Integrating Social Information into Collaborative Filtering for Celebrities Recommendation

Qingwen Liu, Yan Xiong, and Wenchao Huang

School of Computer Science of University of Science and Technology of China
mrhead@mail.ustc.edu.cn, {yxiong,huangwc}@ustc.edu.cn

Abstract. With the exponential growth of users' population and volumes of content in micro-blog web sites, people suffer from information overload problem more and more seriously. Recommendation system is an effective way to address this issue. In this paper, we studied celebrities recommendation in micro-blog services to better guide users to follow celebrities according to their interests. First we improved the jaccard similarity measure by significant weighting to enhance neighbor selection in collaborative filtering. Second, we integrated users' social information into the similarity model to ease the cold start problem. Third we increased the density of the rating matrix by predicting the missing ratings to ease the data sparsity problem. Experiment results show that our algorithm improves the recommendation quality significantly.

Keywords: micro-blog, collaborative filtering, user similarity model, data sparsity.

1 Introduction

As the age of Web 2.0 comes, social media becomes more and more popular. Recently, the micro-blog web sites have shown a great charm, with millions of users joining in it. The micro-blog web sites fundamentally provide a public platform for their users to seek and share information, to communicate with others, and to build online friendships. It can be seen as a hybrid of email, instant messaging and news broadcasting systems. Unlike other social networks like Facebook or LinkedIn, the following relationship between users in micro-blog system is not necessarily reciprocal. For this reason, people can follow anyone they like without requiring acceptance. Building friendship in this way lowers down the cost of expanding one's network and allows some users to be followed by many users without following many themselves, effectively becoming celebrities or stars [1].

In the view of the exponential growth of micro-blog user population and volumes of content generated by them, it gets difficult for users to choose whom to follow and what to read. Users may easily be flooded with information streams. Personalized recommendation is an important way to address this issue and it has been well studied by both academia and industry recently. In this paper, we will study the problem of recommending celebrities or stars in micro-blog

A. Selamat et al. (Eds.): ACIIDS 2013, Part II, LNAI 7803, pp. 109–118, 2013.

services. We are motivated by the rich social information to provide potential evidences for users' similarity computation and missing ratings prediction in collaborative filtering (CF). And we prosed a novel collaborative filtering framework which improve jaccard similarity measure by significant weighting and integrate social information to ease the data sparsity problem and enhance user similarity modeling.

The rest of the paper is organized as follows. Section 2 covers related works on collaborative filtering and social recommending systems. We describe a novel approach which integrates social information into collaborative filtering to address the cold-start problem and the data sparsity problem in Sec. 3. Evaluation metric and experiment results are demonstrated in Sec. 4. Finally, we conclude in Sec. 5.

2 Related Work

We will review related works from 2 different research areas: CF algorithms and the role of social features played in recommendation systems.

The fundamental assumption of CF is that if two users have rated some items similarly, or they have similar behaviors (e.g. watching, buying, listening), and hence they will rate or act on other items similarly [2]. One of the biggest challenges in CF is the data sparsity problem, which leads to the failure of finding similar users or items. The density of available ratings in commercial recommending systems is often less than 1% [3] and the density of our data set is 0.64%. Many algorithms have been proposed to overcome the data sparsity problem. In [4], a dimensionality reduction technique, Singular Value Decomposition (SVD), is employed to remove unrepresentative or insignificant users or items and map the rating space into a lower dimensional semantic space. However, some information about users or items may be discarded by SVD, thus resulting a decrease in the recommendation quality. P.Melville et al. proposed a hybrid model named content-boosted CF to address the data sparsity problem, in which external content information was used to produce predictions for new users and new items [5]. The result of this method is promising, in our paper rich social information is extracted from micro-blog web sites to enhance collaborative filtering. H.Ma et al. increased the density of the rating matrix by predicting the missing ratings using a user-based and item-based combined model [6]. No using of external information in this method will limit the recommendation quality, thus we propose to integrate the social information to address this issue.

To understand micro-blog usage, Akshay et al. showed how users with similar intentions connect with one another by analyzing the user intentions associated at a community level in [7]. The findings motivated us to use social information to discover similar users for CF based recommendation. In [8], different content-based recommending systems are built by using different types of social information to recommend URLs extracted from micro-blog content. Ido Guy et al. measured user similarity based on social features from two aspects: users' social network structure and users' content information [9]. The results of these two papers both show that adding social features into traditional recommendation algorithms can significantly improve accuracy. Daly systematically studied how to measure the network effects of recommending social connections and how different social recommending algorithms differ [11]. His findings guide us to choose appropriate types of social information for celebrities recommendation. Chen et al. claimed in [10] that content information is more effective than other kinds of social information in people recommendation. Finally, Hannon evaluated a range of different user profiling and recommendation strategies in [12]. It found that a mixture profiling strategy which use both contents and social connections can produce better received recommendations.

3 Recommender System Description

3.1 Problem Definition

Formally, we will formulate our problems as follows. Let $\mathcal{U}$ be a user set and let $\mathcal{C}$ be a celebrity set in micro-blog web sites. The following relationships between users and celebrities are denoted by a $|\mathcal{U}| \times |\mathcal{C}|$ matrix, called user-item rating matrix. Every entry $r_{ui} \in \{-1, 0, 1\}$ represents the value that user $u \in \mathcal{U}$ rated item $i \in \mathcal{C}$ where 1 means user u followed item i, -1 means user u refused to follow item i and 0 means the user has not rate the item yet. Given a user u and an item i, let $\mathrm{P}(u, i)$ be a recommending function that measures the preference of user u on item i, i.e. $\mathrm{P} \in \{\mathcal{U} \times \mathcal{C} \to \mathbb{R}\}$. Then given a user u and an item list $\mathcal{L}$, we will rank the items in $\mathcal{L}$ according to $\mathrm{P}(u, i)$ and select top N items as the recommending items for u. More formally:

$$\forall u \in \mathcal{U} \quad \mathcal{S}_u = \underset{i \in \mathcal{L}}{\mathrm{argTopN}} \quad \mathrm{P}(u, i) \tag{1}$$

where $\mathcal{S}_u$ is the recommendation result.

3.2 Significant Weighting for Jaccard Similarity Measure

User similarity computation is a critical step for collaborative filtering algorithms. We claim that jaccard similarity is a more natural way to model similarity between two binary rating vectors than other similarity measures by its definition. In our problem, we define jaccard similarity of two rating vectors as (2).

$$sim(v_1, v_2) = \frac{\sum_{i=1}^{n} 1\{r_{1i} = r_{2i} \wedge r_{1i} \neq 0\}}{\sum_{i=1}^{n} 1\{r_{1i} \neq 0 \wedge r_{2i} \neq 0\}} \tag{2}$$

where $1\{*\}$ is an indicator function: $\{true, false\} \to \{1, 0\}$ and r_{1i} or r_{2i} is the ith rating of vector v_1 or v_2.

This definition has two disadvantages. First, positive ratings are much more informative than negative ratings in our problem, but (2) treats both identically.

Second, (2) will overestimate the similarity of users who happen to rate quite a few items identically but who may not have similar overall preference. The estimation is not reliable since too few co-ratings have no statistical significance. To address the first problem, we give different weights to positive ratings and negative ratings by using the following equation:

$$\mathrm{sim}^{'}(v_1, v_2) = \frac{\sum_{i=1}^{n} 1\{r_{1i} = r_{2i} = 1\} + \lambda \sum_{i=1}^{n} 1\{r_{1i} = r_{2i} = -1\}}{\sum_{i=1}^{n} 1\{r_{1i} \neq 0 \wedge r_{2i} \neq 0\}} \tag{3}$$

where λ is a parameter between 0 and 1. To address the second problem, we follow the intuition that computing without enough supporting evidence (co-ratings) should be punished. Thus, a penalty function was introduced by (4).

$$\mathrm{pun}(v_1, v_2) = \frac{\min(\sum_{i=1}^{n} 1\{r_{1i} \neq 0 \wedge r_{2i} \neq 0\}, \phi)}{\phi} \tag{4}$$

where $\sum_{i=1}^{n} 1\{r_{1i} \neq 0 \wedge r_{2i} \neq 0\}$ means the number of items rated in common and ϕ is a threshold which is greater than 1. By applying this penalty function, we get the new similarity measure in (5).

$$\mathrm{jaccard\_sim}(v_1, v_2) = \mathrm{sim}^{'}(v_1, v_2) \times \mathrm{pun}(v_1, v_2) \tag{5}$$

Equation (5) will devalue the similarity of v_1,v_2 if the number of co-rated items are smaller than ϕ and give different weights to positive and negative ratings.

3.3 Social Information Integrating for Neighbor Selection

Given a user, his/her neighbor set is composed of two parts. One is computed by jaccard similarity based on the rating matrix and the other is computed by social information. In micro-blog web sites, the social information can be classified into 3 types, which are content of posts, social connections and social activities. Accordingly, we will model users' social similarity from three aspects by (6).

$$\mathrm{social\_sim}(u, v) = \alpha * \mathrm{sim}_{\mathrm{C}}(u, v) + \beta * \mathrm{sim}_{\mathrm{n}}(u, v) + \gamma * \mathrm{sim}_{\mathrm{a}}(u, v) \tag{6}$$

where α, β, γ are three parameters which determine the weights of different types of similarity. Following the approach in [8], we build a profile vector for each user from the words that were included in their posts. Each entry of the profile vector is weighted by the term-frequency inverse-user-frequency (TF-IUF) of the corresponding word. $\mathrm{sim}_{\mathrm{C}}(u, v)$ is then computed as the cosine similarity between their profile vectors.

$$\mathrm{sim}_{\mathrm{C}}(u, v) = \frac{\sum_{i \in W} w_{ui} w_{vi}}{\sqrt{\sum_{i \in W} {w_{ui}}^2}\sqrt{\sum_{i \in W} {w_{vi}}^2}} \tag{7}$$

where W is the set of words which are extracted from users' posts and w_{ui} is the weight of the ith word of user u.

In practice, if a user u follows another user v in micro-blog web sites, user u may be interested in user v as an information seeker or they might be friends in the real world. Motivated by this, we model the social connection similarity by a binary function as (8).

$$\mathrm{sim_n}(u,v) = \begin{cases} 1 & \text{u follows v} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

Users have three types of social activities, mention, repost and comment. Given a user u, let $\mathcal{A}_u$ represent the set of users who has ever been interact with u in micro-blog web sites. Intuitively, more activities imply more intimate relationship and more intimate relationship implies more similar interests between users. According to this, we model the action similarity by (9).

$$\mathrm{sim_a}(u,v) = \frac{\#\mathrm{men}_{uv} + \#\mathrm{rep}_{uv} + \#\mathrm{com}_{uv}}{\max\limits_{v' \in \mathcal{A}_u} (\#\mathrm{men}_{uv'} + \#\mathrm{rep}_{uv'} + \#\mathrm{com}_{uv'})} \tag{9}$$

where $\#\mathrm{men}_{uv}$ is the number of times that user u mentioned user v, $\#\mathrm{rep}_{uv}$, $\#\mathrm{com}_{uv}$ is the number of times that user u reposted and commented on user v's posts.

Once the user similarity is modeled, we can select the neighbor set for users. H.Ma argued in [6] that the top-N neighbor selection method is misleading when a user actually has few neighbors and that selecting the ones whose similarity is greater than some threshold as neighbors results in more accurate recommendations. In our problem, for every user u, we generate two neighbor sets of u according to (10)(11).

$$\mathrm{T}_u = \{v | v \in \mathcal{U} \wedge \mathrm{jaccard\_sim}(u,v) > \varphi\} \tag{10}$$

$$\mathrm{S}_u = \{v | v \in \mathcal{U} \wedge \mathrm{social\_sim}(u,v) > 0\} \tag{11}$$

where S_u the neighbor set that is computed based on social information, T_u is the neighbor set that is computed based on the rating matrix and φ is a threshold between 0 and 1.

Now we have demonstrated the two methods for neighbor selection. To take advantage of both methods, we first make predictions using T_u and S_u respectively, and then combines the predictions linearly by (12).

$$\begin{aligned} \mathrm{P}(u,i) = \bar{r_u} + \theta \cdot \frac{\sum\limits_{v \in \mathrm{S}_u} (r_{vi} - \bar{r_v}) \cdot \mathrm{social\_sim}(u,v)}{\sum\limits_{v \in \mathrm{S}_u} \mathrm{social\_sim}(u,v)} + \\ (1-\theta) \cdot \frac{\sum\limits_{v \in \mathrm{T}_u} (r_{vi} - \bar{r_v}) \cdot \mathrm{jaccard\_sim}(u,v)}{\sum\limits_{v \in \mathrm{T}_u} \mathrm{jaccard\_sim}(u,v)} \end{aligned} \tag{12}$$

3.4 Effective Missing Ratings Prediction

Addressing the data sparsity problem is one of the most critical issues in collaborative filtering. A lot of methods have been proposed to deal with this problem as mentioned in Sec. 2. Missing ratings prediction is an intuitive, simple and effective way to increase the density of the rating matrix. A model which chooses to predict the missing ratings or not according to confidence is proposed by H.Ma in [6]. Significant improvement has been seen in this model. However, it only iterates the original rating matrix to produce a denser one without accessing external information, which will limit the predicting quality. To this end, we integrate social information to provide evidence for missing ratings prediction in our model. As illustrated in Sec. 3.3, we generate two neighbor sets from social information and rating information. Then, for a missing rating r_{ui}, we use the two neighbor sets to predict missing ratings by equation (12). In our problem the rating mode is binary, so the result produced by equation (12) can be viewed as the confidence for positive or negative ratings. And at last, we determine the prediction of r_{ui} by a parameter ζ as (13).

$$r_{ui} = \begin{cases} 1 & \mathrm{P}(u,i) > \zeta \\ -1 & \mathrm{P}(u,i) < -\zeta \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

where 1 represents a positive rating, -1 represents a negative rating and 0 represents a missing rating.

4 Evaluation Metric and Experimental Analysis

4.1 Evaluation Metric and Data Set

We are most interested in a system that can recommend items in a ranked list where the most user-interested items take top positions rather than a method that accurately predicts the numeric rating of every item. To analyze this, we use the predicted score to rank the recommended items, and apply the Mean Average Precision at N i.e. MAP@N to measure the recommendation quality. We evaluate our algorithm in the data set provided by Tencent Inc. for KDD CUP 2012, which represents a sampled snapshot of Tencent Weibo users' preferences for various items. The user-item rating matrix in this data set contains 42118498 distinct binary ratings rated by 1392873 users on 4710 items. The density of the matrix is 0.64%. In addition to the rating matrix, the data set contains rich social information about users and items. We divided the ratings into two parts: the ratings made before 22:16:00 5th November 2011 as training data, and the rest ratings as testing data. To set up the experiments effectively, we sampled 6000 users and their ratings randomly from the training data and built three training sets containing 1000, 2000, 3000 users respectively. Then we sampled 200 testing users and their ratings from the testing data accordingly.

4.2 Experiments and Analysis

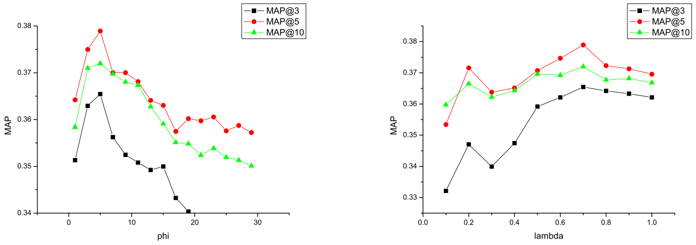
We have described how to improve jaccard similarity measure to fit our problem, how to integrate social information into neighbor selection process and how to predict the missing rating to make the rating matrix denser to improve recommendation quality. Accordingly, we will conduct several experiments to answer the following questions:

1. Does the improved jaccard similarity measure help to improve prediction accuracy? If it does, what is the effect of the parameter λ and ϕ?
2. Does the social similarity model help to improve prediction accuracy? If it does, how do the jaccard similarity model and social similarity model benefit each other?
3. We implemented $3 \times 2 = 6$ algorithms from the following two dimensions:
   (a) neighbor selection
      i. use social information only
      ii. use rating matrix only
      iii. combine both
   (b) missing ratings prediction
      i. predict missing ratings
      ii. not predict missing ratings

   Among all the 6 algorithms, which one performs the best?

Question 1. To answer Question 1, we build a model which does not incorporate the social information and the missing rating predicting process for clarity. First, we set ϕ to 5, and vary the range of λ from 0 to 1 with a step value of 0.1. Then we plot the MAP-λ curve to show the impact of λ. Fig. 1 shows how λ affects MAP@3, MAP@5, MAP@10 respectively. Setting λ to 1.0 means equally weighting positive and negative ratings and decreasing λ means reducing the influence of negative ratings. As we see in Fig. 1, MAP increases as we reduce λ from 1.0 to 0.7, which implies that reducing the influence of negative ratings does help to increase the recommendation accuracy. If we continue to reduce λ to 0, MAP decreases. So, we get the best performance when $\lambda = 0.7$ on our experiment data set.

To show the effects of ϕ, we set λ to 0.7, and vary the range of ϕ from 1 to 29 with a step value of 2. Then we plot the MAP-ϕ curve to show the impact of ϕ. Fig. 1 shows how ϕ affects MAP@3, MAP@5, MAP@10 respectively. The purpose of introducing ϕ is to devalue the similarity of users who have too few co-ratings and make the similarity computation more sensible. The larger the value of ϕ, the similarity of the users who have few co-ratings will be devalued more seriously. Setting the value of ϕ to 1 means computing user similarity normally. As we see in Fig. 1, MAP increases as we increase the value of ϕ from 1 to 5, which implies that introducing the penalty function to similarity computation does help to improve recommendation quality. If we continue to increase the value of ϕ to 29, we can see that MAP decreases on the overall trend. Thus, we get the best performance when $\phi = 5$ on our experiment data set.



(a) Impact of ϕ (b) Impact of λ

Fig. 1. Impact of significant weighting

Question 2. To answer Question 2, we combine the two neighbor selection methods to make predictions. For clarity, we remove the missing rating prediction step. Parameter θ balances the effect of social features and the effect of ratings. It takes advantages of these two neighbor selection methods. If $\theta = 0$, we only use the rating matrix to compute neighbor set for users, and if $\theta = 1$, we only use the social features to compute neighbor set for users. In other cases, we combine the predictions based on the two neighbor sets to get the final predictions. To show how the two neighbor selection methods benefit each other, we first set λ to 0.7 and set ϕ to 5 respectively, and then vary the range of θ from 0 to 1 with a step value of 0.1 and plot MAP-θ curve.

Observed from Fig. 2, we draw the conclusion that combination of the two neighbor selection methods does help to improve prediction accuracy significantly. Figure 2 shows that as the value of θ increases from 0 to 0.3, MAP increases. As the value of θ continues to increase, MAP decreases on overall trend. We get the best performance at $\theta = 0.3$, which may indicate that the rating information is more important than social information.

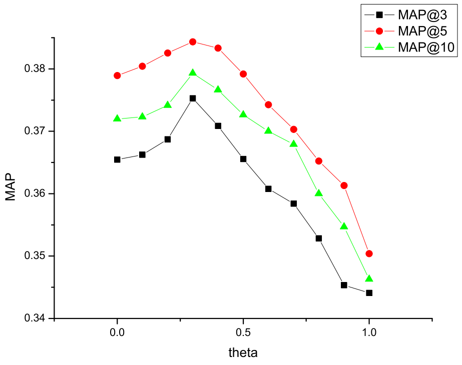


Fig. 2. Impact of θ

Question 3. To answer Question 3, we build a model that makes predictions using neighbor selection, missing rating prediction combined as a single factor. Thus, we can compare all the 6 algorithms individually side by side. In these algorithms, we set the parameters to the best values according to the previous experiment results, i.e. $\lambda = 0.7, \phi = 5, \alpha = 0.2, \beta = 0.2, \gamma = 0.6, \varphi = 0.1$, $\theta = 0.3$ and $\zeta = 0.5$ (the tuning process of $\alpha, \beta, \gamma, \varphi$ and ζ is not included in this paper due to space limitation). Table 1 illustrates the performance of the 6 algorithms. The result suggests that the algorithm integrating social information for neighbor selection and predicting missing ratings outperform other algorithms.

Table 1. Comparison of Algorithms

		Predict missing	Not predict missing
MAP@3	SNS	0.349	0.344
	REG	0.373	0.365
	COM	**0.387**	0.375
MAP@5	SNS	0.354	0.350
	REG	0.384	0.379
	COM	**0.395**	0.384
MAP@10	SNS	0.347	0.346
	REG	0.378	0.372
	COM	**0.392**	0.379

5 Conclusion and Future Work

In this paper, we studied the celebrities or stars recommendation problem on micro-blog web sites. First, we improved jaccard similarity by significant weighting to make the similarity measure more reasonable. Second, we integrated social information for neighbor selection. Third, we predicted missing ratings to enhance collaborative filtering. The experiment results showed that our approach improves the recommendation quality significantly. We claim that our recommendation framework is easy to be generalized to fit other collaborative filtering problems, which are provided with external information about users. However, domain-specific properties may have great impact on the effectiveness of the algorithms and more specific user similarity models need to be developed.

Further study may explore more social features to deepen our understanding on user similarity modeling. For example, we may use the sequential information such as time stamp of ratings to make session analysis to find similar patterns for users as the evidence for similarity computation. In addition to the users' social information, items' social information is valuable to leverage to enhance item based collaborative filtering.

Acknowledgement. We would like to acknowledge Tencent Inc. for providing the exciting and comprehensive data set. This work was supported in part by the National Natural Science Foundation of China (No.61170233, No.61232018, No.61272472, No.61202404, No.61272317)and China Postdoctoral Science Foundation (No.2011M501060).

References

1. Brzozowski, M.J., Romero, D.M.: Who Should I Follow? Recommending People in Directed Social Networks. In: AAAI Conference on Weblogs and Social Media (2011)
2. Goldberg, K., Roeder, T., Gupta, D., Perkins, C.: Eigentaste: a constant time collaborative filtering algorithm. Information Retrieval 4(2), 133–151 (2001); Advances in Artificial Intelligence, 1–20 (2009)
3. Sarwar, B., Karypis, G., Konstan, J., Reidl, J.: Item based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web, pp. 285–295 (2001)
4. Billsus, D., Pazzani, M.: Learning collaborative information filters. In: Proceedings of the 15th International Conference on Machine Learning (1998)
5. Melville, P., Mooney, R.J., Nagarajan, R.: Contentboosted collaborative filtering for improved recommendations. In: Proceedings of the 18th National Conference on Artificial Intelligence, pp. 187–192 (2002)
6. Ma, H., King, I., Lyu, M.R.: Effective missing data prediction for collaborative filtering. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 39–46 (2007)
7. Java, A., Song, X., Finin, T., Tseng, B.: Why we twitter: understanding microblogging usage and communities. In: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis (2007)
8. Chen, J., Nairn, R., Nelson, L., Bernstein, M., Chi, E.H.: Short and Tweet: Experiments on Recommending Content from Information Streams. In: Proceedings of the 28th Conference on Human Factors in Computing Systems, pp. 1185–1194 (2010)
9. Guy, I., Zwerdling, N., Carmel, D., Ronen, I., Uziel, E., Yogev, S., Ofek-Koifman, S.: Personalized recommendation of social software items based on social relations. In: Proceedings of the 3rd Conference of Recommender Systems, pp. 53–60 (2009)
10. Chen, J., Geyer, W., Dugan, C., Muller, M., Guy, I.: Make new friends, but keep the old - Recommending people on social networking sites. In: Proceedings of the 27th Conference on Human Factors in Computing Systems, pp. 201–210 (2009)
11. Daly, E.M., Geyer, W., Millen, D.R.: The network effects of recommending social connections. In: Proceedings of the 4th ACM Conference on Recommender Systems (2010)
12. Hannon, J., Bennett, M., Smyth, B.: Recommending twitter users to follow using content and collaborative filtering approaches. In: Proceedings of the 4th ACM Conference on Recommender Systems (2010)