Towards Optimal Active Learning for Matrix Factorization in Recommender Systems

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 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. The optimal active learning selects a query that directly optimizes the expected error for the test data. This approach is applicable for prediction models in which this question can be answered in closed-form given the distribution of test data is known. Unfortunately, there are many tasks and models for which the optimal selection cannot efficiently be found in closed-form. Therefore, most of the active learning methods optimize different, non-optimal criteria, such as uncertainty. Nevertheless, in this paper we exploit the characteristics of matrix factorization, which leads to a closed-form solution and by being inspired from existing optimal active learning for the regression task, develop a method that approximates the optimal solution for recommender systems. Our results demonstrate that the proposed method improves the prediction accuracy of MF.**

***Keywords*-Recommender System; Matrix Factorization; Active Learning**

I. Introduction

Recommender systems help web users to address information overload in a large space of possible options [1]. 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 [2], [3], [4]. 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 [3]. Model-based collaborative techniques provide recommendations by estimating parameters of statistical models for user ratings. Matrix Factorization (MF) [5] and Aspect Model (AM) [6], [7] are two popular model-based methods. Nevertheless, recent research (especially as has been demonstrated during the Netflix challenge[1]) indicates that MF is a superior prediction model compared to other approaches [5].

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 [8], [9]. 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 [9], [8].

In this paper, we propose a novel method for applying active learning in recommender systems. Due to the rapidly increasing interest in MF as a powerful prediction model in recommender systems, the proposed method introduces an active learning approach designed to take into account the characteristics of MF in order to improve its accuracy. The proposed method is inspired from optimal active learning for regression problem. Assuming the distribution of the test data is known, it is possible to find the optimal active learning algorithm for specific regression models [10]. As MF is actually a regression problem, it makes sens to use the same approach for active learning in MF. Given the test items are known, we develop a method which approximates the optimal active learning for MF. It capitalizes on the updating mechanism of MF and allows us to formulate a new criterion for the selection of the queried items, in terms of reducing the expected prediction error. A detailed experimental evaluation is performed, whose results demonstrate the superiority of the proposed method. Our results provide insight into the effectiveness of the proposed criterion for selecting the

[1] www.netflixprize.com

queried items, as it compares favorably to methods that use MF but are based on simplistic criteria.

The rest of this paper is organized as follows: Section II the related work is reviewed. The proposed active learning algorithm is detailed in Section III, followed by the experimental results in Section IV. Finally, the conclusions are stated in Section V.

II. Related Work

In this section, we first summarize the related works on active learning, since several examined techniques are inspired by them. Next, we focus on the related works that applies active learning in recommender systems.

A. Related Works on Active Learning

Cohn et al. [10] describe how optimal data selection techniques can be applied to statistically-based learning algorithms like a mixture of Gaussians and locally weighted regression. The algorithm selects instances that, minimizes the expected error on test data. There are two main drawbacks for this method: at first, it assumes that the distribution of test data is known which is not almost the case. Second, it might find a query that is optimal for improving the prediction model but does not exist in the pool data. Therefore, knowing the answer of this query is not possible. Roy and McCallum [11] eliminated these two drawbacks by considering a limited pool data and then estimating the true test distribution from this pool. However the computation time of this method is high, especially when the size of the pool data is large. Moreover, the estimation of the test data distribution from the pool data might be inaccurate. Baram and et al. [12] proposed a framework based on multi-armed bandit algorithm to combine several active learning algorithms. The idea behind this framework is that the best active learning algorithm depends on the dataset and should be determined run time. In this framework all of the active learning algorithms have chances to select queries. After each query and updating the prediction model, the probability of the active learning algorithm, which had selected the query, is changed based on the gained improvement in the classifier. More improvement indicates the suitability of the algorithm and increases the probability and vice versa.

Osugi and et al. [13] proposed a lighter version of Baram et al. [12] that includes only two active learning algorithms. One algorithm selects examples which are close to decision boundary (exploitation) and the other algorithm selects examples that are far from the decision boundary (exploration). Nguyen and Smeulders [14] offer a framework to incorporate clustering into active learning. This method, which is based on logistic regression, chooses samples close to the classification boundary and samples which are cluster representatives. Furthermore, within the set of cluster representatives, it starts with the highest density clusters first. Poupart [15] proposed an idea to apply Reinforcement Learning (RL) for active learning problem. This method explicitly models the sequence of queries with a Markov Decision Process (MDP) and learns the best sequence of queries with RL. The introduced idea has been applied for patient treatment in a hospital that is different from pool-based active learning problems. In order to apply this idea for pool-based active learning problem, one should define state, action, and reward function in RL according to the characteristics of active learning which is non-trivial.

B. Related Works on Active Learning for Recommender System

Active learning, in the context of the new-user problem, was introduced by Kohrs and Merialdo [16]. 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. [4] expanded this work, by considering the popularity of items and also personalizing the item selection for each individual user. Boutilier et al. [17] apply the metric of expected value of utility to find the most informative item for 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 AM which is similar to applying active learning for parameter estimation in Bayesian networks [18]. This method uses the entropy of the model as the loss function. However, this work 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, this 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). Harpale and Yang [9] extended [8] by relaxing this assumption that a new user can provide a rating for any queried item. This approach personalizes active learning to the preferences of each new user as it queries only those items for which users are expected to provide a rating for. Karimi et. al. [19] 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 [20] compared AM with MF and showed that MF is more suitable for applying active learning in recommender systems because it is more accurate and faster. Karimi et. al [21] 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 parameters with the provided rating will change the parameters as much as possible. Its goal is to explore the latent space to get closer to the optimal parameters. Then, it exploits the learned parameters and slightly adjusts them.

Compared to all aforementioned methods which are based on heuristics, the criterion proposed in the current paper for selecting the queried items opts for minimizing directly the resulting expected error. This approach has already been studied in the classic active learning [10] and in this paper, we apply it for recommender systems. Regarding the aspect of personalization, our proposed method follows the majority of work in this domain and is based on a pre-specified pool of items that can be queried. However, we address this topic of personalization, i.e., of the dynamic formation of this pool, as our main point of future work, since MF models provide a good basis toward this direction.

III. Proposed Active Learning for Matrix Factorization

In this section, we first give a formal definition of the active learning problem. Then, we use it to define the proposed criterion for the active learning in recommender systems. Next, a comprehensive description of MF model is explained because the proposed method capitalizes on its characteristics. Finally, we explain in detail the proposed method. For brevity, we will describe the proposed method in terms of the new-user problem. However, a similar approach can be applied for the new-item problem as well, because MF models are usually symmetric.

A. The Relation Between the Problem and Classic Active Learning

Consider the problem of learning a binary classifier on a partially labeled database $D \subseteq X \times Y$ in which $\{x_1, ..., x_n\} \subset R^d$ are instances and $y \in \{1, -1\}$ are labels. The training data D is divided into labeled L and pool data P. First, the classifier $\hat{f} : X \to Y$ is trained with L. Then active learning selects N instances x from pool data P and asks their label $f(x)$ from an oracle. The goal of active learning is to select queries that will result in maximum gained accuracy after retraining the classifier with the queried instances.

In myopic active learning, it is supposed that the next query is the last query, so the instance which is *likely* to be the best one is selected. The selection is done based on an objective function (we call it selection function σ in this paper):

$$\sigma : X \times Y^X \to R$$

$$(x, \hat{f}) \longmapsto \sigma(x, \hat{f})$$

σ depends on the current classifier $\hat{f}$. The best query is one that minimizes σ:

$$x^* = \operatorname*{argmin}_{x \in P} \sigma(x, \hat{f})$$

Usually one query is selected in each iteration. This process repeats until N queries are selected. Although in this paper, we deal with MF as the prediction model which is different from classifiers, but the general principal of this formal definition of active learning holds and we will use it in the next section.

B. Problem definition

Given the new user u to whom we pose queries for selected items asking for a rating, the domain of all items is divided into: i) P_u, i.e., the pool of available items for posing queries (see discussion at the previous section regarding the notion of pool); M_u, i.e., the set of test items. At step l, user u will be asked the l-th query in order to give a rating for one of the items selected from P_u. As described, the total number of queries (max value of l) is in general small, usually not more than ten [9], [8]. The objective of active learning is to select those items from P_u that will help in optimizing the total prediction accuracy with which the underlying prediction model, MF in our case, predicts the ratings of the new user u. As the optimization of accuracy requires the definition of an error measure, we adopt for this purpose one of the most commonly applied error measure in recommender systems, which is the Mean Absolute Error (MAE). For the new user u, MAE can be expressed as follows:

$$MAE_u = \frac{1}{|M_u|} \sum_{i \in M_u} |r^*_{ui} - r_{ui}| \tag{1}$$

where 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 MAE_u based on the MF model.

C. Matrix Factorization in Recommender Systems

Matrix Factorization is the task of approximating the true, unobserved ratings-matrix R. The rows of R correspond to the users U and the columns to the items I. Thus the matrix has dimension $|U| \times |I|$. The predicted ratings $\hat{R}$ are the product of two feature matrices $W : |U| \times k$ and $H : |I| \times k$ where the u-th row w_u of W contains the k features that describe the u-th user and the i-th row h_i of H contains k corresponding features for the i-th item. 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. 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 common form of MF is finding a low-norm approximation (regularized factorization) to a fully observed data matrix minimizing the sum-squared difference to it.

The predicted rating of user u to item i ,$\hat{R}$, is the inner product of the user u features and item i features:

$$\hat{r}_{ui} = w_u h_i^T \tag{2}$$

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 [22]:

$$Opt(S, W, H) = \sum_{(u,i)\in S} (r_{ui} - w_u h_i^T)^2 + \lambda(\|h_i\|^2 + \|w_u\|^2) \tag{3}$$

where 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. The usual method to train MF is gradient descent [5]. This algorithm loops through all ratings in the training set and repeats the loop until Z iterations. The training could be stopped if the error on the training is smaller than δ.

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 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 online updating, we use the method introduced in [22]. In this method after getting a new rating for a new user, the user's latent features are initialized to a random setting and then learned using all ratings of the new user. The details of this method is described in Algorithm 1.

Algorithm 1 Online updating for new-user problem according to [22]

initialize $u^\star$-th row in W
repeat
loop {repeats until Stopping criteria met}
  for $r_{u,i} \in C(u^\star, .)$ **do**
    for $f \leftarrow 1, ..., k$ **do**
      $w_{u,f} \leftarrow w_{u,f} - \alpha \frac{\partial}{\partial w_{u,f}} Opt(S, W, H)$
    end for
  end for
end loop
return (W,H)

where $C(u^\star, .)$ is the training data of the new user, i.e only ratings provided by the new user.

After driving the gradient of function Opt, updating with respect to each item i can be performed as follows:

$$w_{uf}^{l+1} = w_{uf}^l - 2\alpha(r_{ui} - r_{ui}^*)h_{if}, \; \forall \text{ item } i \tag{4}$$

where α is the learning rate, w_{uf} is the f-th user parameter (coordinate of vector w_u in the latent space) after l queries, w_{uf}^{l+1} is the updated user parameter based on the previous value w_{uf}^l, and r_{ui}, r_{ui}^* are the predicted and true rating, respectively of user u for item i.

D. Proposed Method

The aim of the active learning is to improve the accuracy of the prediction model on test data. Therefore, the best strategy is to select a query that directly optimizes the expected error for the test data. This approach is applicable for prediction models in which this question can be answered in closed-form [10]. Unfortunately, there are many tasks and models for which the optimal selection cannot efficiently be found in closed-form. Therefore, most of the active learning methods optimize different, non-optimal criteria, such as uncertainty [23]. Nevertheless, in the following we exploit the characteristics of matrix factorization, which leads to a closed-form solution (see Proposition 1) and by being inspired from [10] develop a method that approximates the optimal solution. We have to refer to Section III-B, where optimality is defined in terms of reducing MAE on test data.

Proposition 1: The optimal selection of the queried item for a new user u is based on the following equation:

$$m_u^* \simeq \underset{m\in P_u}{\operatorname{argmin}} \sum_{i\in M_u} \left| r_{ui}^* - r_{ui} + 2\alpha(r_{um} - r_{um}^*) \sum_{f=1}^{k} h_{mf} h_{if} \right| \tag{5}$$

Proof. As described in Section III-B, our objective is to minimize the MAE of the new user u. Since, based on MF, the ratings are calculated according to Equation 2, Equation 1 can be rewritten as following:

$$MAE_u = \frac{1}{|M_u|} \sum_{i\in M_u} \left| \sum_{f=1}^{k} w_{uf}^* h_{if}^* - \sum_{f=1}^{k} w_{uf} h_{if} \right| \tag{6}$$

where w^* and h^* are the optimal user and item parameters in the latent space that ideally show how the latent factors influence the rating behavior of users and items. [2]

Equation 6 is the current test error before asking a new query. The goal of the active learning algorithm is to minimize it in the next step (after getting a new rating) as much as possible by querying the best item. As we already mentioned, for retraining we exploit the online updating technique [22]. Therefore, only the user parameters w_u change in the next step and item parameters h_i are fixed. Assuming that $l \geq 0$ items have been selected so far for user u, let w_{uf}^l denote the new user parameters (i.e., after the selection of l items) and define the test error in step l as :

$$MAE_u{}^l = \frac{1}{|M_u|} \sum_{i\in M_u} \left| \sum_{f=1}^{k} w_{uf}^* h_{if}^* - \sum_{f=1}^{k} w_{uf}^l h_{if} \right| \tag{7}$$

[2]As it is not possible to obtain w^* and h^* , we will later provide a simple and effective way to estimate them.

Therefore it follows that, the optimal query m^* is one that produces the minimum ${MAE_u}^{l+1}$ by providing new user parameters w_{uf}^{l+1}:

$$m_u^* = \underset{m \in P_u}{\operatorname{argmin}} \frac{1}{|M_u|} \sum_{i \in M_u} \left| \sum_{f=1}^{k} w_{uf}^* h_{if}^* - \sum_{f=1}^{k} w_{uf}^{l+1} h_{if} \right|. \tag{8}$$

Since $|M_u|$ is a constant number, it can be dropped from the previous equation, giving the following:

$$m_u^* = \underset{m \in P_u}{\operatorname{argmin}} \sum_{i \in M_u} \left| \sum_{f=1}^{k} w_{uf}^* h_{if}^* - \sum_{f=1}^{k} w_{uf}^{l+1} h_{if} \right| \tag{9}$$

According to the law of consistent estimator in statistics [24], item parameters converge asymptotically to the true parameters, when there is infinity training data points. Typically, however, the convergence to the true parameters is reached earlier, since the difference – for large enough training sample sizes – between current and true item parameters is negligible. As the item parameters have been trained with training data including adequate number of ratings for each item, the current item parameters approximate well the true parameters. Therefore, we can substitute h_{if}^* with h_{if}. Followed by a simple algebraic manipulation we will have:

$$m_u^* \simeq \underset{m \in P_u}{\operatorname{argmin}} \sum_{i \in M_u} \left| \sum_{f=1}^{k} (w_{uf}^* - w_{uf}^{l+1}) h_{if} \right| \tag{10}$$

According to Equation 10, the optimal item to be selected, is the one that will cause the updated user parameters w_{uf}^{l+1}, i.e., after the selected item ($l+1$-th) will be queried, to be as close as possible to the true user parameter w_{uf}^*. This is exactly what we already expected. The test error reduces as the accuracy of the estimation of the true user parameters improves.

Based on the Equation 4, we can express Equation 10 as follows:

$$m_u^* \simeq \underset{m \in P_u}{\operatorname{argmin}} \left(\sum_{i \in M_u} \left| \sum_{f=1}^{k} \Big(w_{uf}^* - w_{uf}^l + 2\alpha (r_{um} - r_{um}^*) h_{mf} \Big) h_{if} \right| \right) \tag{11}$$

and with simple algebraic manipulations as follows:

$$m_u^* \simeq \underset{m \in P_u}{\operatorname{argmin}} \left(\sum_{i \in M_u} \left| \sum_{f=1}^{k} (w_{uf}^* - w_{uf}^l) h_{if} + 2\alpha (r_{um} - r_{um}^*) h_{mf} h_{if} \right| \right) \tag{12}$$

The inner product between w_u^l and h_i is equal to the predicted (by MF) rating, r_{ui} of user u for the test item i. Moreover, the inner product between w_u^* (the true user parameters of u) and the item parameters h_i for test item i, corresponds to the true rating. Based on the above, we result with the following equation:

$$m_u^* \simeq \underset{m \in P_u}{\operatorname{argmin}} \sum_{i \in M_u} \left| r_{ui}^* - r_{ui} + 2\alpha (r_{um} - r_{um}^*) \sum_{f=1}^{k} h_{mf} h_{if} \right| \tag{13}$$

□

It is worth to compare Equation 13 with Equation 1. The only difference in Equation 13 is the additional term $2\alpha(r_{um} - r_{um}^*) \sum_{f=1}^{k} h_{mf} h_{if}$. This term, actually, predicts how retraining the new user parameters with the item m can remove the test error of the item i which is $r_{ui}^* - r_{ui}$. This is exactly what we are looking for. The aim of the proposed active learning is to decrease the test error which is the summation over all errors of the test items. Therefore, it is reasonable reach this term in the final equation. Another interesting point in Equation 13 is that the difference between the predicted rating r_{um} and the true rating r_{um}^* is weighted according to the inner product between h_m and h_i. This means that if the pool item m is close to the all test items i (i.e., small value of their inner product), then it is considered as more representative.

Equation 13 requires the knowledge of the actual rating r_{ui}^* to compute the test error $r_{ui}^* - r_{ui}$. However, it is not possible to obtain r_{ui}^* since the rating of test data is unknown. To solve this problem, the proposed active learning method optimizes the query selection based on the worse test error. It means that given the test error of item i is maximum, which item is the best item for query. The test error of item i is maximum if the actual rating r_{ui}^* will be different from what is expected to be. In the datasets like MovieLens and Netflix, the most of the ratings are above 3 (in Movielens 3.8 and in Netflix 3.6). Therefore, in order to find the upper-bound for test error we substitute r_{ui}^* with 1. This approximation hopefully gives the worse test error and allows us to develop a robust active learning. In this way, we are cautious and conservative about the current predicted ratings r_{ui} and rely the active learning method on the upper-bound test error. Given the learned model of the new user is not accurate (that is why we need to ask additional ratings), this simplification is reasonable and realistic.

Moreover, Equation 13 also requires the knowledge of actual pool item r_{ui}^*. However, it is not possible to obtain r_{um}^* because the user u has to be first asked before providing a rating for them. Following a simple but effective approach, r_{um}^* can be approximated by $\bar{R}_m$ which denotes the average rating of all training users for the item m.

This simplifications leads to the following criterion for item selection:

$$m_u^* \simeq \operatorname*{argmin}_{m \in P_u} \sum_{i \in M_u} \left| 1 - r_{ui} + 2\alpha(r_{um} - \bar{R}_m) \sum_{f=1}^{k} h_{mf} h_{if} \right| \tag{14}$$

Being based on Proposition 1, the criterion of Equation 14 is, therefore, toward an optimal item selection. Evidently, the effectives of the applied approximation depends on the quality of the approximation of r^*_{um}. Our experimental results in the following section will indicate the effectiveness of the above proposed criterion. Nevertheless, we address the issue of investigating additional approximations for r^*_{um} as our main topic of future work.

IV. Experimental Result

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

A. Experimental set up

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 (MAE) versus the number of queried items, which is simply denoted as the *number of queries*. Non-myopic active learning [21], and random selection are used as the baseline. We do not compare against active learning algorithms based on AM because active learning based on MF is more accurate and faster than active learning based on AM [20]. Furthermore, as the existing active learning algorithms based on AM [8], [9] use the Bayesian approach to find the best query, they are too slow and inapplicable for recommender systems.

The experiments are run 10 times where the evaluation folds including test users and test items are randomly selected. Please note that the necessary use of k-fold validation was not performed in several related works [8], [9], because these approaches are too time-consuming and, thus, cross validation would have required prohibitive long time for evaluation. The experiments are done using the best hyper parameters of MF which are shown in Table I

Table I
MF hyper parameters

α in training	α in retraining	λ	k	δ	Z
0.01	0.001	0.15	10	$1e^{-5}$	100

We use the MovieLens(100K)[3] dataset in our experiments. MovieLens contains 943 users and 1682 items. The dataset was randomly split into training and test sets. The training dataset consists of 343 users (the same number used in [9]) and the rest of users are in the test dataset. Each test user is considered as a new user. The latent features of the new user are initially trained with three random ratings. 20 rated items of each test user are separated to compute the error. The test items are not new item and already appeared in the training data. The remaining items are in the pool dataset, i.e the dataset that is used to select a query. For simplicity, we assume that the new user will always be able to rate the queried item. In our experiment, 10 queries are asked from each new user. Therefore, the pool dataset should contain at east 10 items which exist in the training data. Considering 10 queries and 20 test items, each test user has given ratings to at least 30 items.

[3]www.grouplens.org/system/files/ml-data0.zip

B. Results

Figure 1 illustrates the comparison between the proposed method, non-myopic active learning [21], and random selection in terms of MAE as a function of the number of queried items. The non-myopic method works well in the first queries but it finally converges to the random selection. This convergence also happens for active learning in AM [8]. Generally, this evidence holds for active learning methods aiming to improve the new user parameters using some heuristics. In the optimization theory, usually the heuristics provide a good performance only if the difference between current solution and optimal solution is high. At first, as the new user has provided a few ratings, the new user parameters are inaccurate and are far away from the optimal parameters. But as more ratings are provided by the new user, the accuracy of the estimated parameters also increases and the heuristic-based methods do not gain much improvement. However, the proposed method in this paper has a different approach. It aims to directly optimize the test error. That is why its performance continues and does not converge to the random selection. Therefore, if the new user is ready to provide more ratings, the proposed method can efficiently use them to improve the accuracy.

V. Conclusion

Active learning is a suitable methodology for recommender systems in order to elicit information about the preferences of users. It poses a number of queries to users, asking about preference information (e.g., ratings) about selected items. As users are not willing to answer many of such questions, the problem of selecting the queried items becomes critical. 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. Our approach includes an analytical investigation of the problem of optimal selection of queried items, i.e., those minimizing the expected error. Moreover, we proposed a practical

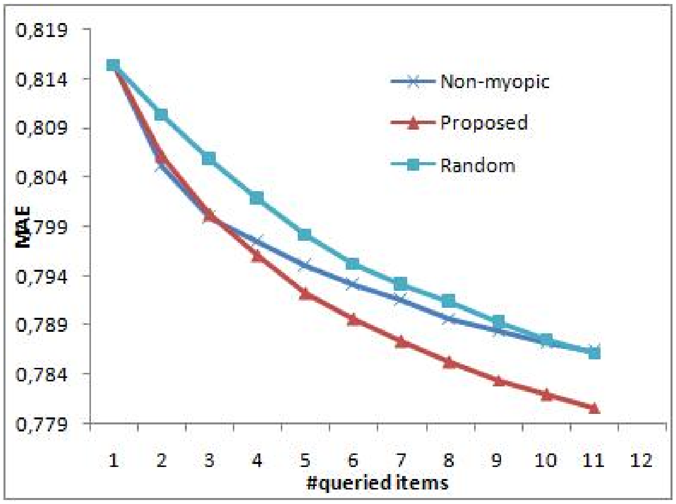

Figure 1. MAE results of the proposed active learning, non-myopic and random

criterion that stems from this analytical investigation, which applies approximation and can be used in real recommender systems, as it results to reduced error.

In our future work, we plan to extend the proposed approach to more sophisticated MF algorithms such as considering user and item biases in the rating prediction. Another future work is to examine different approximation schemes that allow for deriving additional selection criteria based on our analytical investigation.

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

REFERENCES

[1] R. Burke, "Hybrid recommender systems," *User Modeling and User Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002.

[2] 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*, vol. 17, no. 6, pp. 734–749, 2005.

[3] 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*, vol. 40, no. 3, pp. 77–87, 1997.

[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," in *International Conference on Intelligent User Interfaces (IUI)*. ACM Press, 2002, pp. 127–134.

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

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

[7] 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*. ACM, 2003, pp. 259–266.

[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] 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*. ACM, 2008, pp. 91–98.

[10] D. A. Cohn, G. Z., and M. Jordan, "Active learning with statistical models," in *Advances in Neural Information Processing Systems(NIPS)*, 1995.

[11] R. Nicholas and A. McCallum, "Toward optimal active learning through monte carlo estimation of error reduction," in *International Conference on Machine Learning (ICML)*, 2001.

[12] Y. Baram, R. El-Yaniv, K. Luz, and M. Warmuth, "Online choice of active learning algorithms," *Journal of Machine Learning Research*, vol. 5, pp. 255–291, 2004.

[13] T. Osugi, D. Kun, and S. Scott, "Balancing exploration and exploitation: A new algorithm for active machine learning," in *IEEE International Conference on Data Mining (ICDM)*, 2005.

[14] T. Nguyen and A. Smeulders, "Active learning using pre-clustering," in *International Conference on Machine Learning (ICML)*, 2004.

[15] P. Poupart, "Non-myopic active learning: A reinforcement learning approach," in *Google Talk*, March 2009.

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

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

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

[19] 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.

[20] R. Karimi, C. Freudenthaler, A. Nanopoulos, and L. Schmidt-Thieme, "Comparing prediction models for active learning in recommender systems," in *34th Annual Conference of the German Classification Society(GfKl)*. Springer, 2010.

[21] 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.

[22] 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)*. ACM, 2008, pp. 251–258.

[23] G. Schohn and D. Cohn, "Less is more: Active learning with support vector machines," in *International Conference on Machine Learning (ICML)*, 2000.

[24] W. K. Newey and D. Mcfadden, "Large sample estimation and hypothesis testing," in *International Conference on Information Visualisation*, 1994.