Comparison of Recommender System Algorithms focusing on the New-Item and User-Bias Problem

Stefan Hauger[1], Karen H. L. Tso[2], and Lars Schmidt-Thieme[2]

[1] Department of Computer Science, University of Freiburg
Georges-Koehler-Allee 51, 79110 Freiburg, Germany
hauger@informatik.uni-freiburg.de
[2] Information Systems and Machine Learning Lab, University of Hildesheim
Samelsonplatz 1, 31141 Hildesheim, Germany
{tso,schmidt-thieme}@ismll.uni-hildesheim.de

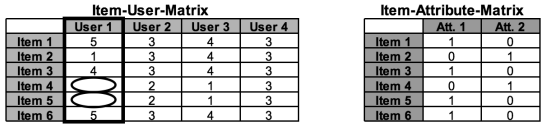
Abstract. Recommender systems are used by an increasing number of e-commerce websites to help the customers to find suitable products from a large database. One of the most popular techniques for recommender systems is collaborative filtering. Several collaborative filtering algorithms claim to be able to solve i) the new-item problem, when a new item is introduced to the system and only a few or no ratings have been provided; and ii) the user-bias problem, when it is not possible to distinguish two items, which possess the same historical ratings from users, but different contents. However, for most algorithms, evaluations are not satisfying due to the lack of suitable evaluation metrics and protocols, thus, a fair comparison of the algorithms is not possible.

In this paper, we introduce new methods and metrics for evaluating the user-bias and new-item problem for collaborative filtering algorithms which consider attributes. In addition, we conduct empirical analysis and compare the results of existing collaborative filtering algorithms for these two problems by using several public movie datasets on a common setting.

1 Introduction

A Recommender system is a type of customization tool in e-commerce that generates personalized recommendations, which match with the taste of the users. Collaborative filtering (CF) (Sarwar et al. (2000, 2001)) is a popular technique used in recommender systems. It is used to predict the user interest for a given item based on user profiles. The concept of this technique is that the user, who received a recommendation for some sorts of items, would prefer the same items as other individuals with a similar mind set.

However, besides its simplicity, one of the shortcomings of CF are the new-item or cold-start problem. If no ratings are given for new items, it is difficult



Item-User-Matrix

	User 1	User 2	User 3	User 4
Item 1	5	3	4	3
Item 2	1	3	4	3
Item 3	4	3	4	3
Item 4	?	2	1	3
Item 5	?	2	1	3
Item 6	5	3	4	3

Item-Attribute-Matrix

	Att. 1	Att. 2
Item 1	1	0
Item 2	0	1
Item 3	1	0
Item 4	0	1
Item 5	1	0
Item 6	1	0

Fig. 1. User-Bias Example

for standard CF algorithms to determine their own clusters by using rating similarity and thus they fail to give accurate predictions. Another problem is the user-bias from historical ratings (Kim and Li (2004)), which occurs when two items, based on historical ratings have the same opportunity to be recommended to a user, but additional information shows that one item belongs to a group which is preferred by the user and the other not. For example, as shown in Figure 1, by applying CF, the probabilities that item 4 and 5 to be recommended for user 1 are equal. When the attributes are also taken into consideration, it can be observed that items 1, 3 and 6 which belong to attribute 1 are rated higher than user 1 than item 2 which belongs to attribute 2. Thus, user 1 has a preference for items related to attribute 1 over items related to attribute 2. Subsequently, by the CF algorithm, a higher probability should be assigned to item 5, which is more attached to attribute 1, than to item 4, which is related to attribute 2.

Recommender system algorithms that incorporate attributes claim to solve the user-bias and the new-item problem, however, no good evaluation techniques exist. For that reason, in this paper, we make the following contributions: (i) we introduce new methods and metrics for evaluating these problems and (ii) through a common experimental setting, we present evaluation results for three existing CF algorithms, which do not take attributes into account, namely user-based CF (Sarwar et al. (2000)), item-based CF (Sarwar et al. (2001)) and Gaussian aspect model by Hofmann (2004) as well as an approach, which takes attributes into account, by Kim & Li (2004). In the next section, we present the related work. In section 3, a brief description of the aspect model by Hofmann and the approach by Kim & Li will be presented. An introduction of the evaluation techniques for the new-item and the user-bias problem will follow in section 4. Section 5 consists of results on the empirical evaluations we have conducted and in section 6 we present the conclusions of the results and discuss possible future work.

2 Related Works

Evaluating CF algorithms is not anything novel as there have already been relatively standard measures for evaluating the CF algorithms. Most of the evaluations done on CF focus on the overall performance of the CF algorithms (Breese et al. (1998), Sarwar et al. (2000), Herlocker et al. (2004)). However, as mentioned in the previous section, CF suffers from several shortcomings which are the new-item problem, also known as the cold-start problem, as well as the user-bias problem. It has been claimed that incorporating attributes could help to alleviate these drawbacks (Kim and Li (2004)). In fact, there

exist many approaches for combining content information with CF (Burke (2002), Melville et al. (2002), Kim and Li (2004), Tso and Schmidt-Thieme (2005)). However, there has been lack of suitable evaluations which compute comparative analysis of attribute-aware and non attribute-aware CF algorithms, focusing on these two problems.

Schein et al. (2002) have already discussed methods and metrics for the new-item problem, in which they have introduced a performance metric called CROC curve. However, this metric is only suitable for the new-item problem. In this paper, we use standard performance metric, but introduce new protocols for evaluating the new-item and the user-bias problems. Hence, this evaluation setting allows users to compare the results with standard CF evaluation metrics, which does not restrict to evaluate only the new-item problem, but also on the user-bias problem. In addition, we compare the predicting accuracy of various collaborative filtering algorithms in this evaluation setting.

3 Observed Approaches

In this section, we present a brief description of the two state-of-the-art CF models: the aspect model by Hofmann (2004) and the approach by Kim & Li (2004).

Aspect Model by Hofmann

Hofmann (2004) specified different versions of the aspect model regarding the collaborative filtering domain. In this paper, we focus on the Gaussian model, because it shows the best prediction accuracy for non-specific problems. He uses the aspect model to identify the hidden semantic relationship among item y and users u, by using a latent class variable z, which represents the user clusters associated with each observation pair of a user and an item. In the aspect model, the users and items are considered as independent from each other and every observation can be described by a quartet $< u, y, v, z >$, where v denotes the rating user u has given to item y. For every observation quartet, the probability is then computed as follows:

$$P(u, y, v, z) = P(v|y, z)\ P(z|u)\ P(u)$$

The focus of our evaluation in this paper is on the Gaussian pLSA model, in which $P(v|y, z)$ is represented by the Gaussian density function. In the gaussian pLSA model, every combination of z and an item y has a location parameter $\mu_{y,z}$ and a scale parameter $\sigma_{y,z}$. The probability of the rating, v is then:

$$P(v|y, z) = P(v; \mu_{y,z}, \sigma_{y,z}) = \frac{1}{\sqrt{2\pi}\sigma_{y,z}} exp\left[-\frac{(v - \mu_{y,z})^2}{2\sigma_{y,z}^2}\right]$$

As z is unobserved, Hoffmann used the Expectation Maximization (EM) algorithm to learn the two model parameters: $P(v|y, z)$ and $P(z|u)$. The EM

algorithm has two main steps. The first step is computation of the Expectation (E-Step), which is done by computing the variation distribution Q over the latent variable z. The second step is Maximization (M-Step), in which the model parameters are updated by using the Q distribution computed in the previous E-Step. These two steps are executed until it converges to a local optimal limit. The EM steps for the Gaussian pLSA model are:

E-Step:
$$Q(z; u, y, v, \theta) = \frac{P(z|u)P(v; \mu_{y,z}, \sigma_{y,z})}{\sum_{z'} P(z'|u)P(v; \mu_{y,z}, \sigma_{y,z})}$$

M-Step:
$$P(z|u) = \frac{\sum_{<u',y>:u'=u} Q(z; u', y, v, \theta)}{\sum_{z'} \sum_{<u',y>:u'=u} Q(z'; u', y, v, \theta)}$$

The location and scale parameters would also have to be updated.

Analogously, the same model can be applied by representing the latent class variable z, not as the user communities but as item cluster.

Approach by Kim and Li

The approach by Kim & Li (2004) seeks to solve the problem of user-bias and the new-item with the help of item attributes. They have incorporated attributes of movies such as genre, actors, years, etc. to collaborative filtering. It is expected that when attributes are considered, it is possible to recommend a new item based on just the user's fondness of the attributes, even though no user has voted for the item.

Kim & Li have a rather similar model as the aspect model by Hoffmann, yet there are several differences. First, class z associates only with the item, but not with the users in contrast to the pLSA model by Hofmann. Note that, the latent class z in this approach is regarded as an item clusters, instead of the user communities. Furthermore, they have applied some heuristic techniques to compute the corresponding model parameters, which can be done in two steps. First, using attributes, they clustered the items in different cliques with a simple K-means clustering algorithm. After clustering the items, they computed the probability of every item, i.e. the value indicating how much the item belongs to every clique. Then, an item-clique matrix with all the probabilities is derived. In the second step, the original item-user matrix is extended with the item-clique matrix, thus the attribute-cliques are just used as normal users.

Class z is built with the help of the extended item-user matrix. Every class z consists of a number of items of high similarity. The quality of class z is responsible for the accuracy of the later prediction of the use vote. A K-Medoids clustering algorithm using the Pearson's Correlation is used to compute the classes. After clustering the items into class z, a new item for each class z is created using the arithmetic mean. This new item is then the representative vector of the class z.

With the help of these representative items and a group matrix, which stores the membership of every item of the item-user matrix, it is possible to compute the expected vote for a user. In calculating the prediction, it is assumed that class z satisfies the Gaussian distribution. Let V_y be the rating vector of item y, V_z the representative vector of item cluster z, $ED(\cdot)$ the Euclidean distance, $v_{u,y}$ the user u's vote on item y and U_z the set of items, which are in the same item cluster z, then the membership degree $p(z|y)$ and the mean rating, $\mu_{u,z}$, of user u on class z can be calculated as follows:

$$p(z|y) = \frac{1/ED(V_y, V_z)}{\sum_{z'=1}^{k} 1/ED(V_y, V_{z'})} \qquad \mu_{u,z} = \frac{\sum_{y \in U_z} v_{u,y} p(z|y)}{\sum_{y \in U_z} p(z|y)}$$

4 Evaluation Protocols

New-item Problem

To evaluate the prediction accuracy, we use a protocol which deletes one vote randomly from every user in the dataset, the so-called, AllBut1 protocol (Breese et al. 1998). The new-item problem is evaluated by a protocol similar to the AllBut1 protocol. Likewise, this protocol also deletes existing votes and builds up the model, which is to be evaluated with the reduced dataset. The new items are created by deleting all votes for a randomly selected item. After this is done for the required number of items, one vote is deleted from each user as in the AllBut1 protocol. This protocol has the advantage that the results of the new items can be compared with the results for past-rated items. Mean Absolute Error (MAE) is used as metrics in our experiments.

User-bias Problem

The user-bias problem occurs, when two items have the same rating, but one item belongs to a group of items, which have not been given a good vote by the user, whereas the other item belongs to a group, which was in contrast given a good vote by the user; then the item, which belongs to the good-rated group, should be recommended.

To find a pair of items for an user, all the items, which are rated by the user, are taken into consideration and grouped two times. Once in item groups with equal rating and the second time in items groups with equal attributes. The historical vote vectors of these pairs of items of the users are then compared, excluding the vote of the observed user. In the next step, we select all pairs of items, which are in the same group of equally rated items and different group of attributes. One pair, which is to be predicted, is randomly chosen and deleted from the dataset. This is then done for all users in the database.

For each of these 'user-biased' pairs, the vote prediction for these pairs are computed and compared with the four collaborative filtering algorithms we use in our experiments. MAE metric is used to evaluate the predicting accuracy.

5 Evaluation and Experimental Results

Two datasets are used for our experiments - the EachMovie, containing 2,558,871 votes from 61,132 users on 1,623 movies, and the MovieLens100k dataset, containing 100,000 ratings from 943 users on 1,682 movies. The datasets also contain genre information for every movie in binary presentation, which we used as attributes. The EachMovie dataset contains 10 different genres, MovieLens contains 18. We conduct for both datasets 10 samples, in which 10 trials were run. For each sample 1500 movies are selected, whereas a 1000 users in EachMovie and 600 users in MovieLens are selected. and 20 neighbours for MovieLens and EachMovie for both user- and the item-based CF. No normalization is used in the aspect model and z is set to 40 for both datasets. In the Kim & Li approach, we used 20 attribute-groups and 40 item clusters for both datasets. We have selected the above parameter settings, because they were reported as the parameters which have given the best results in former experiments by the corresponding authors.

At first, we compared four observed approaches, namely the user-based CF, item-based CF, aspect model and Kim & Li approach, using the AllBut1 protocol. In Figures 2 and 3, the aspect model performs the best, the approach by Kim & Li is only slightly worse, while the user- and item-based CF algorithms perform the worst.

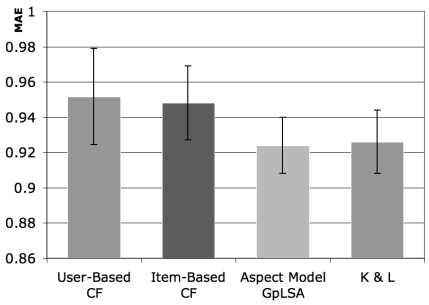


Fig. 2. AllBut1 using EachMovie.

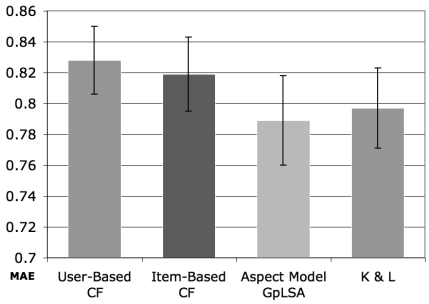


Fig. 3. AllBut1 using MovieLens.

New-item Problem

The results of the new-item problem are presented in Figure 4 and 5. Comparing the performance achieved by the algorithms, which use no attributes and the Kim & Li approach, we can see that the performance of the Kim & Li approach is only negligibly affected when more new items are added, while the predicting accuracy of the other approaches becomes much worse. This phenomenon is in line with our expectations, because it is not possible for algorithms, which do not take the attributes into account, to find any relations between new items and already rated items. As for the Kim & Li approach, there is no difficulty to assign an unrated item to an item cluster, because it includes the attributes. The average standard deviation is about 0.03.

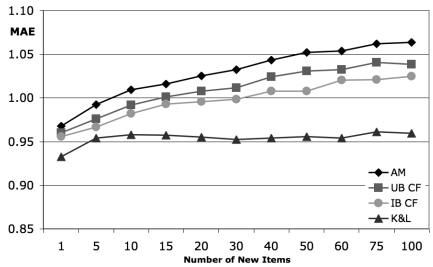


Fig. 4. New-Item using EachMovie.

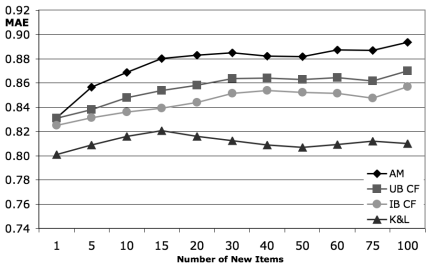


Fig. 5. New-Item using MovieLens.

User-Bias Problem

In the experiments of the user-bias problem, the number of items for prediction is between 60 to 70% of the total number of items, which is a representative amount. Besides, as shown in Figures 6 and 7, our expectations are confirmed. Only the approach by Kim & Li can mine the difference between two items with the same historical rating, but belong to different attributes; while the other approaches do not have any possibility to find the type of items the user likes because they do not take attributes into consideration. It is interesting to see that the aspect model, which performs best in general, performs worst to the user- and item-based CF when special problems such as the user-bias and new-item problem are considered.

6 Conclusion

The aim of this paper is to show that the new-item problem and user-bias problem can be solved with the help of attributes. We have used three CF algorithms, which do not use any attributes, and one approach, which takes the attribute information into account to compute the recommendations in our evaluation. Our evaluations have shown that it is possible to solve the new-item problem and user-bias problem with the help of attributes. In general, the approach by Kim & Li can not surpass the aspect model, but it can solve specific problems of new-item and user-bias more effectively. Especially for the

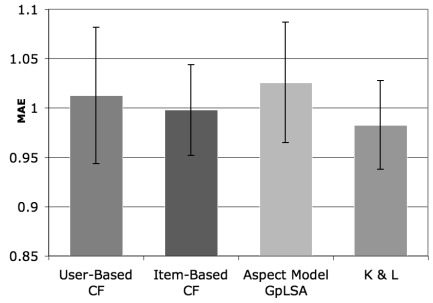


Fig. 6. User-Bias using EachMovie.

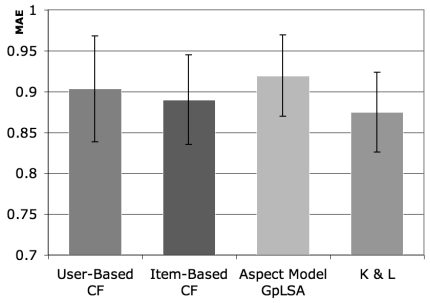


Fig. 7. User-Bias using MovieLens.

new-item problem, where in the reality it is not uncommon to have 30-50 new items being injected to the database. Hence, we can conclude that by applying the right algorithms to the right cases, we can improve the recommendation quality rather significantly.

It can be seen that a small number of attributes could already help to overcome the problem of new-item and user-bias, then it should be possible to improve the results further with more adequate attributes. For future work, it would be interesting to find out, how to select better attributes, and how the attributes affect the performance.

References

BREESE, J.S., HECKERMAN, D., and KADIE, C. (1998): Empirical analysis of predictive algorithms for collaborative filtering. *In Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence, pp. 43–52, July 1998.*

BURKE, R. (2002): Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction. vol. 12(4), pp. 331–370.*

HERLOCKER, J.L., KONSTAN, J.A., TERVEEN, L.G. and RIEDL, J.T. (2004): Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems, vol. 22, no. 1, pp. 5–53, 2004.*

HOFMANN, T. (2004): Latent Semantic Models for Collaborative Filtering. *ACM Transactions on Information Systems, 2004, Vol 22(1), pp. 89–115.*

KIM, B.M. and LI, Q. (2004): Probabilistic Model Estimation for Collaborative Filtering Based on Item Attributes. *IEEE International Conference on Web Intelligence.*

MELVILLE, P., MOONEY, R. and NAGARAJAN, R. (2002): Contentboosted collaborative filtering. *In Proceedings of Eighteenth National Conference on Artificial Intelligence (AAAI-2002), pp. 187–192.*

SARWAR, B.M., KARYPIS, G., KONSTAN, J.A. and RIEDL, J. (2000): Analysis of recommendation algorithms for e-commerce. *In Proceedings of the Second ACM Conference on Electronic Commerce (EC00), 2000, pp. 285–295.*

SARWAR, B.M., KARYPIS, G., KONSTAN, J.A. and RIEDL, J. (2001): Item-based collaborative filtering recommendation algorithms. *In Proceedings of the 10th international conference on World Wide Web. New York, NY, USA: ACM Press, 2001, pp. 285–295.*

SCHEIN, A.I., POPESCUL, A., UNGAR, L.H. and PENNOCK, D.M. (2002):Methods and metrics for cold-start recommendations. *In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. New York, NY, USA: ACM Press, 2002, pp. 253–260.*

TSO, K. and SCHMIDT-THIEME L. (2005): Attribute-aware Collaborative Filtering. *In Proceedings of 29th Annual Conference of the Gesellschaft für Klassifikation (GfKl) 2005, Magdeburg, Springer.*