Stochastic Matching and Collaborative Filtering to Recommend People to People

Luiz Augusto Pizzato
School of Information Technologies
University of Sydney, NSW 2006, Australia
luiz.pizzato@sydney.edu.au

Cameron Silvestrini
School of Information Technologies
University of Sydney, NSW 2006, Australia
cameron@silvestrini.com.au

ABSTRACT

The bias towards popular items is not necessarily an undesired outcome of recommender algorithms since a large amount of revenue on e-commerce websites is drawn from these popular items. On the other hand, in domains such as online dating and employment websites, where users and items of the recommendation are both people, a strong bias towards popular users may cause these users to feel overwhelmed and unpopular users to feel neglected. In this paper, we use collaborative filtering (CF) to generate recommendations for all users, and by using stochastic matching we select a number of reciprocal recommendations for each user that maximizes the matches among all users. In this way, all users, regardless of their popularity, will receive the same number of recommendations the number of times they will be recommended to others. This study is the first to apply a stochastic matching solution to balance the number of recommendations given to users in a CF setting. Using historical data, we demonstrate that the proposed recommender improves the chance of finding a successful relationship in comparison to CF recommendations.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering*

General Terms

Algorithms, Experimentation

Keywords

Recommender systems, Stochastic Matching, People Recommender

1. INTRODUCTION

There is a known bias towards popular items in most recommender algorithms [9]. This bias is not entirely an undesirable effect, since popular items can derive a large amount of revenue for e-commerce websites. However, for domains where people are recommended to other people such as online dating, high popularity can have some unwanted side effects. Controlling the popularity of these users is a very important issue for people recommender systems. Popular users are popular because they have some characteristics that are highly desired by other users, and having them in the system is extremely beneficial. They might be responsible for a large proportion of revenue, and they can boost the perception of usefulness for the other users. However, if these popular users are overused (i.e. allowed to be contacted by others too many times), they may feel overwhelmed and leave the system altogether.

On the other side of the popularity spectrum, we can find unpopular users, who receive very little to no attention. Most recommender systems will favor what has been previously liked, and unpopular users have very few likes, resulting in a bias against unpopular users. These unpopular users may be feeling neglected, which can harm their experience and cause them to leave the website.

When looking into the popularity spectrum and people recommender systems, there is a right balance between the number of recommendations that each user should receive and the number of times they should be recommended to other users. Depending on how users can access other people's profiles, popular users may receive lots of attention regardless of the number of times they are present in other people's recommendations. However, unpopular users may only receive attention if they are present in the right person's recommendations, which may involve sending the recommendation to many people.

In this paper, we propose an approach to remove the popular user bias by ensuring the every user receives the same number of recommendations as they are recommended to others. Our approach uses collaborative filtering (CF) combined with a stochastic matching approach in order to maximize the number of successful matches across different levels of popularity. In our approach, two users are always recommended to each other. That is, a user A is only recommended to a user B if user B is also recommended to a user A. This ensures that popular users are not recommended more often than unpopular users.

Several recent studies have used online dating data in their recommender systems, for instance [2, 14, 5, 13] focus on finding a ranked list of users whose chance of a positive interaction with another user is higher for those users near the top of the list than for those users near the bottom. A

RecSys'11, October 23–27, 2011, Chicago, Illinois, USA.

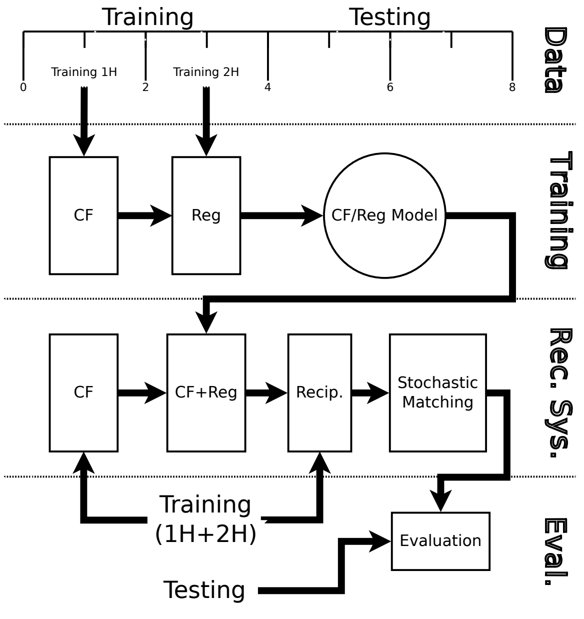

Figure 1: Stochastic Matching CF Approach

few recent studies [14, 3, 10] have shown the positive effect of reciprocal preference in people recommenders.

For job recommender systems, [12] suggest implementing a pareto optimal solution to find good overall matchings between employee and employers. Stable matching algorithms such as Gale and Shapley [6] are designed to match people to people in areas such as the stable marriage problem, stable room mate problem, and assigning papers to reviewers [8].

Given that in online dating relationships are undirected and are either successful or not (i.e. we cannot compare one relationship to another), we treat this matching problem as a stochastic matching problem as defined by [4]. In stochastic matching, we have a list of people and the probabilities that these users will successfully connect to each other in a relationship. Given these probabilities, the matching algorithm tries to find the best combination of matches that will result in the highest number of successful relationships.

This paper is organized as follows: Section 2 describes how the experiment was conducted. Section 3 describes the algorithms used for training process and testing, while Section 4 presents the evaluation. In Section 5, we present some concluding remarks.

2. EXPERIMENT SETUP

The study conducted in this paper is summarized by Figure 1: we have used data from an online dating website, where users contact each other and have the opportunity to either respond positively or negatively to each contact. We have used 8 weeks of data, where the first 4 weeks were used for training and last 4 weeks for testing. The whole dataset contains nearly 900,000 interactions of more than 30,000 users who reside in Sydney, Australia.

Our training phase uses the first half of the training data to create a set of recommendations using CF. Following this stage, we found a regression model (CF/Reg Model) to transform each CF score into the probability of the user sending a message in the second half of the training data.

To generate the recommendations, we first apply CF over the whole 4 weeks of the training data, then using the regression model we transform the CF scores into probabilities of communication between the users receiving the recommendations and the recommended users. After this, we convert the one-way probabilities into a reciprocal probability, which represent the probability of a user A liking a user B and user B liking A back or vice-versa (i.e. B initiating the liking action). After these probabilities are computed, we can apply a stochastic matching algorithm to obtain the best overall reciprocal recommendations. Finally, our evaluation is carried out using the 4 weeks of the testing set.

3. STOCHASTIC MATCHING COLLABORATIVE FILTERING

As illustrated by Figure 1, in order to find the best overall N recommendations for all users in an online dating setting, we implemented a stochastic matching algorithm that works on recommendations given by CF. In Section 3.1, we describe our CF implementation; then in Section 3.2, we describe the stochastic matching approach; finally in Section 3.3, we describe how these components work as one recommender.

3.1 Collaborative Filtering

Using the historical data from previous user interactions, our CF recommends users based on the number of messages the recommended users have received from the group of users with a similar behaviour to the user receiving the recommendations.

Given a set of interactions I between all users x and y in the set of users U:

$$I = \{\overrightarrow{xy} \ : \forall x \in U, \forall y \in U\}$$

and given the set of users $R(x)$ who received messages from a user x, and the set of users $S(x)$ who sent messages to x:

$$R(x) = \{y | \overrightarrow{xy} \in I\} \qquad S(x) = \{y | \overrightarrow{yx} \in I\}$$

our collaborative filtering $CF(x)$ recommends for user x a list of users u, who have received messages from users x' who have sent messages to the same users y that x has messaged, but were not messaged by x. More precisely, our CF recommender is defined as:

$$\begin{aligned} CF(x) = \{u \in R(x')\backslash R(x) \quad | \quad & \forall y \in R(x) : x' \in S(y), \\ & \forall x' \in S(y) : y' \in R(x')\}\} \end{aligned}$$

Each recommendation is scored according to the number of times the connection between message sender and message receiver was traversed until it found the user y'. This collaborative score is represented by $CFS(x, y')$. Figure 2 illustrate the recommendation process and the weights given for each y'. We can observe that *Earl* and *Gustav* are recommended to *Alice* because they were both messaged by *Carmen* and *Fern*, who sent messages to users also messaged by *Alice* (y). In this implementation of CF, because *Carmen* and *Alice* have messaged two users in common (*Bob* and *Daniel*), all users messaged by *Carmen* and not messaged by *Alice* (*Earl*) will be added a score of two. For the example of Figure 2, *Earl* with a score of 2 will be recommended before *Gustav*, who has a score of 1.

This CF approach will give higher scores for users who received more messages from a group of alike users. This

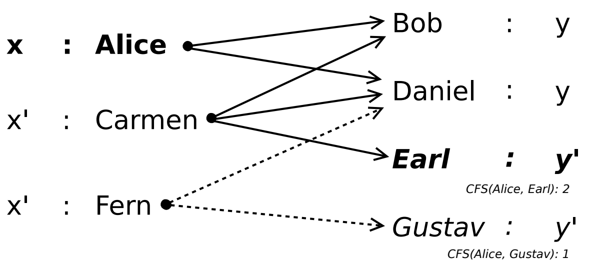

Figure 2: CF examples

type of approach can favour popular users over unpopular ones. In the next section we explain how stochastic matching works and how it can help to minimize this effect in CF recommendations.

3.2 Stochastic Matching

Given a list of probabilities of successfully matching two people, the stochastic matching algorithm focuses on finding N matches for each person such that it maximizes the number of expected matches for the whole population.

For this problem, the online dating recommendations are represented as an undirected, weighted graph $G = \{V, E\}$, where people are modeled as vertices V and the possible connections between people as edges E.

The graphs edges $e = (u, v) \in E$ represents that the two distinct people u and v were recommended to each other. In our stochastic matching problem, the final set of edges is whether a successful connection was established. Initially, the outcome of the recommendation is unknown and each edge e is associated with a independent probability $P(\overleftrightarrow{uv})$ that the recommendation is successful. Therefore, maximizing the number of successful relationships is equivalent to finding the maximum number of matches for the edge set E.

Chen et al. [4] describe that after a successful match, both people would leave the dating site, and they would be removed from the graph. In practice, a successful match in online dating does not necessarily remove the user from the dating game; in particular when the definition of success may be simply the agreed communication between two users (as in this study). Therefore, our stochastic matching approach does not remove the vertices of successful edges.

Chen et al. [4] also noticed that people have limited patience, which would cause a user to voluntarily leave the dating website after a number of unsuccessful matches. However, patience levels are very hard to define a priori, in particular because someone's patience may change with time. For instance, at a certain point, the users level of patience is high, but after receiving a very bad recommendation they may feel annoyed and decide to leave the website. In this study, we use patience levels N as a way to provide a fixed number of recommendations for each user.

In summary, the stochastic matching problem selects N edges per user in G based on the probability of occurrence of each edge e such that the number of successful edges found are maximized. Different algorithms have been proposed to deal with the stochastic matching problem.

We have implemented several approaches to stochastic matching, such as greedy [4], independent and dependent rounding [7], and their combination with linear programming rounding [1]. Unfortunately, we were unable to run any of the above algorithms except greedy with our time and memory constraints for the amount of data reported in this experiment. However, we observed that with the patience level set to 1, greedy outperforms all the other algorithms in the number of successful matches and in speed/memory performance.

The greedy algorithm works as follows:

1. order all edge probabilities $P(\overleftrightarrow{uv})$ in descending order;
2. remove the top most edge (u, v) and check if either u or v have been recommended N times;
    - if not, recommend v to u and u to v.
3. repeat step 2 until no more edges exists;

3.3 Recommender system setup

Because the stochastic matching algorithms need probabilities in order to process the best N number of matches, we need to convert the CF integer scores to probabilities scores. We performed the conversion between CF scores and probabilities by finding a regression function using a normalized CF score from the Training 1H data set and the probabilities of each recommendation in Training 2H. We experimented with different normalization for the CF scores, but we found the best strategy was to normalize the CF scores based on the largest CF score for all users. The regression model with best coefficient of determination is logistic ($R^2 = 0.5914$) such that $P(\overrightarrow{uv})$ gives the probability of user u sending a message to user v based on the CF score $CFS(u, v)$.

$$P(x) = 0.0128 + 0.0036 \times log(x)$$

Having the CF scores and the regression model at hand, we obtain a set of recommendations and probabilities of these recommendations to be accepted by the user. However, because a successful interaction between two users is depended upon the positive response of the recommended user, we aggregate the probabilities $P(\overrightarrow{uv})$ with the probability $P(v_+)$ of v responding positively to a message from any user; therefore giving the probability $P(\overrightarrow{uv_+})$ of a successful interaction between the users u and v when initiated by u.

$$P(\overrightarrow{uv_+}) = P(\overrightarrow{uv}) \times P(v_+)$$

The probability of a successful relationship $P(\overleftrightarrow{uv})$ between two users u and v is independent of whether u or v have initiated the communication, such that:

$$P(\overleftrightarrow{uv}) = P(\overrightarrow{uv_+}) + P(\overrightarrow{vu_+}) - P(\overrightarrow{uv_+}) \times P(\overrightarrow{vu_+})$$

After having these probabilities, we can apply a stochastic matching algorithm that obtains the best N recommendations for all users, while maximizes the chance of a successful relationship.

4. EVALUATION

To evaluate our approach, we analyzed whether a recommendation approach can increase the chance of a successful interaction. We measured the success rate $S@N$ for different values of N, which represent the proportion of successful interactions over all known interactions found in the recommendation set (N recommendations per user).

There is an overall 0.22% chance that a single recommendation chosen randomly from all possible recommendations

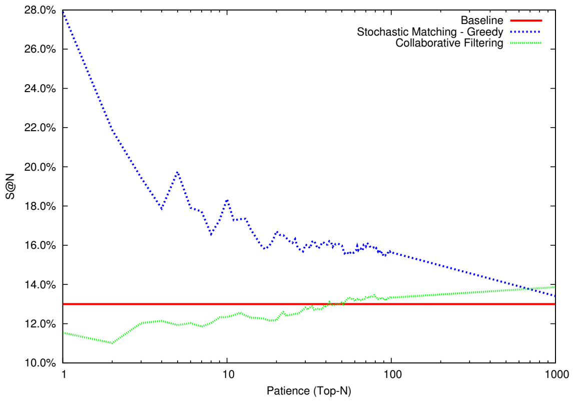

Figure 3: Comparison between CF and greedy

results in a message being sent from one user to another, and a chance of 0.03% that a successful interaction occurs. This gives a baseline $S@N$ of 13.87%. Using this value, we can establish whether the stochastic matching algorithm is performing better than random and is selecting recommendations that are likely to help users to find better matches.

Figure 3 shows the success rate for the recommendations given by the greedy stochastic matching algorithm and the recommendations given by the CF directly (without applying a stochastic matching). We can observe that the stochastic matching process increases the success rate of the CF recommendations by selecting recommendations that are more likely to result in a successful interaction. The baseline value shows that the stochastic process performs better than randomly selecting CF recommendations. It is important to notice that the CF algorithm does not consider the preferences of the users being recommended; i.e. CF is not optimized for positive responses.

We also observed that some users are the first (top-1) recommendation for more than 150 different users, and some are recommended more than 1,000 times in the whole set of CF recommendations. The stochastic matching approach removes the problem of overloading popular users by providing the same number of recommendations to users as the number of times they are recommended to others.

5. CONCLUDING REMARKS

This research presented the first attempt to apply stable matching algorithms to people recommenders. We have shown that stochastic matching can improve the results of CF in a people recommender setting, in particular when considering positive replies for messages. We have also shown that by using a greedy stochastic matching approach we can remove the popular/unpopular user problem of CF.

We have implemented different algorithms for stochastic matching that showed to be too resource intensive for the single processing architecture we used for these experiments. Future research should investigate if they can applied to cloud computing environment and what results they could yield. Other algorithms such as the ones described in [11] and Gale and Shapley [6] could be adapted to this problem.

We will also carry further investigations on ways to find probabilities for each edge from recommendations. We will investigate the use of different types of CF implementations including different ranking that account for reciprocity (e.g. [3]) and the use of the content-based reciprocal compatibility scores of RECON [14].

Acknowledgements

This research was funded by the Smart Services Co-operative Research Centre.

6. REFERENCES

[1] N. Bansal, A. Gupta, J. Li, J. Mestre, V. Nagarajan, and A. Rudra. When LP is the cure for your matching woes: Improved bounds for stochastic matchings. In *Algorithms – ESA 2010*. Springer, 2010.

[2] L. Brožovský and V. Petříček. Recommender system for online dating service. *CoRR*, abs/cs/0703042, 2007.

[3] X. Cai, M. Bain, A. Krzywicki, W. Wobcke, Y. Kim, P. Compton, and A. Mahidadia. Collaborative filtering for people to people recommendation in social networks. In *Advances in AI*. Springer, 2011.

[4] N. Chen, N. Immorlica, A. Karlin, M. Mahdian, and A. Rudra. Approximating matches made in heaven. In *Automata, Languages and Programming*. 2009.

[5] F. Diaz, D. Metzler, and S. Amer-Yahia. Relevance and ranking in online dating systems. In *33rd international conference on Research and development in information retrieval (SIGIR)*, 2010.

[6] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69:9–15, 1962.

[7] R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan. Dependent rounding and its applications to approximation algorithms. *J. ACM*, 53:324–360, 2006.

[8] M. Huang, B. Liu, and L. Hong. On assigning papers to reviewers. In *2nd International Workshop on Database Technology and Applications*, 2010.

[9] T. Jambor and J. Wang. Optimizing multiple objectives in collaborative filtering. In *4th ACM conference on Recommender systems*, 2010.

[10] Y. Kim, A. Mahidadia, P. Compton, X. Cai, M. Bain, A. Krzywicki, and W. Wobcke. People recommendation based on aggregated bidirectional intentions in social network site. In *Knowledge Management and Acquisition for Smart Systems and Services*. Springer, 2010.

[11] N. Leblanc, R. Morrison, and H. Redway. A match made in silicon: Marriage matching algorithms for dynamic microsimulation. In *2nd general conference of the International Microsimulation Association*, 2009.

[12] J. Malinowski, T. Keim, O. Wendt, and T. Weitzel. Matching people and jobs: A bilateral recommendation approach. In *39th Annual Hawaii International Conference on System Sciences*, 2006.

[13] B. McFee and G. Lanckriet. Metric learning to rank. In *27th International Conference on Machine Learning (ICML'10)*, 2010.

[14] L. Pizzato, T. Rej, T. Chung, I. Koprinska, and J. Kay. Recon: a reciprocal recommender for online dating. In *4th ACM conference on Recommender systems*, 2010.