Application of nonnegative matrix factorization in recommender systems

Conference Paper · November 2012

DOI: 10.1109/ISTEL.2012.6483108

3 authors:


Mehdi Hosseinzadeh Aghdam
University of Bonab


Morteza Analoui
Iran University of Science and Technology


Peyman Kabiri
Iran University of Science and Technology

Application of Nonnegative Matrix Factorization in Recommender Systems

Mehdi Hosseinzadeh Aghdam, Morteza Analoui and Peyman Kabiri
School of Computer Engineering
Iran University of Science and Technology
Tehran, Iran
mhaghdam@iust.ac.ir, analoui@iust.ac.ir, Peyman.Kabiri@iust.ac.ir

Abstract—**Recommender systems actively collect various kinds of data in order to generate their recommendations. Collaborative filtering is based on collecting and analyzing information on users' preferences and estimating what users will like based on their similarity to other users. However, most of current collaborative filtering methods often suffer from two problems: sparsity and scalability. This paper proposes a framework for collaborative filtering by applying nonnegative matrix factorization, which alleviates the problems via matrix factorization. Experimental results on benchmark dataset are presented to show that our method is indeed more tolerant against both sparsity and scalability, and obtains good performance in the meanwhile.**

Keywords-component; recommender sysem; collaborative filtering; matrix factorization

I. INTRODUCTION

Collaborative Filtering (CF) is a natural choice for designing Recommender Systems (RSs) since it provides recommendations by centrally analyzing the user-item matrix alone [1]. CF can generate user specific recommendations based on historical user preferences. Inside a CF, the user interests on involved items (e.g., films, books, purchase records, etc.) are quantized into a user-item rating matrix, where high ratings denote strong preferences. So the problem of CF can be regarded as the problem of missing data estimation, in which the main task is to estimate the unknown user-item pairs based on known entries with minimum accumulative error [2]. CF supposes that users sharing the same ratings on past items likely to agree on new items. Research on CF can be grouped into two categories: memory-based and model-based [3].

Memory-based methods compute similarities between users or between items and apply them to recognize the top most similar neighbors. Then the unknown rating is estimated by combining the known rating of the neighbors. However, there exist two essential problems for memory-based approaches. The first one is sparsity. These methods rely on exact matches of two user-item vectors, which cause the methods to sacrifice RS coverage and accuracy. More specifically, since the correlation coefficient is only defined between users who have rated at least two items in common, or the items which have been unrated, then many pairs of users-items will have no correlation at all. As a consequence, memory-based RSs cannot precisely determine the neighborhood and recognize the items to recommend, which will surely lead to poor recommendations. The second one is scalability. In practice, both the number of users and items can be quite large. This may slow down the recommendation process significantly since K-Nearest Neighbor (KNN) based methods will need too much computations in this case [4].

Different from memory-based methods, model-based methods require establishing a model using training instances that can estimate the unknown ratings of a user. For examples decision tree [5], aspect models [6], clustering models [7], latent factor models [8], Bayesian network [5] and dimension-reduction approaches [9] are model-based methods. However, creating and updating of a model are often time-consuming since there are usually many free parameters to tune [10].

The Matrix Factorization (MF) based methods have become popular when building CF, due to the high accuracy and scalability. A low-rank MF method begins with the assumption that there exist a small number of latent factors (features) that can explain the rating behavior of users. Users and items are displayed as feature vectors in this latent space, where similarity between a user-item pair represents the tendency of the user to rate that item. While exact interpretability is never easy with latent feature models, when the items are movies, one can imagine that latent factors implicitly capture features such as personal information for users (e.g., age, gender and occupation) or Information about the items (e.g., genre, actors and directors) [11].

In a typical MF method to CF, a user and an item are displayed as unknown feature vectors whose dimensions are considered as latent features. These feature vectors are learnt so that inner products match the known preference ratings. This is similar to the problem of building weighted approximations of the preference matrix by taking the product of low-rank factor matrices, where weights are chosen such that known ratings are emphasized in measuring the quality of the approximation. Once the features are learnt, they prepare estimation for unknown ratings which may then be used for producing recommendations. Various models differ in the approximation criteria or the loss function they employ, and variants may be derived by adding different kinds of regularization to avoid overfitting [11].

Much of the prior work in these contexts has explored unconstrained SVD-like factorizations, while we focus on the use of Nonnegative Matrix Factorizations (NMF) [12, 13].

978-1-4673-2073-3/12/$31.00 ©2012 IEEE

NMF imposes nonnegative constraints on the latent features. The history of each user behavior is approximated as a mixture of item, i.e., basis vectors in the item space. By disallowing subtractive basis, non-negativity constraints lend a "part-based" interpretation to the predictive model. NMF with generalized KL-divergence loss is equivalent [14] to Probabilistic Latent Semantic Analysis [15] which has previously been used for CF tasks [16].

The rest of this paper is organized as follows. Section 2 explains recommender systems methods. Problem definition is described in Section 3. Section 4 presents nonnegative matrix factorization model. Section 5 reports computational experiments. It also includes a brief discussion of the results obtained and finally we conclude thepaperin the last section.

II. Recommender Systems

Recommender systems are software tools that actively collect various kinds of data in order to generate their recommendations. Data is primarily about the items to propose and the users who will receive these recommendations. But, since the information sources available for RSs can be very diverse, eventually, whether they can be exploited or not depends on the recommendation method. The recommendation problem truly arises as an independent field of research in the mid 1990's. It has deeper roots in several other areas like information retrieval and cognitive science. Methods for this problem are normally grouped in two categories: content-based and collaborative filtering methods [17].

The core of content-based (cognitive) methods [18, 19] is to recognize the common attributes of items that have received a favorable rating from a user, and then recommend to user new items that share these attributes. In content-based RSs, rich information explaining the nature of each item is assumed to be achievable in the form of a feature vector. The user profiles can then be used to recommend new items to a user, by proposing the item whose feature vector is most similar to the profile vector of user, for example, using the cosine similarity [18] or the minimum description length [19]. This method can also be used to estimate the rating of a user for a new item [19]. Bayesian methods using content information have also been proposed to estimate ratings [18].

RSs based purely on content generally suffer from the problems of limited content analysis and over-specialization [17]. Limited content analysis emerges from the fact that the system may have only a limited amount of information on its users or the content of its items. There are many reasons for this lack of information. For example, because of privacy issues a user may not provide personal information, or the exact content of items may be difficult or costly to get for some kinds of items, such as music or images. Finally, the content of an item is often inadequate to determine its characteristic. Over-specialization is a result of the way in which content-based systems recommend new items, where the estimated rating of a user for an item is high if this item is similar to the ones liked by this user. Solutions suggested for this problem include adding some randomness or filtering out items that are too similar [18].

In contrast to content-based methods, collaborative (social) filtering methods depend on the ratings of a user as well as those of other users in the system[17]. The main idea is that the rating of a user for a new item is likely to be similar to that of another user, if both users have rated other items in a similar way. Also, a user is likely to rate two items in a similar way, if other users have given similar ratings to these two items. CF methods overcome some of the limitations of content-based methods. For example, items for which the content is not available or difficult to get can still be recommended to users through the feedback of other users. In addition, CF recommendations are based on the quality of items as rated by neighbors, instead of depending on content. CF methods can recommend items with very different content, as long as other users have already shown preference for these different items.

CF methods can be grouped in the two general categories of memory-based and model-based methods. In memory-based (neighborhood-based or heuristic-based) [17], the ratings collected in the system are directly used to estimate ratings for new items. Memory-based methods can be implemented in two ways known as user-based or item-based recommendation. User-based methods, such as GroupLens, Bellcore video, and Ringo, evaluate the preference of a user for an item using the ratings for this item by other users, termed neighbors, that have similar rating patterns. The neighbors of a user are typically the users whose ratings on the items rated by both users are most correlated to those of user. Item-based methods estimate the rating of a user for an item based on the ratings of user for items similar to this item. In Item-based methods, two items are similar if several users of the system have rated these items in a similar way. Unlike memory-based methods, which use the stored ratings directly in the estimation, model-based methods use these ratings to learn a predictive model. The main idea is to model the user-item interactions with factors displaying latent features of the users and items in the system, like the preference class of users and the category class of items. This model is then trained using the available information, and later used to estimate ratings of users for new items. Model-based methods for the task of recommending items are numerous and include Bayesian Clustering [5], Latent Semantic Analysis [15], Latent Dirichlet Allocation [20], Maximum Entropy [21], Boltzmann Machines [22], Support Vector Machines and Singular Value Decomposition [17].

According to recent progress on RSs, one most successful kind of method to CF is based on MF. MF based recommenders work by transforming both users and items into the same latent feature (factor) space, characterizing each entity with a feature vector inferred from the exist in gratings, and then making predictions for unknown ratings using the inner products of the corresponding vector pairs. The earliest work of this kind is proposed by Sarwar et al. employing the Singular Value Decomposition (SVD) [23]. More recently, several MF techniques have been successfully applied to implementing CF recommenders, including the Probabilistic Latent Semantic Analysis [16], the Maximum Margin Matrix Factorization [24], and the Expectation Maximization for Matrix Factorization [25]. During the Netflix Prize, Brandyn Webb published the Regularized Matrix Factorization (RMF), which is accurate, highly efficient and easy to implement. Inspired by RMF, many researchers have further investigated MF based approaches. They have proposed sophisticated MF based CF recommenders [26].

III. PROBLEM DEFINITION

There are a set of users $U = \{u_1, u_2, \dots, u_n\}$ and a set of items $\mathbb{I} = \{i_1, i_2, \dots, i_m\}$ in RSs. The ratings expressed by users on items are given in a rating matrix $R = [r_{ij}]_{n\times m}$. In this matrix r_{ij} indicates the rating of user i on item j. r_{ij} can be any real number, but often ratings are integers in the range [1..5]. The task of a RS is as follows: Given a user $i \in \mathbb{U}$ and an item $j \in \mathbb{I}$ for which r_{ij} is unknown, estimate the rating for i on item j using matrix R. This paper applies nonnegative matrix factorization to learn the latent features of users and items and estimate the unknown ratings using these latent features. Let $U \in \mathbb{R}^{n\times k}$ and $I \in \mathbb{R}^{m\times k}$ be latent user and item factor matrices, with row vectors U_i and I_j representing k-dimensional user-specific and item-specific latent feature vectors of users i and item j, respectively. The matrix factorization attempts to learn these latent features and exploit them for recommendation.

IV. NONNEGATIVE MATRIX FACTORIZATION MODEL

Nonnegative matrix factorization has been investigated by many researchers, but it has achieved popularity through the researches of Lee and Seung reported in Nature and NIPS [13, 27]. In order to the argument that the non-negativity is critical in human perception they presented simple methods for finding nonnegative representations of nonnegative data. The basic NMF problem can be considered as follows: Given a nonnegative matrix $R \in \mathbb{R}_+^{n\times m}$ (with $r_{ij} \geq 0$ or equivalently $R \geq 0$) and a rank k, find two nonnegative matrices $U \in \mathbb{R}_+^{n\times k}$ and $I \in \mathbb{R}_+^{m\times k}$ which factorize R as well as possible (the rank of matrices U and I is lower than the one of matrix R, i.e. $k \leq \min(n, m)$), that is:

$$R \approx UI^T \tag{1}$$

In order to estimate factor matrices U and I in the NMF, we need to consider the similarity measure to quantify a difference between the matrix R and the approximate NMF model matrix $\hat{R} = UI^T$. The choice of the similarity measure mostly depends on the probability distribution of data. The simple way, use Frobenius-norm measure:

$$D_F = \frac{1}{2}\|R - UI^T\|_F^2 \tag{2}$$

This is also termed as the squared Euclidean distance. The above cost function is convex with respect to either the entries of the matrix U or the matrix I, but not both. Alternating minimization of such a cost function leads to the ALS (Alternating Least Squares) which can be explained as follows:

1) Initialize matrix U randomly or by using a specific deterministic strategy.

2) Estimate matrix I from the matrix equation $U^T U\, I^T = U^T R$ *by solving:*

$$\min_I D_F = \frac{1}{2}\|R - UI^T\|_F^2, \qquad with\ fixed\ U.$$

3) Set all the negative entries of the matrix I to zero or some small positive value.

4) Estimate matrix U from the matrix equation $I^T I U^T = I^T R^T$ *by solving*

$$\min_U D_F = \frac{1}{2}\|R^T - IU^T\|_F^2, \qquad with\ fixed\ I.$$

5) Set all the negative entries of the matrix U to zero or some small positive value.

Another commonly used cost function for NMF is the generalized Kullback-Leibler divergence (also termed the I-divergence) [27]:

$$D_{KL} = \sum_{i,j}\left(r_{ij}\ln\frac{r_{ij}}{[UI^T]_{ij}} - r_{ij} + [UI^T]_{ij}\right). \tag{3}$$

Most existing methods minimize only one type of cost function by alternately switching between sets of parameters. In this paper, we adopt a more general and flexible method in which instead of one cost function we use two cost functions (with the same global minima); one of them is minimized with respect to U and the other one with respect to I. This method is fully justified as U and I may have different distributions or different statistical properties and therefore different cost functions can be optimal for them.

The results and convergence supported by NMF methods usually highly depend on initialization. So, it is important to have efficient and consistent strategies for initializing matrices U and I. On the other hand, the efficiency of many NMF methods is affected by the selection of the starting matrices. Poor initializations often yielda slow convergence, and in certain instances may lead even to an incorrect or irrelevant solution. The problem of initialization matrices becomes even more complicated for large NMF problems and when certain constraints are applied on the factored matrices involved [12].

V. EXPERIMENTS

A. Dataset

We conducted a series of experiments with real usage data on the MovieLens dataset available at: http://www.grouplens.org/system/files/ml-data_0.zip. This dataset consists of 100,000 ratings on an integer scale from 1 to 5 given to 1642 movies by 943 users, where each user has rated at least 20 movies. We applied the 5-fold cross validation in our experiments. In each fold we have 80% of data as the training set and the remaining 20% as the test data.

B. *Evaluation* metrics

There are several kinds of measures for evaluating the performance of collaborative filtering approaches [4]. Weuse two popular metrics, the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE), to measure the estimation quality. The metrics MAE is defined as:

$$MAE = \frac{1}{R_{test}} \sum_{i,j} \left| r_{ij} - \hat{r}_{ij} \right|, \quad (4)$$

$\hat{r}_{ij}$denotes the rating user i gave to item j as predicted by a method, and R_{test} denotes the number of tested ratings. The metrics RMSE is defined as:

$$RMSE = \sqrt{\frac{1}{R_{test}} \sum_{i,j} \left(r_{ij} - \hat{r}_{ij} \right)^2}. \quad (5)$$

From the definitions, we can see that a smaller MAE or RMSE value means a better performance.

C. Experimental Results

We evaluate the results on NMF model and comparison shown in Table I.

TABLE I. RESULTS ON MOVIELENS DATASET

Table Head	MAE	RMSE
Fold-1	0.9221	1.2140
Fold-2	0.8951	1.1826
Fold-3	0.9054	1.1959
Fold-4	0.9039	1.1917
Fold-5	0.9215	1.2132

VI. CONCLUSION

Recommender systems attempt to estimate what the most suitable items are, based on the user's preferences and limitations. In order to complete such a computational task, RSs gather from users their preferences, which are either explicitly expressed as ratings for items, or are concluded by interpreting user behaviors. This paper focuses on the nonnegative matrix factorization in recommender systems. This model is a matrix factorization based method. Experiments on MovieLens dataset show the superiority of NMF.

ACKNOWLEDGMENT

This research was supported by the Iran Telecommunication Research Center (ITRC).

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 Trans. Knowl. Data Eng, vol. 17, pp. 734–749, 2005.

[2] X. Luo, H. Liu, G. Gou, Y. Xia, and Q. Zhu, "A parallel matrix factorization based recommender by alternating stochastic gradient decent,"Engineering Applications of Artificial Intelligence, 2011.

[3] G. Chen, F. Wang, and C. Zhang, "Collaborative Filtering Using Orthogonal Nonnegative Matrix Tri-factorization," Seventh IEEE International Conference on Data Mining – Workshops, pp.303-308, 2007.

[4] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," In Proc. of the WWW Conference, 2001.

[5] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," In Proc. of UAI, 1998.

[6] T. Hofmann and J. Puzicha "Latent class models for collaborative filtering," In Proc. ofIJCAI, 1999.

[7] L. H. Ungar and D. P. Foster, "Clustering methods for collaborative filtering," In Proc. Workshop on Recommendation Systems at AAAI, Menlo Park, CA, 1998. AAAI Press.

[8] J. Canny, "Collaborative filtering with privacy via factor analysis," In *Proc. of SIGIR*, 1999.

[9] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: a constant time collaborative filtering algorithm," *Information Retrieval*, 4(2), pp. 133–151, 2001.

[10] G.-R. Xue, C. Lin, Q. Yang, W. Xi, H.-J. Zeng, Y. Yu, and Z. Chen, "Scalable collaborative filtering using cluster-based smoothing," In Proc. of SIGIR, 2005.

[11] V. Sindhwani, S.S. Bucak, J. Hu, and A. Mojsilovic, "A Family of Non-negative Matrix Factorizations forOne-Class Collaborative Filtering Problems," In Proc. RecSys, New York, USA, 2009.

[12] A. Cichocki, R. Zdunek, A. Phan, and S. Amari, "Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation," Wiley, 2009.

[13] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," In Nature,401, pp. 788–791, 1999.

[14] C. Ding, T. Li, and W. Peng, "On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing" Computational Statistics and Data Analysis, pp. 3913–3927, 2008.

[15] T. Hofmann, "Probabilistic latent semantic indexing," Proceeding of SIGIR, pp. 50–57, 1999.

[16] T. Hofmann, "Latent semantic analysis for collaborative filtering," In ACM Transactions on Information Systems, 2004.

[17] C. Desrosiers and G. Karypis, "A Comprehensive Survey ofNeighborhood-based Recommendation Methods," In.F. Ricci, L. Rokach, B. Shapira, P. B., Kantor, (Eds.), Recommender Systems Handbook. , Springer, 2011.

[18] D. Billsus, M. J. Pazzani, "User modeling for adaptive news access. User Modeling and User-Adapted Interaction," vol. 10, pp. 147–180, 2000.

[19] K. Lang, "News Weeder: Learning to filter netnews," In Proc. of the 12th Int. Conf. on Machine Learning, pp. 331–339. Morgan Kaufmann publishers Inc. San Mateo, USA, 1995.

[20] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," Journal of Machine Learning Research, vol. 3, pp. 993–1022, 2003.

[21] C. L. Zitnick and T. Kanade, "Maximum entropy for collaborative filtering," In. Proc. of the 20th Conf. on Uncertainty in Artificial Intelligence, pp. 636–643. AUAI Press, Arlington, USA, 2004.

[22] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann machines for collaborative filtering," In. Proc. of the 24th international Conf. on Machine learning, pp. 791–798, New York, USA, 2007.

[23] B. Sarwar, G.Karypis, J.Konstan,J.Reidl, "Applicationofdimensionality reduction inrecommendersystems-a casestudy," ACMWebKDD,2000.

[24] N. Srebro, J. D. M. Rennie, T. S. Jaakola, "Maximum-margin matrix factorization," Adv. Neural Inf. Process. Syst, 17, 2005.

[25] M. Kurucz, A. Benczu´r, K. Csaloga´ny, "Methodsforlargescalesvdwith missing values," 13thACMSIGKDDInt.Conf.onKnowledgeDiscoveryand Data Mining California, ACM, 2007.

[26] Y. Koren, R. Bell, "Advances in Collaborative Filtering," In.F. Ricci, L. Rokach, B. Shapira, P. B., Kantor, (Eds.), Recommender Systems Handbook. , Springer, 2011.

[27] D. D. Lee and H. S. Seung, "Algorithms for Nonnegative Matrix Factorization," MIT Press, vol. 13, 2001.