieval queries. It is possible to use them to find a matching offer and then recommend a meeting between the two users.

After lemmatization of the expertise description text, we obtain the $\mathbf{M}_d$ matrix between words and expertise. To take into consideration that some words are more meaningful than others, we use the TF-IDF technique [6] to obtain $\mathbf{T}_d$ and compute the cosine similarity matrix $\mathbf{S}_d$.

With $|D|$ the total number of expertise descriptions and $|\{d_j : t_i \in d_j\}|$ the number expertise descriptions with the lemma t_i, we have:

$$idf_i = \log \frac{|D|}{|\{d_j : t_i \in d_j\}|} \tag{1}$$

Taking IDF as the vector of idf_i for all terms yields the TF-IDF matrix:

$$\mathbf{T}_d = \mathbf{M}_d \cdot \mathrm{IDF} \tag{2}$$

And finally, $\mathbf{S}_d$ is defined as:

$$\mathbf{S}_d = \frac{\mathbf{T}_d^{\mathrm{T}} \times \mathbf{T}_d}{||\mathbf{T}_d^{\mathrm{T}} \times \mathbf{T}_d||} \tag{3}$$

The score between two users is defined as the highest similarity between all the demands of one user and all the offers of the other one. Finally, for each user, we select the N best recommendations. We will refer to this technique as "text similarity".

3.2 Tag similarity

The text similarity technique applied over free text descriptions has two weaknesses: the vocabulary can be large, resulting in a sparce matrix, and we run into the polysemy issue (people can use different words to mean the same thing). These weaknesses can be alleviated with tags because the auto-completion feature and the existing profiles tags provide suggestions that end up reducing the vocabulary space, and because users may have a natural tendency to avoid ambiguous terms for the choice of tags.

Using the same algorithm as described in 3.1, but using tags instead of lemmatized descriptions, we compute recommendations based solely on tag similarity. Starting from the knowledge-tags adjacency matrix $\mathbf{M}_t$, we compute the TF-IDF matrix $\mathbf{T}_t$ and then the similarity matrix $\mathbf{S}_t$. This technique is referred to as "tag similarity".

3.3 Combining tags and text similarity with latent semantic analysis

We combine the tags and text similarity measures into a single space and use Latent Semantic Analysis (LSA) in the hope of increasing the relevance of the similarity measure between the queries and the expertise offers.

Firstly, we build a new matrix $\mathbf{M}_c$ of tags and words combined in the column space and expertise profiles the row space. We use the TF-IDF technique to build the new weighted matrix $\mathbf{T}_c$ and compute the cosine similarity between expertise descriptions $\mathbf{S}_c$.

As described in [11], the latent semantic analysis can merge terms and tags together into concept dimensions and, using the singular value decomposition, it can thereby reduce the number of dimensions. In order to do this, the matrix $\mathbf{T}_c$ were decomposed using singular value decomposition: $\mathbf{T}_c = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$. We reduced the number of dimensions by nullifing the smallest values of $\mathbf{\Sigma}$ and build a new $\mathbf{T}_{\mathrm{SVD}}$ matrix and a new cosine similarity between expertise descriptions $\mathbf{S}_{\mathrm{SVD}}$. We tried several dimensions and 50 latent factors seemed to give the best results. Future work should be done to validate and confirm this value.

3.4 Collaborative filtering

The collaborative filtering approach uses browsing data along with two levels of interactions, messages and meetings. We create a vote matrix $\mathbf{M}_{CF}$ initialized to 0 and add the following values to the corresponding entry in the matrix:

Seen profile: 1 – the user (column) has seen the potential target's profile (row)
Message: 2 – the user (column) has sent a message to the potential target (row)
Meeting: 4 – the user (column) has met the target (row)

The choice of the values 1/2/4 is based on the intuition of their respective importance, and no attempt to optimize it was made for this study. This will be done in a future work.

The values are additive. For example, if a user has seen a profile and sent a message, the resulting value will be 3. Although unlikely, it's possible for a user to meet someone else without viewing his profile: by answering directly to a meeting request without checking the profile of the requesting user.

The vote matrix $\mathbf{M}_{CF}$ can be considered a directed graph where weights are assigned to the edges. Figure 5 shows an example of such structure, with S and R representing users.

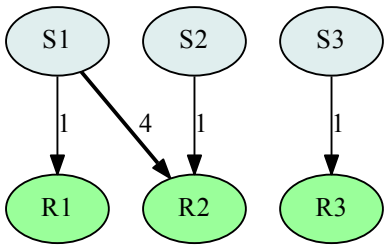

Fig. 5. Weighted graph of relationships.

The first step towards a personalized recommendation is the similarity calculation: how to evaluate the proximity between two users? As usual in recommender systems, in order not to bias recommendations towards targets who are registered for a long time (and therefore with more votes), we rely on the cosine to measure similarity which eliminates such bias.

Considering the previously defined matrix $\mathbf{M}_{CF}$, the similarity is given by:

$$\mathbf{S}_{CF} = \frac{\mathbf{M}_{CF}^{\mathrm{T}} \times \mathbf{M}_{CF}}{||\mathbf{M}_{CF}^{\mathrm{T}} \times \mathbf{M}_{CF}||} \tag{4}$$

The process of finding a recommendation in a graph like the one represented figure 5 is much like searching for $S_2 \rightarrow R_2 \leftarrow S_1 \rightarrow R_1$ links.

The cosine calculation between users helps to find all the $S_1 \leftrightarrow S_2$ links and the relevant links are calculated with: $\mathbf{R}_{CF} = \mathbf{S}_{CF} \times \mathbf{M}_{CF}^{\mathrm{T}}$ or $\mathbf{M}_{CF} \times \mathbf{S}_{CF}$. The results obtained are the number of $S \rightarrow R \leftarrow S \rightarrow R$ links found, divided by the number of outgoing links, because we use the cosine similarity. Finally, we must remove the existing links to have a weighted list of recommendations.

3.5 Combining tags and text similarity with collaborative filtering

The first technique, combining tags and description similarity, gives scores between expertise queries or expertise offers. The recommendation scores is given by the technique described in section 3.1 and is referred to as the $\mathbf{R}_{desc+tags}$ matrix between users.

The collaborative filtering technique gives recommendation scores between users directly, $\mathbf{R}_{CF}$.

We combine these scores by calculating a weighed geometrical mean between the two values. We tried several weights, as well as a weighted arithmetical mean and the best results were achieved with these values:

$$\mathbf{R}_{CF+tags+sem} = (\mathbf{R}_{CF} + 1)^{1/3} * (\mathbf{R}_{desc+tags} + 1)^{2/3} \tag{5}$$

Those scores should be validate in further experiments.

4 Experiments

4.1 Evaluation framework

To assess the results of the different algorithms, we trained them with data from before October 1st and tested on data after October 1st 2012.

A recommendation is successful if the user viewed the profile and met the recommended profile. A recommendation is unsuccessful if the user viewed the profile and did not meet the recommended profile. In the test data set, we observe 452 meetings for 13,894 view events, i.e. 3.25%. This value is our baseline and is the expected precision rate for a random recommender system. A gold standard based on manual recommendations is described below and will serve as an upper comparison point.

We calculated the recall-precision curve for each algorithm, varying the number of recommendations per user.

$$precision = \frac{\text{correctly recommended user}}{\text{number of recommendations}} \tag{6}$$

$$recall = \frac{\text{correctly recommended user}}{\text{number of good recommendations}} \tag{7}$$

It is important to note that even with a perfect recommender system, we cannot realistically expect to obtain a 100% precision rate. Indeed, the leap

from browsing a profile to participating in a face to face meeting is large in terms of engagement, and many are not willing to make this leap. So far, there are only few people that are moving from browsing and messaging to actually meeting and our algorithm provides a signifcant improvement, compared to a random recommender.

To have a sense of the best result we could expect, we made 282 manual recommendations. For those recommendations, we sent an email to the users, explaining in sentences why they should meet. The precision-recall rates of these recommendations serve as our gold standard.

4.2 Results

First, we tried each of the three algorithms separately. The recall-precision curve are reported in figure 6.

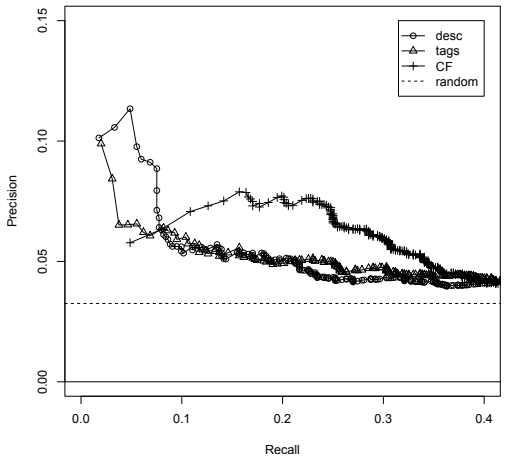

Fig. 6. Three basic algorithms

Note that while tags and text similarity techniques have a better precision at smaller recall rates, the collaborative filter gives better precision at higher recall rates.

The combination between tags and text similarity is shown in figure 7.

Globally, combining tags and text similarity data gives a better precision than the individual techniques, but only at recall rates between 0.05 and 0.4. Text similarity is slightly better between 0 and 0.05, i.e. with few recommendations.

Finally, the combination of the text similarity and tags recommendations algorithm with the classical collaborative filtering is shown in figure 8. This

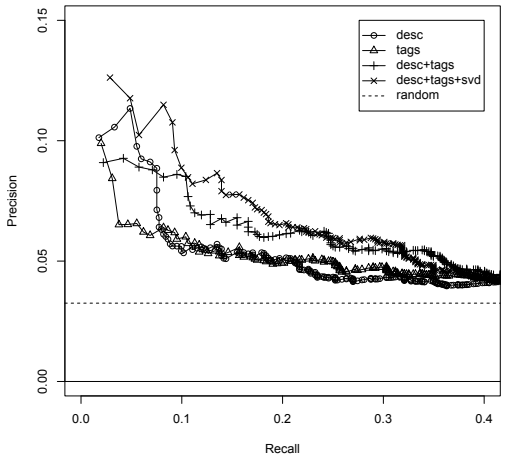

Fig. 7. Combination of text similarity and tags

Figure also reports the results obtained with our gold standard, the manual recommendations.

For fewer recommendations, i.e. at a smaller recall rate, the text similarity algorithm is still better, but if we want to do more recommendations, the combination approach performs better than any other algorithm. Furthermore, the precision obtained by our manual recommendations is slightly better than our algorithm, but this was expected because it is likely that human judgement will have a relatively high accuracy and, moreover, the personal email is an incentive to contact and meet the person that is not given in the other conditions and adds a positive bias. The fact that the best algorithm closely approaches this gold standard is encouraging.

5 Conclusions

People-to-people recommendations is a rapidly growing field that generates strong research interest. This study explores a particular and relatively new type of people-to-people recommendation, namely recommending people with sought expertise profiles. We investigate how text similarity and collaborative filtering techniques can be combined to outperform each individual technique.

Our results show that open text descriptions and tags can be combined in a single semantic space and that LSA can be applied to this space to further improve this technique. When performing only one recommendation per user, this approach of combining techniques is shown to almost match manual recommendations, which benefits from a positive bias provided by a personal email

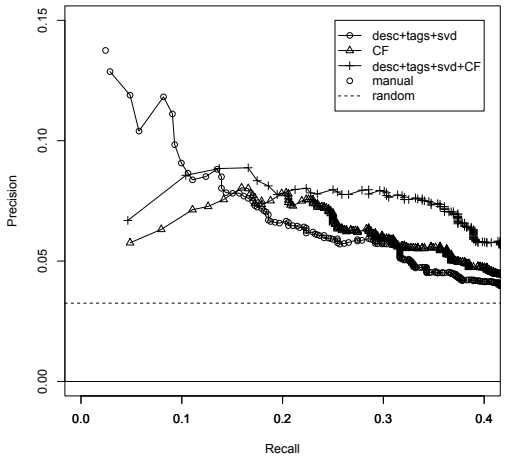

Fig. 8. Recall-precision for the combination

incentive. This is quite encouraging. The collaborative approach is shown to provide more accurate recommendations at higher recall rates, thereby providing an alternate source of recommendations under the condition where larger number of recommendations are required.

In future work, we will do more detailled analysis on the recommendations, especially to determine how significant is the contribution from each separated approach. Furthermore, we intend to use other available information, such as meeting evaluation, geographical data and friendship links to improve our results. We should also spend some time improving the collaborative filter, by validating different values we used in this study, as well as by using similarity and other people-to-people particularities. Finally, we will measure the effect of the third person in the recommendation. Are people more willing to meet someone if a third party is a warrant?

References

1. B. Bahritidinov, E. Sanchez, and M. Lama. Recommending teachers for collaborative authoring tools. In *Advanced Learning Technologies (ICALT), 2011 11th IEEE International Conference on*, pages 438–442, July 2011.
2. Xiongcai Cai, Michael Bain, Alfred Krzywicki, Wayne Wobcke, YangSok Kim, Paul Compton, and Ashesh Mahidadia. Collaborative filtering for people to people recommendation in social networks. In Jiuyong Li, editor, *AI 2010: Advances in Artificial Intelligence*, volume 6464 of *Lecture Notes in Computer Science*, pages 476–485. Springer Berlin Heidelberg, 2011.
3. Xiongcai Cai, Michael Bain, Alfred Krzywicki, Wayne Wobcke, YangSok Kim, Paul Compton, and Ashesh Mahidadia. Learning to make social recommendations: A

model-based approach. In Jie Tang, Irwin King, Ling Chen, and Jianyong Wang, editors, *Advanced Data Mining and Applications*, volume 7121 of *Lecture Notes in Computer Science*, pages 124–137. Springer Berlin Heidelberg, 2011.
4. Soude Fazeli, Francis Brouns, Hendrik Drachsler, and Peter Sloep. Exploring social recommenders for teacher networks to address challenges of starting teachers. 2012.
5. E-180 inc. E-180 platform. `http://www.e-180.com`, March 2013.
6. Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.
7. Heung-Nam Kim, Ae-Ttie Ji, Inay Ha, and Geun-Sik Jo. Collaborative filtering based on collaborative tagging for enhancing the quality of recommendation. *Electronic Commerce Research and Applications*, 9(1):73 – 83, 2010.
8. Joseph A. Konstan, Bradley N. Miller, David Maltz, Jonathan L. Herlocker, Lee R. Gordon, and John Riedl. Grouplens: applying collaborative filtering to usenet news. *Commun. ACM*, 40(3):77–87, March 1997.
9. Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, August 2009.
10. Sangeetha Kutty, Lin Chen, and Richi Nayak. A people-to-people recommendation system using tensor space models. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, SAC '12, pages 187–192, New York, NY, USA, 2012. ACM.
11. Thomas K. Landauer, Peter W. Foltz, and Darrell Laham. An Introduction to Latent Semantic Analysis. *Discourse Processes*, 25(2-3):259–284, 1998.
12. Felix Modritscher, Barbara Krumay, Sandy El Helou, Denis Gillet, Sten Govaerts, Erik Duval, Alexander Nussbaumer, Dietrich Albert, Ingo Dahn, and Carsten Ullrich. May i suggest? three ple recommender strategies in comparison. May 2011.
13. Luiz Pizzato, Tomek Rej, Thomas Chung, Irena Koprinska, and Judy Kay. RECON: a reciprocal recommender for online dating. In *Proceedings of the fourth ACM conference on Recommender systems - RecSys '10*, RecSys '10, pages 207–214, New York, NY, USA, 2010. ACM.
14. LuizAugusto Pizzato, Tomek Rej, Kalina Yacef, Irena Koprinska, and Judy Kay. Finding someone you will like and who won't reject you. In JosephA. Konstan, Ricardo Conejo, JoséL. Marzo, and Nuria Oliver, editors, *User Modeling, Adaption and Personalization*, volume 6787 of *Lecture Notes in Computer Science*, pages 269–280. Springer Berlin Heidelberg, 2011.
15. J. Salter and N. Antonopoulos. CinemaScreen Recommender Agent: Combining Collaborative and Content-Based Filtering. *IEEE Intelligent Systems*, 21(1):35, 2006.
16. Rory L. L. Sie, Hendrik Drachsler, Marlies Bitter-Rijpkema, and Peter Sloep. To whom and why should i connect? co-author recommendation based on powerful and similar peers. *Int. J. Technol. Enhanc. Learn.*, 4(1/2):121–137, July 2012.