Comparing Prediction Models for Active Learning 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
{karimi,freudenthaler,nanopoulos,schmidt-thieme}@ismll.uni-hildesheim.de

Abstract. Recommender systems help web users to address information overload. Their performance, however, depends on the amount of information that users provide about their preferences. Users are not willing to provide information for a large amount of items, thus the quality of recommendations is affected. Active learning for recommender systems has been proposed in the past, to acquire preference information from users. Early active learning methods for recommender systems used as underlying model either memory-based approaches or the aspect model. However, matrix factorization has been recently demonstrated (especially after the Netflix challenge) as being superior to memory-based approaches or the aspect model. Therefore, it is promising to develop active learning methods based on this prediction model. In this paper, we thoroughly compare matrix factorization with the aspect model to find out which one is more suitable for applying active learning in recommender systems. The results show that beside improving the accuracy of recommendations, the matrix factorization 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.

1 Introduction

Recommender systems guide users in a personalized way to interesting or useful objects in a large space of possible options. There are several techniques for recommendation and collaborative filtering is one them [1, 2]. Given a domain of items, users give ratings to these items. The recommender system can then compare the user's ratings to those of other users, find the most similar users based on some criterion of similarity, and recommend items that similar users have already liked.

Evidently, the performance of recommender systems depends on the number of ratings that the users provide. This problem is amplified even more in the case where we lack ratings due to a new user (cold-start problem). There are different solutions to deal with this problem. The first solution is to use the meta data of the new user. However, even a few ratings are more valuable than the meta data [21]. Therefore, the new user is requested to provide ratings to some items. But a well identified problem is that users are not willing to provide ratings for a large amount of items [4, 5]. Therefore, the queries presented to the new user have to be selected carefully. To address this situation active learning methods have been proposed to acquire those ratings from the new user that will help most in determining his/her interests [5, 4]. Another approach for the new user problem is to use implicit feedback. It means the recommender system uses implicit information from the user (browsing, viewing events) that can be used to quickly adjust his/her user model to his/her real taste, while interacting with the system [22]. In this paper, we focus on the active learning approach and do not deal with the other solutions.

Early active learning methods for recommender systems were developed based on Aspect Model (AM) [4, 5]. However, Matrix Factorization (MF) has been demonstrated (especially after the Netflix challenge) as being superior to other techniques. Therefore, it is promising to develop active learning methods based on this prediction model. In this paper we examine AM and MF for the new user problem in recommender systems. For this problem, in addition to the accuracy, training time of the prediction model is also important. It is because the preference elicitation of the new user is an interactive scenario and long time interruptions cause the new users to leave the conversation.

This paper is organized as follows: in section 2, related work is reviewed. In section 3, MF and AM are explained. In section 4, the training algorithms of MF and AM are compared. The experimental result is given in section 5. Finally the conclusion is stated in section 6.

2 Related Work

Active learning, in the context of the new-user problem, was introduced by Kohrs and Merialdo [9]. 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. Al Mamunur et al. [6] expanded this work, by considering the popularity of items and also personalizing the item selection for each individual user. Boutilier et al. [10] 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 [4] developed a new active learning algorithm based on AM which is similar to applying active learning for parameter estimation in Bayesian networks [11]. 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 [5] extended [4] by relaxing the 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 [12] applied the 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 [13] developed a non-myopic active learning which capitalizes explicitly on the update procedure of the MF model. Initially, this method queries items that if the new user's features are updated with the provided rating, it 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. [14] 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. Karimi et. al. [15] improved the most popular item selection according to the characteristics of MF. It finds similar users to the new user in the latent space and then selects the item which is most popular among the similar users.

The idea of using decision trees for cold-start recommendation was proposed by Al Mamunur et. al [8]. Golbandi et. al [7] improved [8] by advocating a specialized version of decision trees to adapt the preference elicitation process to the new user's responses. Zhou et. al [20] modified [7] by associating matrix factorization to decision trees. Karimi et. al [16] proposed another approach to introduce matrix factorization in decision trees which is more scalable compare to [20]. Karimi et. al [17] improved the decision trees by splitting the nodes of the trees in a finer-grained fashion. Specifically, the nodes are split in a 6-way manner instead of 3-way split. Karimi et. al [18] proposed an innovative approach for active learning in recommender systems. The main idea is to consider existing users as (hypothetical) new users and solve an active learning problem for each of them. In the end, we aggregate all solved problems in order to learn how to solve the active learning problem for a real new user.

3   Background

In this section, a short introduction to AM and MF is provided.

3.1   Aspect Model

The Aspect Model is a probabilistic latent space model, which models user interests as a mixture of preference factors [24, 25]. The latent class variables

$f \in F := \{f_1, f_2, ..., f_k\}$ are associated with each user u and each item i. Users and items are independent from each other given the latent class variable f. The probability for each observation tuple (u, i, r) is calculated as follows:

$$p(r|i, u) = \sum_{f \in F} p(r|f, i)p(f|u) \qquad (1)$$

where $p(f|u)$ is a multinomial distribution and stands for the likelihood for user u to be in the latent class f. $p(r|f, i)$ is the likelihood of assigning item i with rating r for class f. In order to achieve better performance, the training ratings of each user are normalized with zero mean and variance 1 [25]. The parameter $p(r|f, i)$ is a Gaussian distribution $N(\mu_{i,f}, \sigma_{i,f})$ with latent class mean $\mu_{i,f}$ and standard deviation $\sigma_{i,f}$.

3.2 Matrix Factorization

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 $\hat{R}$ of user u to item i is the inner product of the user u features and item i features $h_i^T w_u$. 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 [3]:

$$\hat{r}_{ui} = \mu + b_i + b_u + h_i^T w_u \qquad (2)$$

where μ is the global average, b_i and b_u are item and user bias respectively. 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:

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

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 details of MF learning algorithm is described in [3].

When MF is applied to 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. To solve this problem we use the sigmoidal function to automatically truncate the predicted rating to the range of minimum and maximum ratings. Therefore, the predicted ratings are computed as follows:

$$\hat{r}_{ui} = MinRating + \frac{(MaxRating - MinRating)}{1 + e^{-(\mu + b_i + b_u + h_i^T w_u)}} \tag{4}$$

3.3 Retraining Policy

When a new user enters the recommender system, the prediction model (AM or MF) 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 which means after a first training, further retraining is only done for new users.

For online updating, we use the method introduced in [23]. In this method after getting a new rating for the new user, the user's latent features are initialized to a random setting and then learned using all ratings of the new user. The complexity of retraining is the same as the training but the size of training data, S, is only the number of ratings used for online updating which is just the ratings provided by the new user.

When the online updating technique is applied in MF, the learning step should be reduced. This is because the number of training data (ratings provided by the new user) is small and updating the new user's latent features should be done more precisely and carefully. In our experiments the learning step in the training phase is 0.01 and is reduced to 0.001 when online updating is performed.

4 Comparing AM and MF

The training algorithm for MF has the time complexity of [23] :

$$O(L \times |S| \times k) \tag{5}$$

where L is the maximum number of iterations. The learning algorithm stops if the RMSE on the training data is smaller than ϵ.

The training algorithm for AM is shown in Algorithm 1. In this algorithm, the convergence criterion is the same as the convergence criterion in MF. According to this algorithm the time complexity of AM is $O(L \times |S| \times k)$ which is equal to Equation 5. Therefore MF and AM have the same time complexity. However, AM needs more computations because there are two essential differences between AM and MF.

Algorithm 1 Aspect Model Training Algorithm According to [25]
___
  loop {repeat until convergence}
    for r_{ui} in S **do**
      for $f \leftarrow 1, ..., k$ **do**
        compute E-Step for each f
      end for
      for $f \leftarrow 1, ..., k$ **do**
        update $p(f|u)$, $\mu_{i,f}$, and $\sigma_{i,f}$
      end for
    end for
    for $f \leftarrow 1, ..., k$ **do**
      normalize $p(f|u)$
    end for
    check the convergence
  end loop
___

First, the learning algorithm of MF uses the gradient descent but AM is based on expectation maximization. While in the gradient descent the gradient is computed just by one training sample, in the expectation maximization the amount of change should be computed using all training data. This step is called E-step [24]. The time complexity of the E-step is $O(L \times |S| \times k)$. The second difference is that as AM is a probabilistic approach, the user features must be normalized so the summation of probabilities becomes 1. But MF is an algebraic approach, so it is not necessary to normalize the user features. The time complexity of the normalization is $O(L \times k)$. Finally though the maximum number of iterations L is the same for AM and MF (100 in our experiments), but the effective L in MF is lower than the effective L in AM, because MF converges faster than AM which consequently cuts down the training time.

5   Experimental Results

As 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, we evaluate AM and MF with respect to their accuracy on the new users in terms of prediction error versus the number of queried items (simply denoted as number of queries). The mean absolute error (MAE) is used to evaluate the performance of each test user u :

$$MAE_u = \frac{1}{|M_u|} \sum_{i \in M_u} |r_{ui} - \hat{r}_{ui}| \tag{6}$$

where M_u is the set of test items of user u, r_{ui} is the true rating of user u for item i, and $\hat{r}_{ui}$ is the predicted rating. Since the test dataset includes multiple users, the reported MAE is the average over individual MAE for each test user.

5.1 Data Set

We use the MovieLens(100K)[1] 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 [5]) and the rest of the 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 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. Of course, this is not a realistic assumption because there are items that the new user has not seen before, so it is not possible for him/her to provide the rating. As the focus of this paper is on the suitable prediction model for active learning in recommender system, we will leave this issue for future work. 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. The number of latent dimensions k is 10 according to [5].

5.2 Results

In this section, we compare the accuracy of the active learning algorithm based on MF with the active learning algorithm based on AM. The objective is to show that MF is a better prediction model to be used for developing the active learning algorithm. For this reason, in order to have a fair comparison we focus only on the prediction model and simply apply random selection of the queried items for both MF and AM.

Learning the new user's features usually starts with 3 initial ratings [5, 4]. This can be done in two different ways. The first option is to add the ratings to the training user dataset and train AM or MF with all users together. The further retraining of the new user is done using the online updating technique. The second way is to train the prediction model (AM or MF) only with training users, and then train the new user with three initial ratings using the online updating technique.

For AM, both ways provide the same initial error, i.e before asking any query. But for MF, the error is lower when online updating is used from the beginning (i.e the second way). This evidence shows a new solution to improve the accuracy of MF. MF can not make accurate predictions for users with few ratings [26]. Therefore, after training all users and items, the latent features of such users can be retrained using the online updating technique. This is an open door for further research.

Now we move on to compare MF and AM for 10 queries. Fig. 1 depicts the resulting MAE as a function of the number of queried items. MF outperforms

---

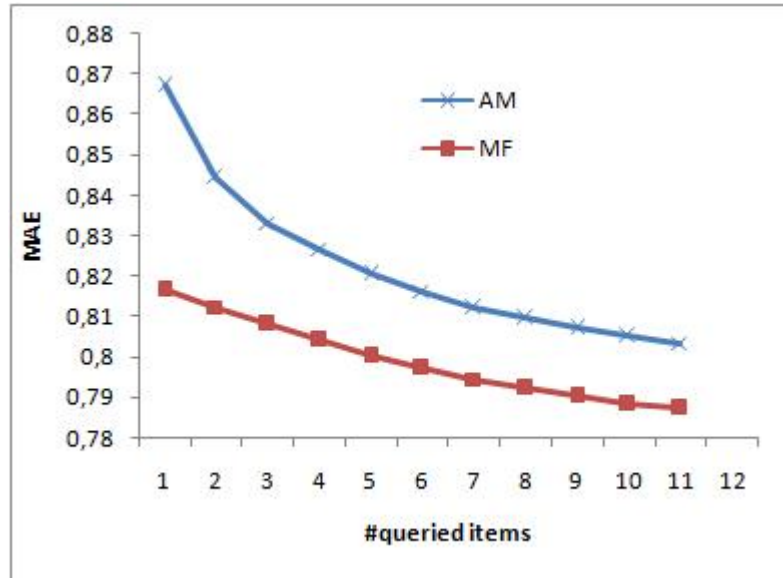[1] www.grouplens.org/system/files/ml-data0.zip

Fig. 1. Active Learning trends for 10 active-iterations

AM, indicating its superiority as the prediction model. In addition to the accuracy, the time required to retrain the new users latent features is also important. It is because the preference elicitation of the new user is an interactive scenario and long time interruptions make the new users leave the conversation. Table 1 compares the retraining time of new users latent features in AM and MF. Although both of them have the same complexity, but due to the reasons that have already been mentioned, MF is faster than AM.

Table 1. Retraining time of new users latent features in AM and MF

	Aspect Model	Matrix Factorization
MovieLens	44.5s	3.9s

6 Conclusion

In this paper, we proposed to develop active learning methods based on matrix factorization. We compared the training algorithm of matrix factorization with the aspect model and showed that matrix factorization is faster and its accuracy is also better.

As the future work, we plan to conduct online survey to validate the significance of our offline evaluation. To this end, it is crucial to design a software with a user-friendly user interface to encourage users to cooperate with the system [19].

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*, vol. 17, no. 6, pp. 734–749, 2005.
2. 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.
3. Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, pp. 30–37, 2009.
4. 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.
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*. ACM, 2008, pp. 91–98.
6. 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.
7. N. Golbandi, Y. Koren, and R. Lempel, "Adaptive bootstrapping of recommender systems using decision trees." in *WSDM*. ACM, 2011, pp. 595–604.
8. A. M. Rashid, G. Karypis, and J. Riedl, "Learning preferences of new users in recommender systems: an information theoretic approach," *SIGKDD Explor. Newsl.*, vol. 10, no. 2, pp. 90–100, Dec. 2008.
9. A. Kohrs and B. Merialdo, "Improving collaborative filtering for new users by smart object selection," in *International Conference on Media Features (ICMF)*, 2001.
10. C. Boutilier, R. S. Zemel, and B. Marlin, "Active collaborative filtering," in *Conference on Uncertainty in Artificial Intelligence(UAI)*, 2003.
11. S. Tong and D. Koller, "Active learning for parameter estimation in bayesian networks," in *Advances in Neural Information Processing Systems(NIPS)*, 2000.
12. R. Karimi, C. Freudenthaler, A. Nanopoulos, and L. Schmidt-Thieme, "Active learning for aspect model in recommender systems," in *IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*. IEEE, 2011.
13. 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.
14. R. Karimi, C. Freudenthaler, A. Nanopoulos, and L. Schmidt-Thieme, "Towards optimal active learning for matrix factorization in recommender systems," in *23th IEEE International Conference on Tools With Artificial Intelligence (ICTAI)*, 2011.
15. R. Karimi, C. Freudenthaler, A. Nanopoulos, and L. Schmidt-Thiemee, "Exploiting the characteristics of matrix factorization for active learning in recommender systems," in *RecSys*, 2012, pp. 317–320.
16. R. Karimi, M. Wistuba, A. Nanopoulos, and L. Schmidt-Thieme. Factorized decision trees for active learning in recommender systems. In *25th IEEE International Conference on Tools With Artificial Intelligence (ICTAI)*, 2013.

17. R. Karimi, A. Nanopoulos, and L. Schmidt-Thieme. Improved Questionnaire Trees for Active Learning in Recommender Systems. Proceedings of the 16th LWA Workshops: KDML, IR and FGWM, Aachen, Germany, September 8-10, 2014.

18. R. Karimi, A. Nanopoulos, and L. Schmidt-Thieme. A supervised active learning framework for recommender systems based on decision trees. User Model. User-Adapt. Interact. 25(1): 39-64 (2015).

19. R. Karimi, Fuzzy Model View Controller Pattern in International Conference on Advances in Intelligent Systems, Theory and Applications in cooperation with IEEE Computer Society, 2004.

20. 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*, ser. SIGIR '11. ACM, 2011, pp. 315–324.

21. Pilászy I, Tikk D (2009) Recommending new movies: even a few ratings are more valuable than metadata. In: RecSys, pp 93–100

22. Zhang L, Meng XW, Chen JL, Xiong SC, Duan K (2009) Alleviating cold-start problem by using implicit feedback. In: Proceedings of the 5th International Conference on Advanced Data Mining and Applications, Springer-Verlag, Berlin, Heidelberg, ADMA '09, pp 763–771

23. Rendle S, Schmidt-Thieme L (2008) Online-updating regularized kernel matrix factorization models for large-scale recommender systems. In: ACM Conference on Recommender Systems (RecSys), ACM, pp 251–258

24. Hofmann T, Puzicha J (1999) Latent class models for collaborative filtering. In: International Joint Conference on Artificial Intelligence, Morgan Kaufmann Publishers Inc., pp 688–693

25. Hofmann T (2003) 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, pp 259–266

26. Salakhutdinov R, Mnih A (2008) Probabilistic matrix factorization. In: Advances in Neural Information Processing Systems (NIPS 2007), pp 134–141