Tags Meet Ratings: Improving Collaborative Filtering with Tag-Based Neighborhood Method

Zhe Wang
National Engineering Research Center of Fundamental Software, Institute of Software, Chinese Academy of Sciences
Graduate University of Chinese Academy of Sciences
wangzhe07@iscas.ac.cn

Yongji Wang
National Engineering Research Center of Fundamental Software, Institute of Software, Chinese Academy of Sciences
ywang@itechs.iscas.ac.cn

Hu Wu
National Engineering Research Center of Fundamental Software, Institute of Software, Chinese Academy of Sciences
wuhu@itechs.iscas.ac.cn

ABSTRACT

Collaborative filtering (CF) is a method for personalized recommendation. The sparsity of rating data seriously impairs the quality of CF's recommendation. Meanwhile, there is more and more tag information generated by online users that implies their preferences. Exploiting these tag data is a promising means to alleviate the sparsity problem. Although the intention is straight-forward, there's no existed solution that makes full use of tags to improve the recommendation quality of traditional rating-based collaborative filtering approaches. In this paper, we propose a novel approach to fuse a tag-based neighborhood method into the traditional rating-based CF. Tag-based neighborhood method is employed to find similar users and items. These neighborhood information helps the sequent CF procedure produce higher quality recommendations. The experiments show that our approach outperforms the state-of-the-art ones.

ACM Classification Keywords

H.3.3 Information Storage and Retrieval: Information Search and Retrieval-Information filtering

General Terms

Algorithms, Experimentation

Author Keywords

Tags, Latent Dirichlet Allocation (LDA), Collaborative Filtering, Neighborhood Method

INTRODUCTION

Workshop SRS'10, February 7, 2010 Hong Kong, China

Nowadays people are inundated by choices. Personalized recommendation is a solution to this problem. Various kinds of recommender systems are employed for better user experience. Collaborative filtering [4, 12] is one of the best techniques of choice therein. This technique tries to identify users that have relevant interests by calculating similarities among user profiles. The idea is that it may be of benefit to one's search for information to consult the behavior of other users who share the same or relevant interests.

Because collaborative filtering recommendation depends on the preference of the users with the same or relevant interests, the similarity computation imposes significant influence on the quality of recommendation. Early item-based and user-based collaborative filtering approaches find similar users or items (neighbors) by calculating Pearson correlation coefficient [23]. These approaches are efficient and effective. But simply comparing the rating records of different users or items cannot help to find the best neighbors. If a user has few ratings for items or this user only gives all his/her ratings to the unpopular ones, it will be difficult for those approaches to find the proper neighbors.

Recently, matrix factorization approaches earn more popularity because of their higher recommendation quality and smaller online costs. One of the most significant differences from early approaches is that they extract the "features" of the users and the items. By this way, they decompose the original preference matrix into several low rank approximates [15]. For the items, every feature reflects the preference by a group of similar users. For the users, every feature reflects their preference for a collection of similar items. By virtue of extracting users' and items' features, matrix factorization approaches are able to find better neighbors and hence produce better recommendations.

Despite the merits mentioned before, the existing matrix factorization approaches [6, 7, 8, 16, 26] fail to ex-

tract sufficient feature information, which reflects the problem of data sparsity. It is because they fit the original matrix by feature extraction only based on the rating data while the rating data are extremely sparse. If we could obtain more ratings, we would surely enhance the quality of fitting process. From this standing point, we propose a better collaborative filtering approach to exploit additional knowledge from the tags as a supplement to ratings.

Tags are simple, ad-hoc labels assigned by users to describe or annotate any kind of resource for future retrieval. Their flexibility means they probably capture a user's perspective and preference with ease. Most recent work focuses on the tag recommendation in which the objects to recommend are tags [18, 20, 22, 27]. In the case of item-based recommendation, users expect to get specific suggestion on which item might be interesting. There are a limited number of solutions for this situation, and most of them do not have a generalized adaptation to different data resources because they ignore abundant rating data [11, 25]. In this paper, we offer a novel personalized recommendation method which matches the case of containing both ratings and tags.

Our approach still shares the main idea of classic neighborhood method, but there are some differences in where to find neighbors. The neighbors are usually found in the ratings for the traditional CF approach [1]. We do not find neighbors directly by this means. First we exploit the latent topic grouping information hidden in tags and then we find groups of the users interested in similar topics and collections of the items under similar topics. To predict the user's rating for the item, we consult the ratings of both of the user's and the item's neighbors by employing a neighborhood method. Thanks to taking into account both tag neighbors and rating neighbors, our method outperforms most popular CF approaches.

The structure of the rest of the paper is as follows. In Section 2 we introduce the background and the related works. In section 3 we explain our improved collaborative filtering method in details. In Section 4 we give two toy examples and compare our method with NMF, PMF and SVD on a popular movie dataset. And finally we conclude this paper with some future work.

PRELIMINARIES

Rating prediction is one of the most popular means to evaluate the performance of collaborative filtering algorithms. From the rating data of most collaborative filtering datasets, we can obtain a $N \times M$ rating matrix R including N users and M items. Matrix R is defined as

$$r_{ij} = \begin{cases} user_i\text{'s rating for } item_j\,, & \text{if } user_i \text{ has rated } item_j \\ 0\,, & \text{otherwise}\,, \end{cases}$$

where $i \in \mathbb{N}^+, j \in \mathbb{M}^+$ and $r_{ij} \in [1, R_{max}]$. The usual evaluation process is the hold-out cross validation[5]. A certain proportion of ratings are hidden for testing and the rest are used for training. The measures of evaluation include complexity and accuracy. Nevertheless, the accuracy is much more important because most of the Collaborative Filtering approaches are offline. Therefore, it is the focus in this paper.

Naive Estimates

One of the most instinctive predicting methods is to compute the mean values. Taking the user's and the item's average biases involved, we get the naive estimate [8]:

$$b_{ij} = \mu + b_i + b_j\,, \tag{1}$$

where b_{ij} indicate the predicted rating of $user_i$ on $item_j$; μ is the global average rating; b_i and b_j denote $user_i$'s and $item_j$'s average bias, respectively.

This naive method is effective and scalable, but it does not take the interaction between users into account. Every user's rating for a item has influences on other users' opinions to that item. This interdependence between the users forms a social network [24] which connects all users together. The personalized recommendations are not delivered in isolation, but in the context of this social network [14]. The neighborhood method is one of the most effective methods to analyze the context.

Neighborhood Method

The aim of the neighborhood method [2] is to find the users who give similar ratings and the items which receive similar ratings. The approximate ratings infer the potential similarity of the future ratings. This is the basic assumption of collaborative filtering. Because the neighborhood method digs out from the neighbors the clues that indicate the potential ratings, it produces better predictions than the naive estimate. The model of the neighborhood method unifying item-based and user-based collaborative filtering approaches is

$$\hat{r}_{ij} = b_{ij} + \sum_{h \in S^k(j;i)} \theta^i_{hj}(r_{ih} - b_{ih}) + \sum_{h \in S^k(i;j)} \theta^j_{ih}(r_{hj} - b_{hj})\,, \tag{2}$$

where $\hat{r}_{ij}$ is the predicted rating; b_{ij} refers to the naive estimate's prediction; $S^k(j;i)$ denotes the set including k nearest rated items neighboring with $item_j$ for a given $user_i$ and r_{hj}; $S^k(i;j)$ denotes the set including k nearest users neighboring with $user_i$ for a given $item_j$ and r_{ih}; θ reflects the different weights of r_{ih}. There are several representations for the weights. The cosine similarity is one the most effective measures to indicate the different weights.

$$Cosine\ Similarty = \frac{\mathbf{a} \cdot \mathbf{b}}{||\mathbf{a}||_2 ||\mathbf{b}||_2}\,,$$

where a and b are both vectors with the same dimension.

The neighborhood method find $item_j$'s k nearest neighbors (k-NN). These neighbors infer the potential value of r_{ij} to different degree according to their similarity with $item_j$. Although there are several different similarity measures employed to compute the similarity between the items, the similarity between items is represented by the distance between their rating vectors. The similarity of the items which have less common raters are structurally lower. If there're high level features extracted to represent the user and the item, the similarity can be better measured this way. Matrix factorization methods learn this lesson.

Matrix Factorization

To extract high level feature, matrix factorization methods try to find the rating matrix's low rank approximations [15, 21]. They focus on fitting the user-item rating matrix by low-rank approximation and use the fitting result to make sequent predictions [6, 7, 8, 16]. The premise behind this low-dimensional factor model is that there is only a small number of factors or features influencing preferences, and that a user's preference for an item is only determined by that user's feature vector and that item's feature vector.

What is related to our work is not the basic matrix factorization methods. Recently, some matrix factorization methods which involve auxiliary information analysis draw our attention. [13] proposes an trust-aware collaborative filtering algorithm. The algorithm is based on the general knowledge that people normally ask friends for recommendations. Due to the memory-based model, this algorithm suffers from huge online costs. Trust values need to be computed like similarity measures. SoRec [10] fuses the existed trust-based approach with Probabilistic Matrix Factorization (PMF) [16]. This methods is model-based, but it cannot be widely applied due to the scarce resource of trust information which involves people's privacy. [9] proposes a relation regularized matrix factorization method for relational data analysis. Yet it is designed for making recommendations concerning objects that have both content and links. The idea of Collective Matrix Factorization [19] is innovative: factorizing multiple matrices simultaneously with shared parameters. The weakness of this method is that the parameter learning process is computationally costly.

TAG-BASED ITEM RECOMMENDATION

Since tags and ratings are two of the most attributes attached to items, we propose a generalized neighborhood recommendation method to make use of them in the same time. Our work is based on the assumption that the behavior of tagging and rating share the same motivation: item classification. In this sense, the latent preference information found in tagging data has more power than that in rating data. Regarding tags, there are two types of recommendation: item recommendation and keyword recommendation. Our concern is item recommendation which is the same with most CF recommendation methods. In the background of electronic commerce and video on demand, proper item recommendations are better since the items are overwhelmingly numerous.

Topic Finding

As with the rating data, the tag data can be represented as a $n \times m$ sparse matrix T given n users and m items,

$$t_{ij} = \begin{cases} user_i\text{'s tags for } item_j\,, & \text{if } user_i \text{ has tagged } item_j \\ null\,, & \text{otherwise.} \end{cases}$$

The users are allowed to give more than one tag to each item. So if the tags are clearly separated, T becomes a three-dimensional tensor. The three dimensions are user, item, and tag. This is a tough case to take care of and it is why there is little work on extract preference information from this data resource. Innovatively, we divide T into user-tag and item tag matrices representing the tags given by the users and the tags received by the items, respectively. The user-tag and item-tag matrices are denoted as T^U and T^I which are defined as follows:

$$T^U = [\,\mathbf{t}_1,\, \mathbf{t}_2,\, ...,\, \mathbf{t}_n]^T\,,$$
$$T^I = [\,\mathbf{t}_1,\, \mathbf{t}_2,\, ...,\, \mathbf{t}_m]^T\,,$$

where $\mathbf{t}_u$ denotes the tags $user_u$ has given, and $\mathbf{t}_i$ denotes the tags $item_i$ has been given.

When the original tensor T is converted into bags of words in T^U and T^I, we can apply LDA [3] to find latent topic information therein. Processing T^U and T^I separately, we find their latent topics in the form of probability.

$$\theta^U_{ij} = p(topic = j|user = i)\,,$$
$$\theta^I_{ij} = p(topic = j|item = i)\,,$$

where θ^U_{ij} denotes $user_i$' probability of preferring for $topic_j$ and θ^I_{ij} denotes $item_i$' probability of being related to the $topic_j$. This is a kind of "soft clustering". It is possible that a user or item is under multiple topics with the same probability. The similarity between these row vectors in Θ^U and Θ^I more appropriately reflects the users' and items' similarity because clustering is based on the semantics of the tags. The matrices Θ^U and Θ^I are not directly used for rating prediction. Because they are full matrices which are not appropriate for computation, we set a threshold value to reserve high similarity relations and clear the others. Another important reason for this process is that most of the users and items are indirectly related with each other in reality.

After finding the matrices Θ^U and Θ^I, it is easy to employ k-NN clustering to find the groups whose members share the same interests or attributes.

Rating Prediction

We assume all the users who tag the items also give ratings and that all the items which are tagged also receive ratings. If some users actually fail to give either ratings or tags, we still can make use of what they input to the recommender system. Even with few entries, the recommender system still understands what the user wants. We hold this claim because tags are more informative. If the user only put one tag "amine" to some item, we could infer that this is a animation fun. But if this user only give a high rating to the movie "Avatar", what should we infer from this? Which groups of movie does this user like, actions, adventures, fantasies or sci-fis?

Most recommender systems use the integral interval $[1, R_{max}]$ to represent the users' preference on items. It is necessary to normalize the ratings into the range to $[0, 1]$, because only this interval makes sense for probability. There are many mapping methods. One of the most widely used mapping function is $f(x) = (x - 1)/(R_{max} - 1)$. As far as we know, the influence exerted on the final recommendation by different mapping functions is not significantly different.

So our next step is to make rating predictions based on the grouping results stated in the last section. The prediction is made according to a neighborhood method:

$$\begin{aligned} \hat{r}_{ij} &= \mu + b_i^* + b_j^*, \\ b_i^* &= \frac{\sum_{h \in T(i)} (r_{hj} - b_{hj}) I_{hj}^R}{\sum_{h \in T(i)} I_{hj}^R}, \\ b_j^* &= \frac{\sum_{h \in T(j)} (r_{ih} - b_{ih}) I_{ih}^R}{\sum_{h \in T(j)} I_{ih}^R}, \end{aligned} \tag{3}$$

where b_i^* denotes $user_i$'s bias for the topic $T(i)$, and b_j^* denotes $item_j$'s bias for the topic $T(j)$. $T(i)$ and $T(j)$ denote the topic $user_i$ is interested in and the topic $item_j$ is under, respectively. Each topic is a set which contains a number of users or items.

Plus, we give different weights to the neighbors with different distances. The algorithm's weighted variant is

$$\begin{aligned} \hat{r}_{ij} &= \mu + b_i^* + b_j^*, \\ b_i^* &= \frac{\sum_{h \in T(i)} (r_{hj} - b_{hj}) S_{hi} I_{hj}^R}{\sum_{h \in T(i)} S_{hi} I_{hj}^R}, \\ b_j^* &= \frac{\sum_{h \in T(j)} (r_{ih} - b_{ih}) S_{hj} I_{ih}^R}{\sum_{h \in T(j)} S_{hj} I_{ih}^R}, \end{aligned} \tag{4}$$

where θ_h, θ_i and θ_j denote the row vectors of probabilities in Θ. S represents the cosine similarity of the vectors θ_h and θ_j.

EXPERIMENTAL ANALYSIS

Dataset Description

Movielens Dataset is created by Movielens movie recommender which aims to provide online movie recommendation service [17]. Their work is a more involved system rather than a particular algorithm, so we do not delve into it. Their dataset includes three files. One is rating data which contains users' ratings for movies, another is tagging data which contains movies' tags and the user's id who made the tag, and the other is movie overview data which contains the movie's name, release year and genre. The user are allowed to give ratings and tags to the movies they have seen. The ratings are integers between 1 to 5.

We intend to leverage tag analysis to help rating prediction. So we need the movies that have both ratings and tags and the users that give both ratings and tags. After taking the intersection of the rating data and tag data, we get the rating data's subset which contains all the tagged movies. This subset contains 905686 ratings from 4001 users for 7600 movies. The density of these rating data is

$$\frac{905686}{4001 \times 7600} = 2.974\%.$$

From this subset, we randomly and independently choose 20%, 50%, 80% and 99% of rating data as separate training sets. The remaining rating data are used as the testing sets. The experiments are all repeated five times to reduce errors.

Toy Examples

Because the quality of recommendation is eventually reflected in the results of rating prediction accuracy. To obtain a clearer vision about the qualitative quality, we present two toy examples in smaller data volume scale.

First we extract the tags from 6 users. The tag matrix T^U is as follows:

$$\begin{pmatrix} horror & killer & action & horror \\ horror & action & thrill & action \\ fantasy & anime & fantasy & anime \\ anime & Japanese & anime & fantasy \\ documentary & 911 & terrorist & hero \\ historic & documentary & American & realistic \end{pmatrix}$$

We set the hyper-parameter topic number as 3 and conduct LDA analysis to get the probabilistic matrix

$$\Theta^U = \begin{pmatrix} 0.345679 & 0.345679 & 0.308642 \\ 0.364198 & 0.308642 & 0.308642 \\ 0.308642 & 0.345679 & 0.345679 \\ 0.327160 & 0.345679 & 0.327160 \\ 0.345679 & 0.308642 & 0.345679 \\ 0.308642 & 0.345679 & 0.345679 \end{pmatrix}$$

It is quite obvious that $user_1$ and $user_2$ have the same or similar interests. The first column values is the maximum among all three columns for both of him, which infers the topic they are most probably interested in is the first topic. For the same reason, $user_3$ and $user_4$

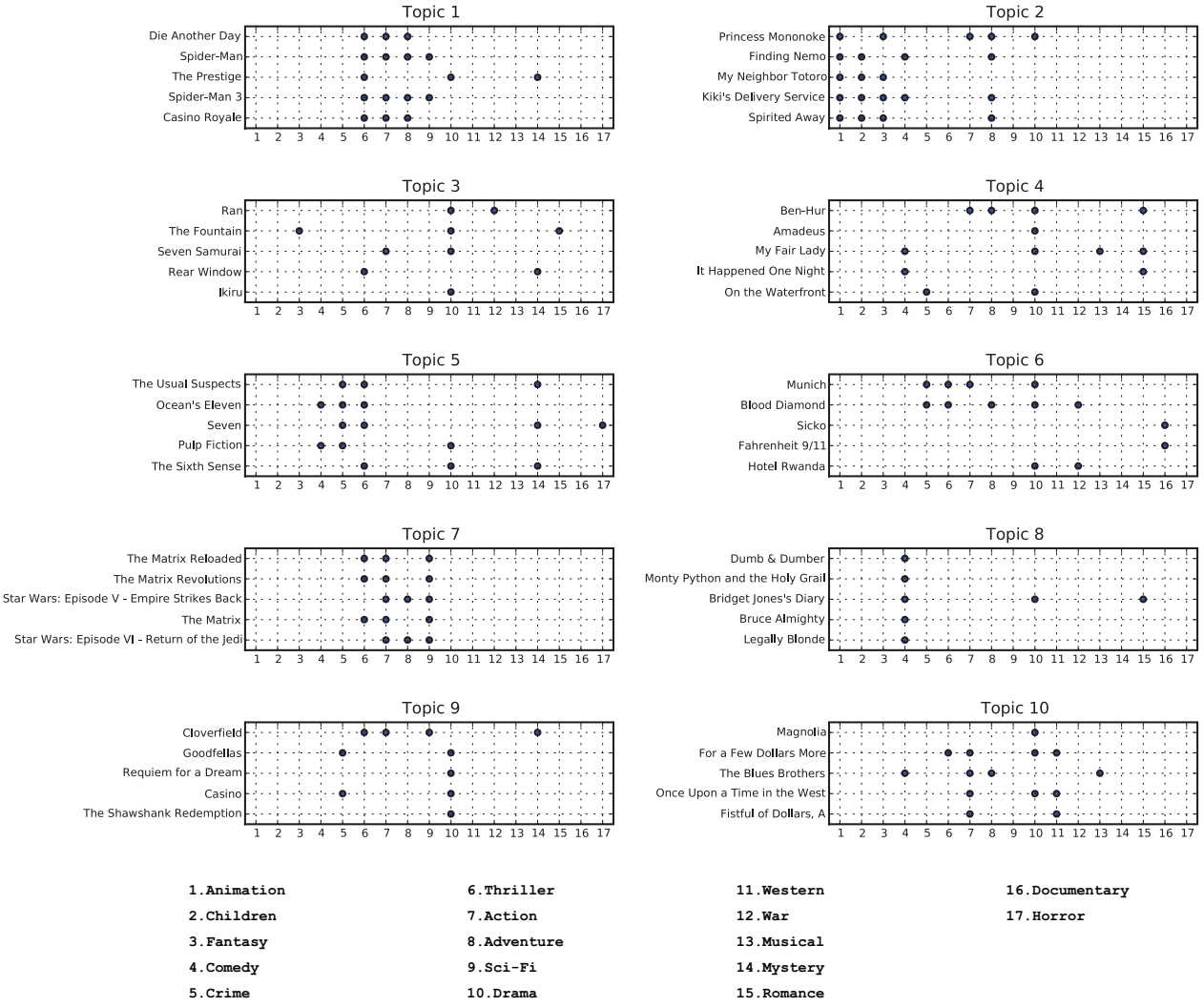

Figure 1. Top 5 movies under 10 different topics and their provided genre information

are similar and $user_5$ and $user_6$ are alike. The subjective grouping result is exactly the same with that of a k-NN method to the matrix $Theata^U$ with the same hyper-parameters. Considering the semantic meanings of these 24 tags, we conclude that the result of the analysis on the user-tag matrix is persuasive. The analysis on the item-tag matrix is effective in a similar way. Yet, there are more merits for analyzing the item-tag matrix because we can obtain the names and genres of the movies from the overview data of the dataset.

In the second example, we extract the tag data in Movielens dataset and get the movie-tag matrix. There are 7600 movies in this matrix. We make the tag analysis to find the latent topics again. For the sake of leaving more space for showing more important results, we set the desired topic number as 10. Fig.1 presents the five most probable movies per topic. According to it, we have several observations as follows:

1. There are at least 4 movies under the same genre for every topic. $Topic1$, $Topic2$ and $Topic3$ all have more than two such common genres. The high co-occurrence frequency of different movies under the same genre reflects the large extent of conformation. The coexistence of movie series like *The Matrix* and *Star Wars* under $Topic7$ illustrates our tag analysis can find not only the movies of the same genres but also the movies of the same series.

2. The five movies in $Topic6$ all reflect big social problems. This problem could be war, terrorist attack, or social security crisis. This explains why these movies under different genres are in the same topic. The topic finding result is not equal to the genre classification. The topic "social problem" may be interesting for some of the users. These details are more valuable for inferring the users' preference than genres.

3. According to the corresponding rating data, the average variance of ratings of the five movies under their

corresponding topic is just 0.102. It illustrates that users hold similar preferences for the movies with similar probability under each topic. So we posit that consulting the movies with similar probability under each topic can help improve personalized rating prediction.

Metrics

We use the Root Mean Square Error (RMSE) metrics to measure the prediction quality of our proposed approach in comparison with other collaborative methods. RMSE is defined as:

$$RMSE = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}(r_{ij}-\hat{r}_{ij})^2 I_{ij}^R}{\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}^R}, \quad (5)$$

where r_{ij} and $\hat{r}_{ij}$ are the actual rating and predicted rating from N users for M movies. Plus, we use rounded value as our predicted rating. The errors of prediction with rounded rating value are more obvious. But whether the rating is rounded or unrounded, the comparison result between different approaches does not change a lot.

Comparison

We compare our approach with two collaborative filtering algorithms:Non-negative Matrix Factorization (NMF) method, PMF method and the improved regularized SVD method. In fact, one of the most difficult problems in our work is to find some coordinate algorithms for comparison. Because our intention is to provide a generalized item recommendation model to combine the use of ratings and tags, most of the related work is inapplicable to the data resource in this situation. We choose three of the most popular algorithms by expediency.

The parameters of these two method also need to be tuned. According to the relative works and our experiments, the best parameters for the PMF approach on Movielens dataset are like these: $\lambda_u = 0.001$, $\lambda_v = 0.0001$. Concerning the improved regularized SVD method, $lrate = 0.001$, $\lambda = 0.02$, $\lambda_2 = 0.05$. We set the number of feature dimensions as 80. We think this assignment is reasonable because the commonly used feature dimension for these matrix factorization is between 30 and 100.

We have six versions of the improved collaborative filtering methods. $Nghbru$ represents the neighborhood recommendation method based only on the user-tag analysis. $Nghbri$ corresponds to the variant based only on the item-tag analysis. $Nghbra$ integrates the use of the user-tag analysis and the item-tag analysis. Each of these three methods, there are two different weighting strategies. One is to use uniform weights, labeled as "Avg"; the other is to use different weights, labeled as "Wgt".

Table 1. RMSE comparison with other approaches (A smaller RMSE value means a better performance)

		RMSE			
Percentage		20%	50%	80%	99%
NMF		1.4854	1.3027	1.1275	1.0762
irSVD		1.3176	1.2591	1.1928	1.1087
PMF		1.1692	1.1187	1.05656	1.0173
Nghbru	Avg	0.8811	0.8799	0.8807	0.8803
	Wgt	0.8802	0.8792	0.8796	0.8788
Nghbri	Avg	0.8802	0.8798	0.8791	0.8789
	Wgt	0.8802	0.8796	0.8790	0.8788
Nghbra	Avg	0.8669	0.8668	0.8665	0.8662
	Wgt	**0.8661**	**0.8658**	**0.8657**	**0.8655**

The results in Table 1 show our neighborhood recommendation method outperforms the improved regularized SVD method more than 41% , NMF 36%, and PMF 23%. We would like to analyze the results more specifically: 1)For all these algorithms in Table 1, the prediction accuracy increases as the training set's percentage ascends. This is reasonable because with high training data percentage, our algorithms find more neighbors to consult. The more neighbors we find, the more accuracy we get. 2)Among our own several methods, the version Nghbra-Wgt presents the best performance. It illustrates that utilizing all the tag information and assigning different weights to this this tag information is meaningful. 3)We also observe that item tag analysis is a little more effective than the user tag analysis. Although the difference is subtle, it explains the fact that the item-based collaborative filtering approaches are more popular than the user-based ones in early works. 4)Besides, we find the performance increase of $Nghbra$ is obvious compared with $Nghbru$ and $Nghbri$. This illustrates that the fusion of the user tag analysis and the item tag analysis is lossless. 5)Nevertheless, the performance of the weighted version of $Nghbra$, $Nghbru$ and $Nghbri$ is not much better than their average counterparts. This can be explained by the homogeneity of users. There are no authorities to give the overwhelmingly important rating. The phenomenon reflects the democracy in the online social network.

Parameter Analysis

For topic finding, we set the Dirichlet priors α and β to $50/K$ and 0.1, respectively (K is the number of topics). These two hyper-parameters are the empirical values for LDA. The threshold value of processing probabilistic matrices Θ^U and Θ^I is set as 0.03 which means statistically impossible. The other two parameters, *iteration number* and *topic number* are unfixed. We explore the optimal solutions for them. Because the parameters of topic finding are different regarding the objects to analyze, we separate the process of tag analysis into user-tag analysis and item-tag analysis. We observe a shape RMSE increases with huge vibrations after 340 iterations for user-tag analysis. This can be seen as the signal of overfitting. Regarding the item-tag analysis,

we observe the optimal iteration number is 480. So we think the optimal iteration number for them is 340 and 480, respectively. And we use these two parameters in the above experiments.

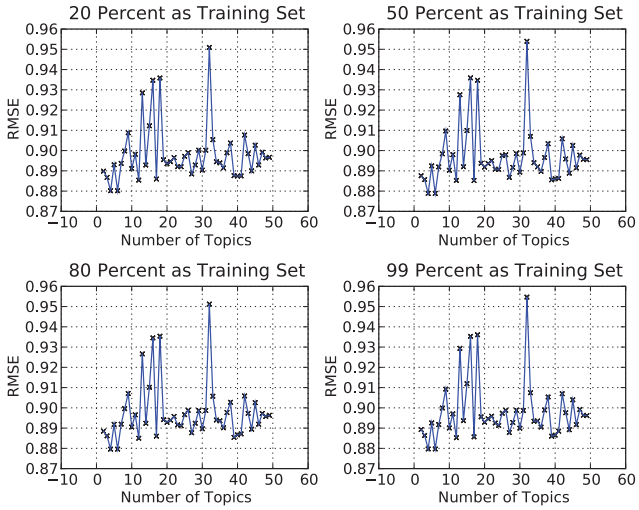

Figure 2. Dependence between the topic number and the prediction accuracy for the items' tag analysis

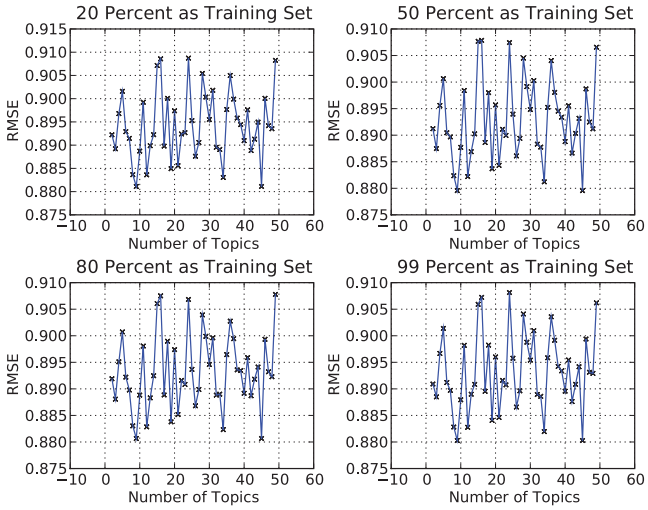

Figure 3. Dependence between the topic number and the prediction accuracy for the users' tag analysis

Compared with the iteration number, we are more interested in the dependence between the topic number and the prediction accuracy. We record the effects to the prediction preciseness for the topic numbers ranging between $(1, 50)$. From Figure 2 we observe that the optimal topic number for the items' tag analysis is less than 25. The optimal value does not necessarily mean the global optimal value. Although just local optimum, the RMSE value near the topic number of 25 is stably less than 0.9. The observation that there are some better parameter choices less than 10 can be explained by the data sparsity. Rating data sparsity is always existent and thus there are possibilities that our algorithm cannot find enough consultants for potential rating inference when the topic number is set relatively large. One probable case is like this: the neighbors found by our approach is very similar to the movie we are to predict, but the user has not given a rating to it.

For user-tag analysis, the optimal topic number is 23 as Figure 3 illustrates. The situation here is similar with that in item-tag analysis. The local minimum is not the global one. The reason for this is the same as mentioned before. What is different is the stable duration here is shorter than that in item-tag analysis and the fluctuation here is more obvious. This observation can be explained by the diversity of personal interests. Compared with movies, the attributes of human beings are more dynamic and diverse. It is easier to find similar items than similar people because the measure in the latter situation is vaguer.

In summary, the optimal topic number is around 25 in both two situations, which means the results are consistent. And the genre number in common use is the same order of magnitude. From this perspective, our results are reasonable. But we must emphasize the fact that our method of topic finding and the common use of genre classification focus on different targets and thus produce different results. Considering the fact that the information of genre demands the knowledge from experts, our method of topic finding has wider range of application.

Discussion

There are some issues concerning implemental details we need to explain her. Because tags are given with high freedom, there are a lot of preprocessing work to do. First, there are many noise such as ":D" in the data. We absolutely should remove them all. But in fact, we are unable to guarantee all noise are cleared off because they lack rules. Second, stoplist is one of the most important parts to manipulate. To our best knowledge, most stoplists used in document clustering remove the word "good" and "great". Concerning our approach, these words reflect the users' preference to the items. It is somehow meaningful. We hesitate to remove these words because it may benefit rating prediction. In our experiments, we remove the prepositions, conjunctions and other less meaningful words while leave emotional words untouched. Third, stemming is also a complicated technique that we must employ. It may be simpler for ordinary document and webpage retrieval. But the bags of tags in our experiment are rather chaotic. We are worried the stemming algorithm may to some extent have a negative influence on the quality of topic finding. If we take better care of these three factors, we should further improve the quality of recommendation to some extent.

CONCLUSIONS AND FUTURE WORK

In this paper we proposed a novel method to alleviate the sparsity problem and improve the quality of the collaborative filtering method. We make use of the tag information to find closer neighbors for the users and the items, respectively. These neighbors give strong inferences for the potential preference of the user for the item. We utilize these inferences to make the rating prediction. According to the experiments, our approach's prediction for the users' preference is much more accurate than the art-of-state ones such as NMF method, PMF method and the improved regularized SVD method.

Finding neighbors is vital for an excellent collaborative filtering algorithm. The motivation of our work is to find better neighbors, which give stronger inference for the prediction. Latent topics connect the users and items with similar interests together. Finding these topics is equal to finding the neighbors. The connection that cannot be discovered in the rating records can be disclosed through learning about the tagging history. This is why our method outperforms the others.

The next step for us is to improve our method in two aspects. One is incrementalization. The neighborhood method enjoys the low computational complexity, but suffers from the rigidity to frequent update. If there are a lot of new entries from the users to items, our current solution fail to deal with this situation. On the other hand, we can fuse the collaborative matrix factorization method with our topic finding model. This is another way to make use of the latent topic information. We believe these methods are both promising solutions to further improve the collaborative filtering technique.

ADDITIONAL AUTHORS

REFERENCES

1. R. Bell and Y. Koren. Improved neighborhood-based collaborative filtering. In *KDD-Cup and Workshop*. Citeseer, 2007.
2. R. M. Bell and Y. Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *ICDM*, pages 43–52, 2007.
3. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
4. D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, 1992.
5. D. M. Hawkins, S. C. Basak, and D. Mills. Assessing model fit by cross-validation. *J. Chem. Inf. Comput. Sci.*, 43(2):579–586, March 2003.
6. T. Hofmann. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 259–266, New York, NY, USA, 2003. ACM.
7. T. Hofmann. Latent semantic models for collaborative filtering. *ACM Trans. Inf. Syst.*, 22(1):89–115, 2004.
8. Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD*, pages 426–434, 2008.
9. W. Li and D. Yeung. Relation regularized matrix factorization. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, 2009.
10. H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: Social recommendation using probabilistic matrix factorization. In *CIKM*, pages 931–940, 2008.
11. A. Marchetti, M. Tesconi, F. Ronzano, M. Rosella, and S. Minutoli. SemKey: A semantic collaborative tagging system. In *Proceedings of 16th International World Wide Web Conference, WWW2007*. Citeseer, 2007.
12. B. Marlin. Collaborative filtering: A machine learning perspective. Master's thesis, University of Toronto, 2004.
13. P. Massa and P. Avesani. Trust-aware recommender systems. In *RecSys '07: Proceedings of the 2007 ACM conference on Recommender systems*, pages 17–24, New York, NY, USA, 2007. ACM.
14. B. J. Mirza, B. J. Keller, and N. Ramakrishnan. Studying recommendation algorithms by graph analysis. *Journal of Intelligent Information Systems*, 20(2):131–160, March 2003.
15. N. S. Nati and T. Jaakkola. Weighted low-rank approximations. In *In 20th International Conference on Machine Learning*, pages 720–727. AAAI Press, 2003.
16. R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, 2007.
17. S. Sen, J. Vig, and J. Riedl. Tagommenders: Connecting users to items through tags. In *WWW*, 2009.
18. B. Sigurbj "ornsson and R. Van Zwol. Flickr tag recommendation based on collective knowledge. 2008.
19. A. Singh and G. Gordon. Relational learning via collective matrix factorization. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 650–658. ACM, 2008.

20. Y. Song, Z. Zhuang, H. Li, Q. Zhao, J. Li, W. Lee, and C. Giles. Real-time automatic tag recommendation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 515–522. ACM, 2008.

21. N. Srebro. *Learning with matrix factorizations*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2004.

22. F. Suchanek, M. Vojnovic, and D. Gunawardena. Social tags: meaning and suggestions. 2008.

23. J. Wang, A. P. de Vries, and M. J. T. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 501–508, New York, NY, USA, 2006. ACM Press.

24. S. Wasserman, K. Faust, and D. Iacobucci. *Social Network Analysis : Methods and Applications (Structural Analysis in the Social Sciences)*. Cambridge University Press, November 1994.

25. X. Wu, L. Zhang, and Y. Yu. Exploring social annotations for the semantic web. In *Proceedings of the 15th international conference on World Wide Web*, page 426. ACM, 2006.

26. S. Zhang, W. Wang, J. Ford, and F. Makedon. Learning from incomplete ratings using non-negative matrix factorization. In *SDM*, 2006.

27. A. Zollers. Emerging motivations for tagging: expression, performance, and activism. *WWW 2007*, 2007.