Non-myopic Active Learning for Recommender Systems Based on Matrix Factorization

Rasoul Karimi, Christoph Freudenthaler, Alexandros Nanopoulos, Lars Schmidt-Thieme
Information Systems and Machine Learning Lab (ISMLL)
Samelsonplatz 1, University of Hildesheim, D-31141 Hildesheim, Germany
Email:[karimi,freudenthaler,nanopoulos,schmidt-thieme]@uni-hildesheim.ismll.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 ratings. In this paper, we consider the new user problem as an optimization problem and propose a non-myopic active learning method to select items to be queried from the new user. The proposed method is based on Matrix Factorization (MF) which is a strong prediction model for recommender systems. First, the proposed method explores the latent space to get closer to the optimal new user parameters. Then, it exploits the learned parameters and slightly adjusts them. The results show that beside improving the accuracy of recommendation, MF approach also results in drastically reduced user waiting times, i.e., the time that the users wait before being asked a new query. Therefore, it is an ideal choice for using active learning in real-world applications of recommender systems.**

I. Introduction

Recommender systems help web users to address information overload in a large space of possible options. Collaborative filtering is a traditional technique for recommender systems that has widely been applied [1]. 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 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 [2]. Model-based collaborative techniques provide recommendations by estimating parameters of statistical models for user ratings [3]. Nevertheless, recent research (especially as has been demonstrated during the Netflix challenge[1]) indicates that matrix factorization (MF) is a superior model-based prediction compared to memory-based approaches and other types of model-based algorithms [3].

Evidently, the performance of collaborative filtering depends on the number of ratings that the users provide. It is because the system is not able to provide a useful recommendation without knowing the user preferences. Providing ratings to items is the common way for users to reveal their preferences. Usually, users give ratings to items when they buy them. However, when a new user enters a recommender system, the system does not know anything about him/her because he/she has not provided any ratings. A simple and effective way to overcome this problem, is to pose to new users queries about their preference (e.g., ratings) for selected items. Nevertheless, the selection of queried items must take into consideration that users are not willing to answer a lot of such queries [4]. This fact impacts negatively on the quality of recommendations. To address this situation, *active learning* methods have been proposed to acquire those ratings from users, that will help most in determining their interests [5], [6].

In this paper, we focus on MF as a powerful model-based recommendation method and propose a novel method for applying active learning in recommender systems. The proposed method introduces, to our knowledge, for the first time a *general* active learning approach based on matrix factorization specially designed for the *new user problem* in recommender systems.

The rest of this paper is organized as follows: in section 2, the related work is reviewed. Then MF is explained in section 3. In section 4, the proposed active learning algorithm is described, followed by the simulation result in section section 5. Finally, the conclusion will be stated.

II. Related Work

Active Learning in the context of the new user problem was introduced by Kohrs and Merialdo [7]. They suggested a method based on nearest-neighbor collaborative filtering which uses entropy and variance as the loss function to identify the best item to query. Mamunur et al. [4] expanded this work by considering the popularity of items and also personalizing the item selection for each individual user. Boutilier et al. [8] applied 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 [5] developed a new active learning algorithm based on Aspect Model (AM) which is similar to the active learning approach towards parameter estimation in Bayesian networks [9]. This method uses the entropy of the model as loss function. However, they do not directly minimize the entropy loss function because the current model could be far from the true model and relying only on the current model could be misleading. Therefore, they use a Bayesian network to take into account the reliability of the current model. The

[1] www.netflixprize.com

proposed Bayesian approach is complex and intractable for real applications. Harpale and Yang [6] extended [5] by relaxing this assumption that a user can provide a rating for any queried item. This, personalized active learning queries for items which are likely to be rated by the user. Karimi et. al. [10] 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 [11] compared AM with MF and showed that MF is more suitable for applying active learning in recommender systems. Rish et. al. [12] proposed a new active learning method for Maximum Marigin Matrix Factorization (MMMF). This method exploits the training algorithm of MMMF which is similar to Support Vector Machines (SVM). Different from our method in this paper, it is not considered for the new user problem and aims to improve the overall performance of the recommender system. Also, it relies on only one form of matrix factorization (MMMF) and is not general. Although MMMF works well for classification, but it is not optimal for rating prediction [13]. Therefore, in order to develop active learning based on MF, other implementations of matrix factorization sould be taken into account. Finally, as MMMF is slow, the proposed method performs experiments on a small subset of the MovieLens dataset including only 50 users and 50 items.Therefore, the scalability of this method is a big issue that makes it inapplicable for recommender systems.

III. Matrix Factorization

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 following [3]:

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

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 [3]:

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

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

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 MF is applied in a specific data set, the predicted ratings should be in the range of the minimum rating and maximum rating of the dataset. However, sometimes this does not happen and we have to explicitly clip them. Although the effect of clipping is neglectable for the whole of the users, but it is significant for the new user. It is because the new user parameters are learned just by a few ratings and sometimes the predicted ratings are smaller than the minimum rating. To solve this problem we use the sigmiodal function to automatically truncate the predicted rating to the range of minimum and maximum ratings. Therefore, the predicted ratings are computed as following:

$$\hat{r}_{ui} = MinRating + \frac{(MaxRating - MinRating)}{1 + e^{-(\mu + b_i + b_u + h_i^T w_u)}} \tag{3}$$

IV. Proposed Active Learning Method for Matrix Factorization

We believe that solving the new user problem in recommender systems is an optimization task. The goal of this optimization task is to learn the new user preferences by querying him/her the most informative items. For that, the existing optimization solutions should be taken into account. Exploration-Exploitation dilemma is a common framework to deal with such tasks. It starts with exploring the environment aiming to learn better solutions that have not been discovered yet. Then it exploits the learned knowledge to improve it as much as possible.

Exploration-Exploitation dilemma is a non-myopic approach. It means it assumes that the next step is not the final step to solve the optimization task and further steps are still required. Therefore, in each step, not only the immediate effect of the action is important, but also its long term effect becomes important. This approach has already been used in the active learning literature for the classification problem [14].

The proposed method is a non-myopic active learning method based on the Exploration-Exploitation dilemma. First, it explores the latent space to get closer to the optimal new user parameters. The optimal new user parameters ideally show the influence of factors on user preferences. Then, it exploits the learned parameters and slightly adjusts them. Both exploration and exploitation algorithms rely on online updating of MF.

Online updating is important because there are already a lot of training users in the system and retraining the whole of MF after getting a new rating needs a time which is not acceptable for the interactive scenario of active learning. For this purpose, we exploit an online updating technique for MF introduced in [15]. In this method, after getting a new rating from the new user, the user latent parameters are restarted to random and then learned again using all ratings. The experimental results show that the accuracy of this online updating method is same as full retraining [15].

In the following sections, the exploration and exploitation algorithms are described. Then the solution for combining them into one method is explained.

A. Exploration Algorithm

As the new user parameters are computed by a few ratings, they are inaccurate and significantly different from the optimal parameters[2]. Therefore, in order to improve the accuracy, we have to drastically change them. This is called exploration. The exploration algorithm aims to select an item that retraining the user parameters with the provided rating will change the user parameters as much as possible.

The new user parameters are computed as follows [15]:

$$w_{u,f} \leftarrow w_{u,f} - \alpha \frac{\partial}{\partial w_{u,f}} Opt(r_{u,i}, W, H) \tag{4}$$

in which f is the latent dimension, α is the learning rate, and $r_{u,i}$ is the provided rating by the new user. As it is clear, the user parameters are changed as much as possible when the gradient is maximum. Therefore, we just need to find an item with the maximum gradient.

The gradient is propositional to the product of item parameters in the latent space and the difference between the predicted rating and the true rating (prediction error) [15]:

$$\frac{\partial Opt(\{r_{u,i}\}, W, H)}{\partial w_{u,f}} \propto (\hat{r}_{u,i} - r_{u,i}) \cdot h_{i,f} \tag{5}$$

As the true rating is unknown, it is not possible to compute the prediction error. However, we already know that most of the ratings in the datasets like MovieLens are above 3. It is because users usually provide ratings for movies that they like. Therefore, it is expected that the prediction error would be large for items with small predicted ratings since the actual ratings likely is different from the predicted rating. Therefore, the exploration algorithm selects item with the smallest predicted rating. This method is called MinRating.

Another possibility to increase the gradient is to select an item with large parameters. However, the gained gradient in this method is not as large as MinRating. In the experimental result section we will return to this point.

[2]That is why active learning is used to get additional ratings from the new user.

B. Exploitation Algorithm

So far, we were exploring the latent space by selecting an item that brings maximum change to the new user parameters. This is a good strategy in the first queries because, as we already mentioned, the user parameters are not accurate. However, as the new user parameters are retrained with more ratings provided by him/her, the accuracy of the estimated parameters also improves. Therefore, it is not necessary to change the user parameters as much as possible. Instead, the amount of change should be decreased so the user parameters be slightly adjust to the optimal parameters. Again based on equ. 5, one could find that the minimum change in the new user parameters happens when the gradient in minimum.

In order to minimize the gradient, the prediction error and item parameters should be minimized. As the value of the prediction error is larger, it makes sense to focus on this part similar the situation that the goal was to maximize the gradient. It means to select an item with the largest predicted rating. Hopefully, the actual rating would also be large and so the prediction error would be small. However, this is not a good strategy. It is because if the actual rating is not as large as the predicted rating, the prediction error would be large. This increases the gradient and consequently the changes of the user parameters. The safer strategy is to focus on the item parameters and minimize them. The values of the item parameters are already ready and it is possible to select the item with the smallest latent parameters. As the gradient is the multiplication of the prediction error and the item parameters, if the item parameters are close enough to zero, it neutralizes the large value of the prediction error. Therefore, the gradient is not changed a lot. The Euclidean norm of item parameters is used to determine overall how small the item parameters are. This method is called MinNorm.

C. Combining Exploration and Exploitation Algorithms

Now we have to come up with a solution to combine exploration and exploitation algorithms. The solution should be in a way that pays more attention to exploration in the first queries and as the user provides more ratings, the weight of exploitation increases. The proposed method does the trade-off between exploration and exploitation by making two ranking lists. In the first list, the items are ranked based on the predicted ratings. The smallest the ratings, the lowest the ranking. And in the second list, the items are ranked based on the norm of parameters. The smallest the norm, the lower the rank. Finally the total rank of item i is computed as follows:

$$total\_rank(i) = rating\_rank(i) \cdot (1-w) + norm\_rank(i) \cdot w \tag{6}$$

where w is the wight of exploitation :

$$w = current\_query\_number - 1/allowed\_query\_numbers \tag{7}$$

In the first query, w is 0. It means that the item is selected only based on the exploration algorithm. As more ratings are provided by the new user, the effect of exploitation to select the query increases. In each step, the item with the minimum total rank is selected for query because it satisfies both criteria for exploration and exploitation respect to their importance.

V. Experimental Results

In this section, we examine experimentally the performance of the proposed method. At first random selection in MF is compared against random selection in AM. This comparison will show the advantages of the MF as the underlying prediction model, which verifies the first contribution of this paper. Then, the proposed criteria for active learning are compared. The aim of this comparison is to show the benefits from the non-myopic active learning for selecting the queried items. That is, we need to combine the power of the MF model with the proposed selection criteria, in order to attain the described improvements.

The experiment is run 10 times where the evaluation folds including test users and test items are randomly selected. We believe that in order to get a valid result for applying active learning in recommender systems, k-fold evaluation is necessary and the result of one fold does not reflect the correct solution. This point was ignored in [5], [6] because they did not use online updating and also the Bayesian approach is too time-consuming even for one fold.

A. Data Set

We use the MovieLens[3] dataset in our experiments. MovieLens contains 943 users and 1682 items. The dataset was randomly split into training and test sets. In MovieLens there are 343 training users (the same number used in [6]). The test data contains only users who have rated more than 33 items which already appeared in the training data. Each new user is considered as a new user and the preliminary model is built using three random initial ratings. The remaining items are split into the pool set and the test set. Active learning algorithm selects items query from the pool set. In this paper we assume that the new user will always be able to rate the items presented by the active learning. We will leave the personalization issue as the future work

B. Results

First, we compare the accuracy of performing active learning based on MF with AM. The objective is to show that MF is a better prediction model to be used for developing active learning algorithms. For this reason, in order to have a fair comparison regarding the selection of the queried items and focus only on the prediction model, we simply apply random selection of the queried items for both MF and AM. Fig. 1 depicts the resulting Mean Absolute Error (MAE) as a function of the number of queried items. Evidently, MF outperforms AM, indicating its superiority as prediction model.

The hyper parameters af MF and AM are given in tables I and II respectively. As the number of the provided ratings of the new user is a few, the learning rate should also be reduced when the new user parameters are retrained with onilne updating. In our experiment, the best retraining α was 0.001.

[3]www.grouplens.org/system/files/ml-data$_0$.zip

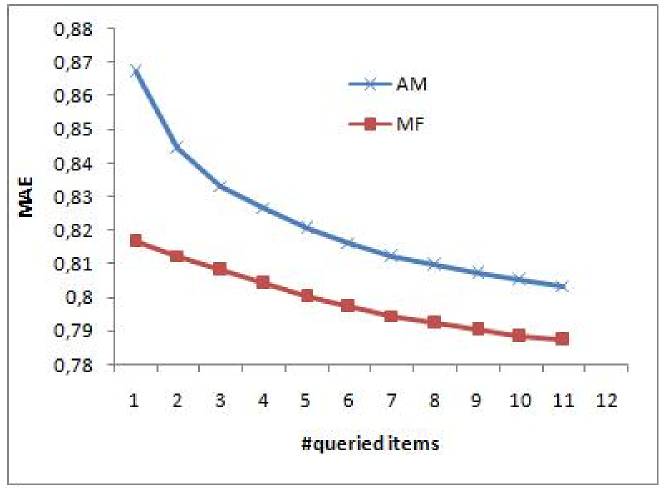

Fig. 1. MAE results of random active learning for MF and AM

TABLE I
MF hyper parameters

α	λ	bias reg.	dimension	#training iterations
0.01	0.15	0	10	100

TABLE II
AM hyper parameters

beta(β)	eta(η)	dimension	#training iterations
0.77	1	10	100

In addition to accuracy, applying active learning for new user problem in recommender systems asks for a further criterion. Preference elicitation of the new user is an interactive scenario and long time interruptions cause the user to leave the conversation. Therefore, active learning method should also be fast and don't require long user waiting times, i.e., the time that the users wait before being asked a new query. As it was already mentioned, the retraining of MF and AM is done using online updating. The retraining time is equivalent to the time that the new user waits before being asked a new query (user waiting time).

Based on the random selection, the average of user waiting time per each query in MF is 3.9 seconds compare to 44.5 seconds in AM. This difference comes from different training algorithms in MF and AM. While the common method for training MF is stochastic gradient descent, AM is based on Expectation Maximization. The details of comparison between these two models is given in [11].

As it was mentioned in section III, using a sigmoidal function to truncate the predicted ratings to be in the range of the minimum and maximum ratings improves the accuracy. In our experiments, the initial error of the new user after three random ratings is 0.816 when the sigmoidal function is used. But without the sigmoidal function it is 0.835.

We now move on to examine the advantage of the proposed criteria for selecting the queried items. These criteria include smallest predicted rating (MinRating), smallest item norm (MinNorm), largest item norm (MaxNorm), combination of MinRating and MinNorm (Non-myopic), and random selection. Fig. 2 illustrates the comparison between these five methods in terms of MAE as a function of the number of queried items.

Clearly, the non-myopic method compares favorably against other methods. This happens because it takes the advantage of both exploration and exploitation algorithms. In the first queries that exploration is necessary, MinRating competes with non-myopic. However, The MinRating's performance does not continue because it never exploits. To clearly understand the role of exploration and exploitation, we need to look at the gradient. Table III depicts the gradient after each queried item for all methods(in total 10 queried items). The gradient of MinRating is larger than other methods almost in all queries. This is useful at the beginning because it leads to exploration. But it does not help as more ratings are provided by the new user. Another interesting result in table III is about MinNorm. The gradient of MinNorm is smaller than other methods and so, its accuracy is much worse than random. This shows that exploitation alone does not work and it should be combined with the exploration algorithm. Finally, the gradient of MinRating is larger than MaxNorm. It shows that, as we expected, to maximize the the gradient, we need to focus on the prediction error than item parameters. This leads to a better accuracy specially in the first queries.

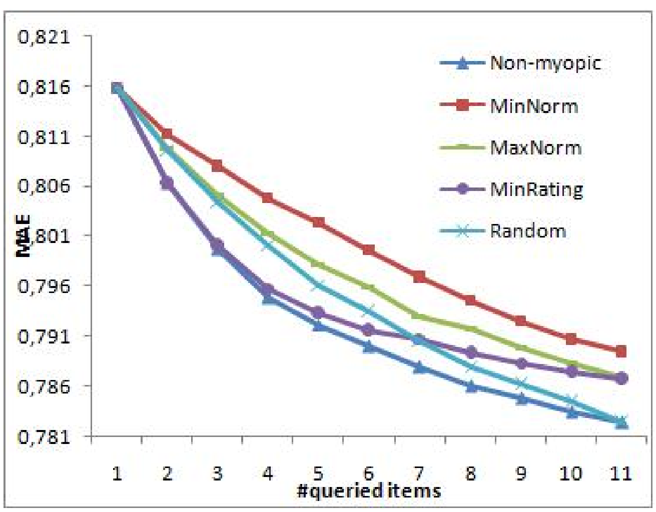

Fig. 2. MAE results of the proposed criteria for active learning

VI. Conclusion

In this paper, we proposed a novel active learning method for recommender systems that is based on Matrix factorization (MF). Our motivation stems from the fact that, in recent research, MF has been demonstrated as a powerful prediction model for recommender systems. We performed a detailed experimental evaluation which shows the power of MF and the proposed method.

TABLE III
The gradient after each query

# query	Non-myopic	MinRating	MinNorm	MaxNorm	Random
1	-0.165	-0.177	-0.060	-0.061	-0.065
2	-0.177	-0.207	-0.063	-0.073	-0.083
3	-0.184	-0.173	-0.022	-0.069	-0.051
4	-0.160	-0.163	-0.013	-0.084	-0.083
5	-0.120	-0.151	-0.047	-0.028	-0.023
6	-0.122	-0.177	-0.029	-0.023	-0.060
7	-0.117	-0.098	-0.001	-0.046	-0.058
8	-0.037	-0.092	-0.018	-0.051	-0.021
9	-0.054	-0.068	0.026	0.042	-0.024
10	0.050	-0.098	-0.013	-0.027	-0.046

Acknowledgment This work is co-funded by the European Regional Development Fund project REMIX under the grant agreement no. 80115106.

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*, p. 734, 2005.
[2] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, J. Riedl, and H. Volume, "Applying collaborative filtering to usenet news," *Commun. ACM*, p. 77, 1997.
[3] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Computer*, vol. 42, p. 30, 2009.
[4] 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," *International Conference on Intelligent User Interfaces(IUI)*, p. 127, 2002.
[5] R. Jin and L. Si, "Bayesian approach toward active learning for collaborative filtering," *Uncertainty in Artificial Intelligence(UAI)*, 2004.
[6] A. S. Harpale and Y. Yang, "Personalized active learning for collaborative filtering," *ACM Conference on Research and Development in Information Retrieval(SIGIR)*, 2008.
[7] A. Kohrs and B. Merialdo, "Improving collaborative filtering for new users by smart object selection," *International Conference on Media Features (ICMF)*, 2001.
[8] C. Boutilier, R. S. Zemel, and B. Marlin, "Active collaborative filtering," *Conference on Uncertainty in Artificial Intelligence(UAI)*, 2003.
[9] S. Tong and D. Koller, "Active learning for parameter estimation in bayesian networks," *Advances in Neural Information Processing Systems(NIPS)*, 2000.
[10] R. Karimi, C. Freudenthaler, A. Nanopoulos, and L. Schmidt-Thieme, "Active learning for aspect model in recommender systems," *IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, 2011.
[11] R. Karimi, C. Freudenthaler, A. Nanopoulos, and L. Schmidt-thieme, "Selecting a suitable prediction model for applying active learning in recommender systems," in *34th Annual Conference of the German Classification Society (GFKL)*. Springer, 2010.
[12] I. Rish and G. Tesauro, "Active collaborative prediction with maximum margin matrix factorization," *International Symposium on Artificial Intelligence and Mathematics*, 2008.
[13] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," *25th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2009.
[14] T. Osugi, D. Kun, and S. Scott, "Balancing exploration and exploitation: A new algorithm for active machine learning," *IEEE International Conference on Data Mining (ICDM)*, 2005.
[15] S. Rendle and L. Schmidt-Thieme, "Online-updating regularized kernel matrix factorization models for large-scale recommender systems," *ACM Conference on Recommender Systems (RecSys)*, 2008.