Exploiting the Characteristics of Matrix Factorization for Active Learning in Recommender Systems

Rasoul Karimi, Christoph Freudenthaler, Alexandros Nanopoulos,
Lars Schmidt-Thieme
Information System and Machine Learning Laboratory
Samelsonplatz 1 University of Hildesheim 31141 Hildesheim Germany
[karimi, freudenthaler, nanopoulos, schmidt-thieme]@ismll.uni-hildesheim.de

ABSTRACT

Recommender systems help web users to address information overload. However their performance depends on the number of provided ratings by users. This problem is amplified for a new user because he/she has not provided any rating. To address this problem, active learning methods have been proposed to acquire those ratings from users, that will help most in determining their interests. However, different from the classic active learning, users (the "oracle") are not always able to provide an answer for queries. The easiest way to solve this problem is to ask most popular items, i.e items which have received many ratings from training users. But it is static and presents the same items to all users regardless of the ratings they have provided so far. In this paper we propose a method that improves the most popular selection strategy using the characteristics of matrix factorization. It finds similar users to the new user in the latent space and then selects item which is most popular *among the similar users*. The experimental results show the proposed method outperforms the most popular method both in terms of error and the number of received ratings.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning Parameter learning

Keywords

Recommender System, Active Learning, New User Problem

1. INTRODUCTION

Recommender systems help web users to address information overload in a large space of possible options [3]. In many applications, such as in e-commerce, users have too many choices and too little time to explore them all. Moreover, the exploding availability of information makes this problem even tougher.

There are several techniques for recommendation. Collaborative filtering is a traditional technique that is widely applied [1, 13, 15]. It makes automatic predictions about the interests of a user by reusing taste information from other users. The underlying assumption of the collaborative filtering approach is that those who agreed in the past tend to agree again in the future.

Collaborative filtering methods fall into two main categories: memory-based algorithms and model-based algorithms. In memory-based techniques, the value of the unknown rating is computed as an aggregate of the ratings of some other (usually, the N most similar) users for the same item [13]. Model-based collaborative techniques provide recommendations by estimating parameters of statistical models for user ratings. Nevertheless, recent research (especially as has been demonstrated during the Netflix challenge[1]) indicates that Matrix Factorization(MF) is a superior prediction model compared to other model-based collaborative methods [14].

Evidently, the performance of collaborative filtering depends on the amount of information that users provide regarding items, most often in the form of ratings. However, a well identified problem is that users are reluctant to provide information for a large amount of items. This fact impacts negatively the quality of generated recommendations. A simple and effective way to overcome this problem, is by posing queries to new users in order that they express their preferences about selected items, e.g., by rating them. Nevertheless, the selection of items must take into consideration that users are not willing to answer a lot of such queries. To address this problem, *active learning* methods have been proposed to acquire those ratings from users, that will help most in determining their interests [5, 8].

However, different from the classic active learning, users (the "oracle") may not know the answer of queries. For example in the movie recommendation scenario, if the new user has not watched the queried movie, he/she is not able to provide a rating for that movie. The easiest way to solve this problem is to ask most popular items, i.e items which have received many ratings from training users. Hopefully, the item which is known for many training users is also known for the new user and he/she can rate it. But the most-popular selection strategy is static and presents the same items to all users regardless of the ratings they have provided so far. An advanced active learning algorithm adapts to the earlier answers given by the new user to improve the preference of elicitation process. In this paper we propose

---

[1]www.netflixprize.com

a method which improves the most popular selection strategy using the characteristics of matrix factorization. It finds similar users to the new user in the latent space and then selects an item which is most popular *among the similar users*. As the similar users change during the active learning process based on the new user's answers to the queries, the active learning algorithm is also automatically adapted to these answers.

The rest of this paper is organized as follows: In Section 2, the related work is reviewed. As the proposed method relies on matrix factorization, a short introduction to matrix factorization is introduces in section 3. The proposed method is explained in section 4 followed by the experimental results in Section 5. Finally, the conclusions are stated in Section 6.

2. RELATED WORK

Active learning, in the context of the new-user problem, was introduced by Kohrs and Merialdo [12]. This work suggested a method based on nearest-neighbor collaborative filtering, which uses entropy and variance as the loss function to identify the queried items. Mamunur et al. [15] expanded this work, by considering the popularity of items and also personalizing the item selection for each individual user. Boutilier et al. [2] apply the metric of expected value of utility to find the most informative item to query, which is to find the item that leads to the most significant change in the highest expected ratings.

Jin and Si [8] developed a new active learning algorithm based on Aspect Model(AM) [7, 6] which is similar to applying active learning for parameter estimation in Bayesian networks [17]. It uses the entropy of the model as the loss function. However, it does not directly minimize the entropy loss function, because the current model may be far from the true model and relying only on the current model can become misleading. To overcome this problem, that work proposes to use a Bayesian network to take into account the reliability of the current model. This Bayesian approach is, however, complex and intractable for real applications (demands excessive execution time). Karimi et. al. [9] applied the simple most popular item selection to AM. The results show that it competes in accuracy with the Bayesian approach while its execution time is in the order of magnitude faster than the Bayesian method.

Karimi et. al [10] developed a non-myopic active learning which capitalizes explicitly on the update procedure of MF model. First, this method queries items that updating the new user features with the provided rating will change the features as much as possible. Its goal is to explore the latent space to get closer to the optimal features. Then, it exploits the learned features and slightly adjusts them. Karimi et. al [11] by being inspired from existing optimal active learning for the regression task, exploits the characteristics of matrix factorization and develops a method which approximates the optimal solution for recommender systems.

All aforementioned methods assume that new users are always able to provide ratings to the queried items. Some methods relax this assumption and develop algorithms for the realistic scenario in which new users might not rate the queried item. [18, 4] leverage a decision tree to address this issue. But training the decision tree is costly in terms of time and memory specially when the size of the dataset is large.

3. MATRIX FACTORIZATION IN RECOMMENDER SYSTEMS

Matrix Factorization (MF) is the task of approximating the true, unobserved ratings-matrix R by $\hat{R} : \mathbb{R}^{|U| \times |I|}$. It maps both users and items to a latent space of dimensionality k. In this space, user-item interactions are modeled as inner products. In the latent space, each item i is represented with a vector $h_i \in R^k$. The elements of h_i indicate the importance of factors in rating item i by users. Some factors might have higher effect and vice versa. In the same way, each user u is represented with a vector $w_u \in R^k$ in the latent space. For a given user the element of w_u measure the influence of the factors on user preferences. Different applications of MF differ in the constraints that are sometimes imposed on the factorization. The most common form of MF is finding a low-rank approximation (unconstrained factorization) to a fully observed data matrix minimizing the sum-squared difference to it.

The resulting dot product, $h_i^T w_u$, captures the interaction between user u and item i. However, the full rating value is not just explained by this interaction and the user and item bias should also be taken into account. It is because part of the rating values is due to effects associated with either users or items, i.e biases, independent of any interactions.

By considering the user and item bias, the predicted rating is computed as follows [14]:

$$\hat{r}_{ui} = \mu + b_i + b_u + h_i^T w_u \qquad (1)$$

in which μ is the global average, b_i is the item bias and b_u is the user bias. The major challenge is computing the mapping of each item and user to factor vectors $h_i, w_u \in R^k$. The mapping is done by minimizing the following squared error [14]:

$$Opt(S, W, H) = \sum_{(u,i) \in S} (r_{ui} - \mu - b_u - b_i - h_i^T w_u)^2 + \qquad (2)$$

$$\lambda(\|h_i\|^2 + \|w_u\|^2 + b_u^2 + b_i^2)$$

in witch λ is the regularization factor, and S is the set of the (u, i) pairs for which r_{ui} is known, i.e the training set (S).

When a new user enters the recommender system, the prediction model should be updated to learn the new user latent features. As there are already a lot of users in the recommender system, training the whole of the model from scratch needs a lot of time. Therefore, we switch to online updating and develop the active learning algorithm based on that. Online updating means after a first training with all users, further retraining is only done for new users. For the online updating, we use the method introduced in [16]. In this method after getting a new rating from the new user, the user's latent features are initialized to a random setting and then are learned using all ratings of the new user.

4. PROPOSED METHOD

There is a trade-off between the information-theoretic value of a new rating and the likelihood of a user being able to provide a rating. A comprehensive method should take both criteria into account. However, in this paper we stick with the second criterion and leave the first one as the future

work. Therefore, the aim of the proposed active learning is to choose a query which is likely to be answered by the new user. As the process of active learning starts without any rating from the new user and ends up just with a few queries (e.g 10 queries), relying solely on the new user's answers does not provide *enough* information to find suitable queries specially when the number of items is large. An appropriate solution to address this problem is to link the new user to the training users and exploit available information from them. In this paper we use the characteristics of matrix factorization, as the prediction model of the recommender system, to make such connection.

When the training of the matrix factorization is done users and items are mapped into the latent space. Users (or items) with similar rating behaviors are mapped into the same region. We exploit this characteristic of matrix factorization to find similar users to the new user. Then the item which is most popular *among the similar users* is selected for the next query. Hopefully, the new user is able to provide a rating for an item which has already been rated by users with similar rating behavior. In this way, the drawback of the global most popular, which asks *static* queries regardless of the answers of the new user, is fixed while the new user is connected to the training users to exploit their information. The number of the similar users k is a hyper parameter. It should not be too large because it affects the scalability of this method in real time. Algorithm 1 describes the proposed active learning.

Algorithm 1 Proposed Active Learning

1: **loop** {repeats until N queries}
2:     find k-similar users
3:     find most popular item among the k-similar users
4:     ask the query
5:     **if** user provides a rating **then**
6:         update the new user features according to [16]
7:     **end if**
8: **end loop**

The accuracy of the proposed method heavily depends on how good the similar users are found. If the computed similar users are accurate, then approximating the rating behavior of the new user using the similar users is also accurate. Otherwise the performance is affected. In order to find similar users cosine similarity is used which is reasonable because in matrix factorization the ratings are predicted by the inner product of user and item features.

5. EXPERIMENTAL RESULT

In this section, we examine experimentally the performance of the proposed method.

5.1 Experimental Setup

The main challenge in applying active learning for recommender systems is that users are not willing to answer many queries in order to rate the queried items. For this reason, we report the performance of all examined methods in terms of prediction error (RMSE) versus the number of queried items, which is simply denoted as the *number of queries*. For the new user i, the RMSE is computed as follows:

$$RMSE = \sqrt{\frac{1}{N} \sum_{u,i \in M_u} (r_{ui} - \hat{r}_{ui})^2} \qquad (3)$$

where M_u is the set of the test items of user u, $\hat{r}_{ui}$ is the predicted rating (by MF) of user u for item i, and r_{ui} is the true (actual) rating. Thus, we examine the problem of selecting at each step, the item for which each new user u will be queried to provide a rating. The item has to be selected in order to minimize the $RMSE$ based on the MF model.

Non-myopic active learning [10] and most popular item selection,i.e item which has received most ratings from training users, are used as the baseline. We use 50 percent of the Netflix dataset in our experiments. The dataset was randomly split into training and test sets, containing 70% and 30% respectively. Each test user is considered as a new user. 30% of the ratings of each test user is separated to compute the error (test data) and the queries are selected from the remaining items (pool data). In our experiment, 10 queries are asked from each new user.

5.2 Results

Figure 1 illustrates the comparison between the proposed method, non-myopic active learning [10], and most popular selection in terms of RMSE as a function of the number of queried items.
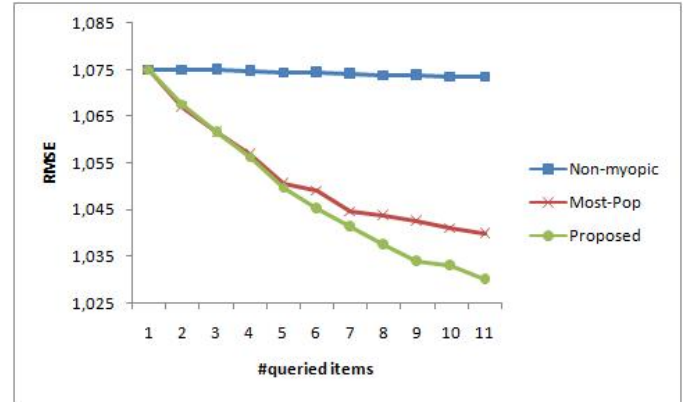

Figure 1: RMSE results of the proposed active learning, non-myopic and most popular versus the number of queries

In the first queries, the most popular method competes with the proposed method. This happens because during the first queries, a little knowledge about the rating behavior of the new user has been revealed. Therefore, it is difficult to find *real* similar users. But as more ratings are provided by the new user, the similar users become closer to the real similar users which consequently increases the probability that the new user provides an answer to the queried item. Another interesting observation of figure 1 is that the non-myopic method does not work at all. This was expected to happen because this method only aims to choose a query which reduces the test error and supposes that the new user always is able to provide a rating to the queried item. However, in this paper we have relaxed this assumption, meaning

that the queried item might not get a ratings. Obviously, this assumption has a significant effect on the performance of active learning methods.

Table 1: The average number of ratings received from the new user after 10 queries

Proposed	2.6468
Most Pop	2.6374
Non-myopic	0.0007

The performance of the active learning methods depends on the number of provided ratings by the new user. The more the ratings, the better the performance. Table 1 shows the average number ratings provided by a new user in all methods. The proposed method receives more ratings compared to most popular while the non-myopic method receives almost no ratings.

6. CONCLUSIONS

In this paper we proposed a new active learning method for the new user problem in recommender systems. It improves the most popular selection strategy using the characteristics of matrix factorization. First it finds similar users to the new user in the latent space and then selects item which is most popular *among the similar users.* Hopefully, the new user is able to provide a rating for an item which has already been rated by users with similar rating behavior. In this way, the drawback of the global most popular, which asks *static* queries regardless of the answers of the new user, is fixed while the new user is connected to the training users to exploit their information.

As the future work, we plan develop a criterion for informative queries and then combine it with the proposed active learning in the way that the trade-off between both criteria is met.

7. ACKNOWLEDGMENTS

8. 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, 2005.

[2] C. Boutilier, R. S. Zemel, and B. Marlin. Active collaborative filtering. In *Conference on Uncertainty in Artificial Intelligence(UAI)*, 2003.

[3] R. Burke. Hybrid recommender systems. *User Modeling and User Adapted Interaction*, 12(4):331–370, 2002.

[4] N. Golbandi, Y. Koren, and R. Lempel. Adaptive bootstrapping of recommender systems using decision trees. In *WSDM*, pages 595–604. ACM, 2011.

[5] A. S. Harpale and Y. Yang. Personalized active learning for collaborative filtering. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 91–98. ACM, 2008.

[6] T. Hofmann. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrievall*, pages 259–266. ACM, 2003.

[7] T. Hofmann and J. Puzicha. Latent class models for collaborative filtering. In *International Joint Conference on Artificial Intelligence*, pages 688–693. Morgan Kaufmann Publishers Inc., 1999.

[8] R. Jin and L. Si. A bayesian approach toward active learning for collaborative filtering. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, 2004.

[9] R. Karimi, C. Freudenthaler, A. Nanopoulos, and L. Schmidt-Thiem. Active learning for aspect model in recommender systems. In *IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*. IEEE, 2011.

[10] R. Karimi, C. Freudenthaler, A. Nanopoulos, and L. Schmidt-Thiemee. Non-myopic active learning for recommender systems based on matrix factorization. In *IEEE Information Reuse and Integration (IRI)*. IEEE, 2011.

[11] R. Karimi, C. Freudenthaler, A. Nanopoulos, and L. Schmidt-Thiemee. Towards optimal active learning for matrix factorization in recommender systems. In *23th IEEE International Conference on Tools With Artificial Intelligence (ICTAI)*. IEEE, 2011.

[12] A. Kohrs and B. Merialdo. Improving collaborative filtering for new users by smart object selection. In *International Conference on Media Features (ICMF)*, 2001.

[13] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. GroupLens: Applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77–87, 1997.

[14] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42:30–37, 2009.

[15] A. M. Rashid, I. Albert, D. Cosley, S. K. Lam, S. M. McNee, J. A. Konstan, and J. Riedl. Getting to know you: Learning new user preferences in recommender systems. In *International Conference on Intelligent User Interfaces (IUI)*, pages 127–134. ACM Press, 2002.

[16] S. Rendle and L. Schmidt-Thieme. Online-updating regularized kernel matrix factorization models for large-scale recommender systems. In *ACM Conference on Recommender Systems (RecSys)*, pages 251–258. ACM, 2008.

[17] S. Tong and D. Koller. Active learning for parameter estimation in bayesian networks. In *Advances in Neural Information Processing Systems(NIPS)*, 2000.

[18] K. Zhou, S.-H. Yang, and H. Zha. Functional matrix factorizations for cold-start recommendation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, pages 315–324. ACM, 2011.