A People-to-People Content-Based Reciprocal Recommender Using Hidden Markov Models

Ammar Alanazi
King Abdulaziz City for Science and Technology
Computer Research Institute
Riyadh, Saudi Arabia
salanazi@kacst.edu.sa

Michael Bain
The University of New South Wales
Faculty of Engineering
Sydney, Australia
mike@cse.unsw.edu.au

ABSTRACT

Users of online social networks such as dating websites often need help to find successful matches. People-to-people recommender systems can be used in social networks to help users find better matches, which requires solving the problem of *reciprocal* recommendation. However, most existing reciprocal recommenders use either profile similarity or interaction similarity to recommend new matches, without considering *temporal* features. In this paper we introduce a method for temporal reciprocal recommender systems using Hidden Markov Models to generate recommendations. Instead of summarising the whole historical data in one past state, we propose a model that formalises historical data on interactions as a series of successive states changing over time and then tries to find the recommended next state. We have implemented this new approach and the results of testing on industrial-scale data from a real dating website show a noticeable improvement over the previous best-performing recommenders.

Keywords

Recommendation Systems; Content-Based; Hidden Markov Models; People-to-people

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information filtering; I.5 [**Pattern Recognition**]: Models—*Statistical*; J.4 [**Social and Behavioural Sciences**]: Sociology

1. INTRODUCTION

Most of the existing work on recommender systems is built on the assumption that users' behaviours are static and do not change over time. More recently, this assumption has begun to be relaxed in work on temporal recommendation [9, 10]. However, to the best of our knowledge no work has addressed the problem of temporal *reciprocal* recommendation, such as occurs in the context of online dating, employment websites, and other people-to-people interactions. Reciprocal recommender systems are necessary when the entity recommended to the active user (the one receiving the recommendation) must consent or reciprocate in some way to being the object of the recommendation; typically this "entity" is another user. Analysis of real-world data from a dating website on people-to-people interactions [2] shows that people's behaviour and activity levels do change over time, which leads to the conclusion that we need a dynamic model to generate better recommendations. To capture these temporal changes, in this paper we describe a Hidden Markov Model [8] reciprocal recommender system that captures how each user's behaviour evolves over time and generates recommendations accordingly.

In this research, the main concern is the changes in individuals' behaviours rather than changes in the whole population's behaviour (i.e. trends). Moreover, social networks (including dating websites) are different from typical item-user networks because there is no obvious categorisation of the users in these networks. For example, in article recommender systems, the problem can be abstracted to recommending an article category to minimise the number of classes that the model has to predict. While people can be categorised by age, gender, interests ...etc, minimising the recommendations problem in social networks to recommending a category instead of recommending a specific user does not lead to accurate results. Unlike the classic items recommenders, where the user can just purchase the recommended item, in this type of recommenders the recommendations are people, and these people also have to accept the other party for the interaction to be considered successful.

This paper proposes a Hidden Markov Model to generate recommendations in a people-to-people domain. The model was evaluated using a commercial dataset of a dating website and the results show a significant improvement in success rate. The following section explains how recommendations evolve over a sequence of events. Dataset and model descriptions are in section 3 followed by section 4 that discusses the evaluation of the experiments. Finally, section 5 summarises some of the related literature.

2. GENERATING RECOMMENDATIONS AS A SEQUENCE OF EVENTS

Although research on recommendation systems has been a popular research area in the past few years, there are still some sides to the research problem that have not been studied enough. One of these sides is the temporal aspect of recommendations. Most of the existing recommendations

RecSys'13, October 12–16, 2013, Hong Kong, China.

http://dx.doi.org/10.1145/2507157.2507214.

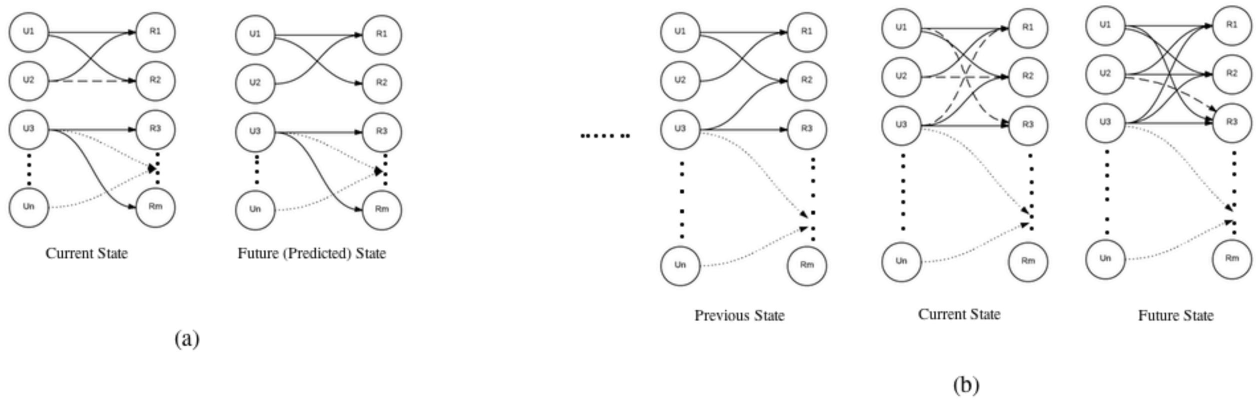

Figure 1: A representation of the recommendations problem as a graph. (a) shows the static representation and (b) shows the dynamic representation of the recommendation system where it is viewed as a graph evolving over time.

algorithms deal with the problem as a two-state problem, in which all historical data is considered one state and the recommendation problem becomes predicting the next state. If the problem was represented as a graph, where vertices are people or items and edges are links between vertices, then the recommendation problem can be minimised to predicting the new links that will appear in the future graph given the links in the present one (figure 1a).

In this paper, we created a model that considers the temporal aspect of the problem and utilises it to better personalise the recommendations. We track the changes of the graph over multiple time periods and observe how it evolves overtime then use this gained knowledge to predict the future graph. This can be extended to predict multiple versions of the graph in the future (i.e. instead of predicting the immediate future state, several successive future states can be predicted, see figure 1b).

3. EXPERIMENTAL DESIGN

3.1 Dataset

The dataset used to test the model is a real-world commercial dataset from a dating website. In the dating domain, there are users who initiate interactions, we call them *senders*, and people who receive interactions, we call them *recipients*. Senders and recipients can overlap which means a user can be a recipient and a sender at the same time. There are different forms of interactions that can be exchanged such as predefined messages, emails and chats. In this research, we use the predefined messages, we call them *messages*, to train and test our model because this is the first method of communication between users in most cases and depending on the success of these messages, users can further their communications and exchange other free text messages.

When a predefined message is sent, the recipient can ignore this message and not reply to it, reply with a positive predefined message or reply with a negative one. We have only considered messages that have replies to them and classify them as positive or negative interactions based on the reply message.

The dataset has over 2 million users and over 20 million interactions exchanged between these users. Therefore, using the whole dataset is not feasible and representative subsets have to be used instead. To generate training and testing data for our model, a time period was randomly selected (e.g. from March 1^{st} to March 15^{th}, 2009) and all active users during this time period were used as the experiment population. Then, for users in the selected population, all their interactions, even interactions outside the selected time period, were obtained and used to build the model. Several populations were generated and average results across these populations will be presented later in this paper.

The average population size is over 158,000 users of which a little over 14,000 were recipients and about 150,000 were senders. These users exchanged just over a million message between them. Each population was divided to 70% training data and 30% testing data.

This dataset was chosen because it is a real-world commercial reciprocal dataset that has temporal dynamics. Although users' life cycles are mostly short in a dating website, there are several lifecycle phases to capture and these changes between phases have their effects on the decision of initiating an interaction and the decision of accepting one.

3.2 Recommendation Using Hidden Markov Models

To capture the concept of recommendations as a sequence of events, we represent each interaction I_k in the model as a sequence of size n as follows:
$I_k = (O_{k-n}, O_{k-n+1}, O_{k-n+2}, ..., O_k)$ if $k \geq n$

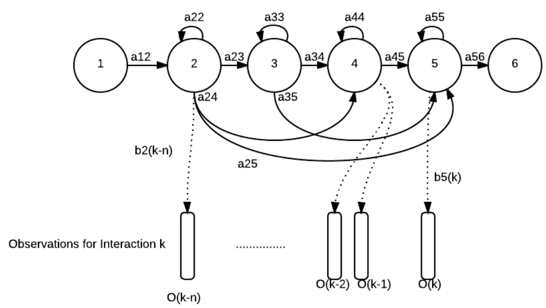

Figure 2: A Hidden Markov Model for people-to-people recommender.

$I_k = (\phi, \phi, ..., O_{k-2}, O_{k-1}, O_k)$ if $k < n$
Where O_k is the k_{th} observation vector. Each observation vector consists of: (1) a selected set of profile data for the sender and the recipient, (2) a set of derived data such as ages of the sender and the recipient and physical distance between them, (3) a set of temporal data such as activity in the past 7 days and activity in the past 28 days. Each interaction is then classified to a successful or a failed interaction. An HMM is then trained using the training data which consists of interactions and their classes and the resulted model is used to predict new interactions. The interactions that are predicted to be successful comprise the generated recommendations. The proposed HMM is shown in Figure 2. We used the Hidden Markov Models Toolkit (HTK) to implement this model.

The model in the top layer is the hidden model that represents the user's behaviour to interactions and the bottom layer shows the observable sequence of interactions. The variable a_{ij} is the transition probability from state i to state j, $b_i(j)$ is the probability of seeing observation vector j in state i and $O(k)$ is the k_{th} observation vector. HMM was the model of choice because it has been reported to work successfully in capturing temporal changes [9] and it has been used in several domains with positive results.

Another factor to consider was to decide whether to implement each observation as a time period (i.e. interactions in a week) or as an event (i.e. a message sent/received). The latter was chosen because a day in an active user's life cycle can be equal to a week or more in a the life cycle of a less active user. This serves as a method of normalisation over the whole population.

Finally, the sequence is built from the perspective of the recipient users because in a reciprocal recommender, the recipient is the one who has to make the final decision.

4. EVALUATION

The proposed model in section 3.2 was evaluated using a dataset obtained from a commercial dating website. Users of the website did not have access to recommendations generated by our model and therefore the model was evaluated using historical data.

4.1 Evaluation Metrics

Two metrics were used to evaluate our model. The first one is the *Success Rate* which is defined as the proportion of generated recommendations that are correct. The other metric used to evaluate the model is *Recall* which is the proportion of successful interactions in the testing set that were predicted by the model. Success rate is equivalent to the standard metric of Precision.

More formally, let R be the set of recommendations generated by the model, R_+ be the subset of R that are correct, I be the set of successful interactions in the testing data and $Size(S)$ is the size of a set S.

$$SuccessRate = \frac{Size(R_+)}{Size(R)}$$

$$Recall = \frac{Size(R_+)}{Size(I)}$$

4.2 Results and Discussion

After training the model with training data, we evaluated the model by asking it to predict whether an interaction will be successful or not. The testing data was different from the training data and it comprise about 30% of the size of the dataset used for the experiment. We have tried different values of n to decide on the one that gives the best results. The predicting accuracy of the system was around 76% but this is for predicting an interaction to be a success or a failure.

The interactions that were predicted to be successful are considered to be the generated recommendations. The success rate in generating these recommendations was 47.1% and recall was 4.9%. The success rate of other collaborative filtering recommendation systems that used the same dataset such as [6] was about 38% with recall 3.6%. Additionally, other content-based recommendation systems that used the same dataset such as [4] have a success rate of about 30% and a recall of about 4%.

In the domain of social networks with the commercial dataset that we have used, it is very difficult to get an improvement in recommendations' success rate and an improvement of about 9% is considered very valuable and significant. Moreover, since the model is content-based, this means that it performs well with new users including users who have no or few interactions [1]. While the results of testing the model on new users who were not included in the training data (table 2) are positive, more specific tests need to be performed to discover the exact figures for each subgroup in the testing population. More precisely, we need to discover what is the success rate of recommendations for users who have not made any interactions yet, for users who have made a few interactions and for people who have made enough number of interactions. More detailed results are shown in table 1 and table 2.

5. RELATED WORK

Although recommender systems have been investigated thoroughly in the literature, there has been little research on temporal aspects of the recommendations problem. Few algorithms [5, 9] were designed recently with temporal aspects and they showed encouraging results. However, all of these algorithms were designed for people-to-item domains.

Another area that has not been researched intensively is the people-to-people recommendation systems. Recent

Table 1: **Summary of results using different n values.**

n	Success Rate	Recall
2	0.482	0.034
3	0.441	0.043
4	0.436	0.045
5	0.447	0.044
6	0.471	0.049
7	0.466	0.033
8	0.459	0.067

Table 2: Results when testing the model with new users that were not included in the test population.

Number of Users	38000
Number of Interactions	81000
Value of n	6
Success Rate	0.466
Recall	0.169

works [4, 6, 3, 7] on such a domain reported significant improvement in people-to-people recommenders. However, the temporal changes in users' preferences and behaviours are not emphasised in these models due the use of static representation of the recommendation problem.

6. CONCLUSION AND FUTURE WORK

In this paper we presented a model for people-to-people recommendations using HMM that can capture the temporal changes of users' behaviours and generates better personalised recommendations based on this. Evaluating this model using a commercial dataset for a dating website shows a significant improvement in the success rate of recommendations.

The model is built using content-based data only and we believe that extending the model to use some collaborative filtering techniques can improve it. For example, a temporal matrix factorization approach to collaborative filtering [10] would be interesting to extend to reciprocal recommendation. In addition, more testing is required to investigate the success rate and recall for users with no or few interactions. Finally, a method of ranking the generated recommendations and selecting the top-N can improve the success rate.

Acknowledgments

We would like to thank Smart Services Cooperative Research Centre and their industrial partners for providing the datasets.

7. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. 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, June 2005.

[2] A. Alanazi and M. Bain. Ranking interaction-based collaborative filtering recommendations using temporal features in online dating. In *Proceedings of the 18th IBIMA Conference*, May 2012.

[3] Y. S. 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*, pages 247–260. Springer, 2010.

[4] Y. S. Kim, A. Mahidadia, P. Compton, A. Krzywicki, W. Wobcke, X. Cai, and M. Bain. People-to-people recommendation using multiple compatible subgroups. In *AI 2012: Advances in Artificial Intelligence*, pages 61–72. Springer, 2012.

[5] Y. Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4):89–97, April 2010.

[6] A. Krzywicki, W. Wobcke, X. Cai, A. Mahidadia, M. Bain, P. Compton, and Y. Kim. Interaction-based collaborative filtering methods for recommendation in online dating. In L. Chen, P. Triantafillou, and T. Suel, editors, *Web Information Systems Engineering, AI WISE 2010*, volume 6488 of *Lecture Notes in Computer Science*, pages 342–356. Springer Berlin / Heidelberg, 2010.

[7] L. Pizzato, T. Rej, T. Chung, I. Koprinska, and J. Kay. Recon: a reciprocal recommender for online dating. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 207–214. ACM, 2010.

[8] L. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of IEEE*, 77(2):257–286, 1989.

[9] N. Sahoo, P. Vir Singh, and T. Mukhopadhyay. A hidden markov model for collaborative filtering. *MIS Quarterly-Management Information Systems*, 36(4):1329, 2012.

[10] J. Sun, K. Varshney, and K. Subbian. Dynamic matrix factorization: A state space approach. In *Proc. 2012 IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1897–1900. IEEE, 2012.