JOURNAL OF RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY 2020(02)

Available online at http://journals.researchub.org

A new non-negative matrix factorization method to build a recommender system

Somaye Arabi naree**, Maryam Mohammadi

Faculty of Mathematical Sciences and Computer, Kharazmi University, Taleghani Avenue, Tehran, Iran

ARTICLE INFO

Article history:

Received 15 Oct 2019

Received in revised form 16 Jan 2020

Accepted 22 Feb 2020

Keywords:
Recommender Systems,
Non-Negative Matrix Factorization,
Update Rules.
.

ABSTRACT

The main aim of this paper is to apply non-negative matrix factorization to build a recommender system. In a recommender system there are a group of users that rate to a set of items. These ratings can be represented by a rating matrix. The main problem is to estimate the unknown ratings and then predict the interests of the users to the items which haven't rated. The main innovation of this paper is to propose a new algorithm to compute matrix factorization in a way that the factorized matrixes would be a good approximation for the initial rating matrix and moreover would be a good source to predict the unknown ratings of the items precisely. The results show that the proposed matrix factorization improves the estimated ratings considerably.

1. INTRODUCTION

Modern consumers are flooded with choices. Electronic vendors offer a huge selection of products, with extraordinary opportunities to meet a variety of special needs and tastes. Matching consumers with the most appropriate products is important to enhancing user satisfaction. Therefore, more retailers have become interested in recommender systems, which analyze patterns of user interest in products to provide personalized recommendations that suit a user's taste. Because good personalized recommendations can add another dimension to the user experience, e-commerce leaders like Amazon.com and Netflix have made recommender systems a salient part of their websites. Such systems are particularly useful for entertainment products such as movies, music, and TV shows. Many customers will view the same movie, and each customer is likely to view numerous different movies. Customers have proven willing to indicate their level of satisfaction with particular movies, so a huge volume of data is available about which movies appeal to which customers. Companies can analyze this data to recommend movies to particular customers.

Broadly speaking, recommender systems are based on one of two strategies: *content filtering* and *collaborative filtering*. The *content filtering* approach creates a profile for each user or product to characterize its nature. The profiles allow programs to associate users with matching products.

An alternative to content filtering relies only on past user behavior—for example, previous transactions or product ratings—without requiring the creation of explicit profiles. This approach is known as *collaborative filtering*. Collaborative filtering analyzes relationships between users and interdependencies among products to identify new user-item associations.

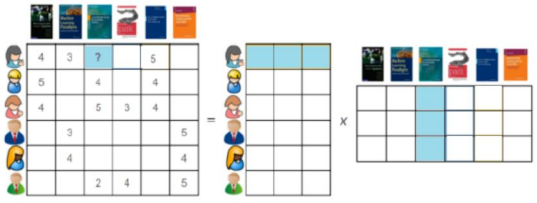
A major appeal of collaborative filtering is that it is domain free, yet it can address data aspects that are often indescribable and difficult to profile using content filtering. While generally more accurate than content-based techniques, collaborative filtering suffers from what is called the *cold start* problem, due to its inability to address the system's new products and users. In this aspect, content filtering is superior.

The two primary areas of collaborative filtering are the *neighborhood methods* and *latent factor models*. Neighborhood methods are centered on computing the relationships between items or, alternatively, between users. The item oriented approach evaluates a user's preference for an item based on ratings of "neighboring" items by the same user. A product's neighbors are other products that tend to get similar ratings when rated by the same user.

Latent factor models are an alternative approach that tries to explain the ratings by characterizing both items and users on factors inferred from the ratings patterns.

1.1. Application of NMF in recommender systems

NMF can be used to decompose the rating matrix to two factors which can represent the group of similar users and similar items [1]. To recommend an item to a special user, the interesting of the similar users can be used. For example consider Figure 1 that shows 6 users and 6 books. The ratings of users to items are represented by a matrix.



**Corresponding author:* Somaye.Arabinaree@gmail.com

DOI: https://doi.org/10.24200/jrset.vol8iss02pp12-16

Figure 1. Application of NMF in collaborative filtering

In this example we have 3 features (k=3) and these features can be related to users or items (books). Assume that we have to estimate the rating that user 1 gives to item 3. At first, the rating matrix is decomposed using NMF. By this factorization, the group of similar users to user 1 will be determined. As shown in Figure 1, users 1, 2 and 3 have similar ratings so they can be considered in a group and users 4 and 5 are considered in a group and similarly user 6 is in another group. Each row of the first factorization matrix represents that the corresponding user belongs to which group. In other words, each row of the first factorization matrix shows which feature is more important for the corresponding user. Each column of second factorization matrix shows that the corresponding book is more important for which group. In other words, which feature of the corresponding book is more important for users. So if the first row of the first factorization matrix would multiplied by the third column of the second factorization matrix, the result reveals that if the third book has which features that are important for user 1.

1.2. Proposed Approach

The aim is to find non-negative matrixes W and H in such a way that the equation (1) is satisfied:

$$A \cong WH = \hat{A} \quad (1)$$

Each row of matrix W represents the correlation between the corresponding user and the features (groups) detected by the factorization method. Also, each column of matrix H represents the correlation between groups and the corresponding items [2]. In another word, considering the ith row of matirx W which has k elements and each element is representative of a group, if the lth element of the row is the largest one among the other elements of this row then the probability that user i is related to group l is larger. In addition, considering the jth column of matrix H which has k elements and each element is representative of a group, if the lth elemnt of this column is the largest one among the other elements of the column then it is more probable that group l would be interested in item j [3].

Hence, to predict the score which user i will give to item j, we can compute the multiplication of row i of matrix W to column j of matrix H. let represent ith row of matrix W with W_i and jth column of matrix H with H_j , then:

$$\hat{a}_{ij} = W_i H_j = \sum_{l=1}^{k} w_{il} h_{lj} \quad (2)$$

The difference between the actual score and the estimated score is called estimation error [4]. Estimation error may be positive or negative according to which of the actual score or the estimated score is higher than the other. Therefore the square of the estimation error is computed to prevent the sign difference:

$$e_{ij}^2 = (a_{ij} - \hat{a}_{ij})^2 = (a_{ij} - \sum_{l=1}^{k} w_{il} h_{lj})^2 \quad (3)$$

In fact, minimizing the estimation error is commonly reformulated as the following optimization problem:

$$min_{W,H}\, f(W,H) = \frac{1}{2}\|A - WH\|_F^2 = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}(a_{ij} - \sum_{l=1}^{k} w_{il} h_{lj})^2 \ , s.t. \ \ W, H \geq 0 \quad (4)$$

The main goal is to minimize the estimation error. To this end, it is essential to determine in which direction the values of w_{il} and h_{lj} should be changed. In other words, it is nessessary to compute the gradient of error according to w_{il} and h_{lj} and then inorder to provid KKT conditions to find the minimum value we should reduce the gradient in each iteration [5]. Therefore, we compute the partial derivation of error according to w_{il} and h_{lj} as shown in raltions (5) and (6):

$$\frac{\partial e_{ij}^2}{\partial w_{il}} = -2\big(a_{ij} - \hat{a}_{ij}\big)\big(h_{lj}\big) = -2e_{ij}h_{lj} \quad (5)$$

$$\frac{\partial e_{ij}^2}{\partial h_{lj}} = -2\big(a_{ij} - \hat{a}_{ij}\big)(w_{il}) = -2e_{ij}w_{il} \quad (6)$$

Now to find the update rules, a coefficient of the above expressions are applied to build an additive form of the update rules:

$$\widetilde{w}_{il} = w_{il} + 2\alpha e_{ij} h_{lj}, \quad (7)$$

$$\tilde{h}_{lj} = h_{lj} + 2\alpha e_{ij} w_{il}. \quad (8)$$

By convergence of (W,H), the gradient becomes zero therefore KKT conditions are satisfied [6]. In this situation the algorithm is converged to a local minimum. Parameter α is a positive value which can be constant or can be updated during each iteration. Generally a little value is considered for this parameter otherwise the algorithm may miss the local minimum.

1.3. Regularization

During the iterations to minimize the estimation error, the value of the entries of the decomposed matrix may be increased that results in error enhancement in each iteration. To prevent the growth of the decomposed matrix entries inordinately, a coefficient of the Euclidean value of the row and column which the score prediction is based on them is added up to the estimation error of the corresponding row and column. By this way, not only the growth of estimation error is controlled but also the growth of the decomposed matrix entries are limited [7]. Therefore we can rewrite the estimated error as the following formula:

$$E_{ij}^2 = (a_{ij} - \sum_{l=1}^{k} w_{il} h_{lj})^2 + \beta\left(\|W_i\|^2 + \|H^j\|^2\right) = (a_{ij} - \sum_{l=1}^{k} w_{il} h_{lj})^2 + \beta(\sum_{l=1}^{k} w_{il}^2 + \sum_{l=1}^{k} h_{lj}^2). \quad (9)$$

So the partial derivations can be recomputed as follow:

$$\frac{\partial E_{ij}^2}{\partial w_{il}} = -2e_{ij}h_{lj} + 2\beta w_{il}, \quad (10)$$

$$\frac{\partial E_{ij}^2}{\partial h_{lj}} = -2e_{ij}w_{il} + 2\beta h_{lj}, \quad (11)$$

Therefore the update rules are defined as follow:

$$\widetilde{w}_{il} = w_{il} + 2\alpha\big(e_{ij}h_{lj} + \beta w_{il}\big), \quad (12)$$

$$\tilde{h}_{lj} = h_{lj} + 2\alpha\big(e_{ij}w_{il} + \beta h_{lj}\big). \quad (13)$$

Usually in practice, parameter β is set to value 0.01. Therefore, using the above update rules and an appropriate and small enough selection for the amount of parameter β, the update rules are applied until the estimation error would be decreased to a desirable amount of value.

1.4. Stop Criterion

Nearly all NMF algorithm implementations use a maximum number of iterations as stopping criteria [8]. However, a fixed number of iterations is not a mathematically appealing way to control the number of iterations executed. There are other choices to select as

stopping criterion. A way to select the stopping criterion is to compare the original matrix A with the estimated matrix $\hat{A}$. The difference between the non zero elements of the original matrix and the corresponding elements in the estimated matrix should not exceed a special value. This limitation can be defined according to a special problem. In this paper, the total estimation error is considered as a factor to set the stopping criteria. The error square is computed as the following relation:

$$E_{ij}^2 = (a_{ij} - \sum_{l=1}^{k} w_{il}h_{lj})^2 + \beta(\sum_{l=1}^{k} w_{il}^2 + \sum_{l=1}^{k} h_{lj}^2). \quad (14)$$

Now, if error is less than a predefined value (such as 0.0001) then the algorithm stops an outputs the estimated matrix.

1.5. Initialization

To start the factorization algorithm, W and H matrixes should be initialized. Usually, a weak initial value (such as random initialization) causes a slow convergence and sometimes irrelevant and incorrect results. Although a suitable initialization does not necessarily guarantee algorithm convergence, it certainly reduces the number of iterations. The performance of NMF algorithm is affected by the initial value of the W and H matrixes. So it is important to use consistent and efficient methods to select initial value for matrixes. In this paper, a multiplicative update rule algorithm is applied. The algorithm transforms the initial random matrixes to an approximation of NMF and then uses this approximation as an initial value for matrixes W and H.

1.6. Multiplicative update algorithm

Multiplicative update algorithm is one of the most applicable algorithms for solving NMF problem [9]:

$W = rand(m,k)$
$H = rand(k,n)$
$for\ i = 1{:}\ t$
$H = H.*\ (W^TA)./(W^TWH + 10^{-9})$
$W = W.*\ (AH^T)./(WHH^T + 10^{-9})$
end

The symbol ./ represents element by element division. In each iteration, the fraction is added by 10^{-9} to prevent divide by zero.

1.7. Convergence of the Multiplicative Update Algorithm to a Local Minimum

If the initial matrixes W and H are strictly positive, then according to the update rules and by assumption that matrix A doesn't have any zero row and column (if it has such row or columns, they can be removed without any damage to the problem), it is obvious that W and H remain positive in each iteration. If seqence (W,H) converges to (W^*, H^*) and the conditions W > 0 and H > 0 are satisfied then following relations are confirmed:

$$\frac{\partial f(W^*,H^*)}{\partial H} = 0 \quad (15)$$

$$\frac{\partial f(W^*,H^*)}{\partial W} = 0 \quad (16)$$

We prove the first relation and the second relation can be proved in a same way. The update rule for H can be rewriten as follows:

$$H = H + [H./(W^TWH)].*\ [W^T(A - WH)]. \quad (17)$$

Raltion (17) can be defined as follows:

$$H = H + [H./(W^TWH)].*\ [W^T(A - WH)]$$
$$= H + [H./(W^TWH).*\ W^TA] - [H./(W^TWH).*\ W^TWH]$$
$$= H + [H./(W^TWH).*\ W^TA] - H$$
$$= H./(W^TWH).*\ W^TA \quad (18)$$

Assume after each update, the resultant matrix H would be very close to the previous matrix H before update. In other words, assume that the matrix resulted from update would be a limitation point for the matrix before update and $H_{ij} > 0$ then according to relation (18):

$$\frac{H_{ij}}{[W^TWH]_{ij}}\left([W^TA]_{ij} - [W^TWH]_{ij}\right) = 0 \quad (19)$$

Because $H_{ij} > 0$ therefore:

$$[W^TA]_{ij} = [W^TWH]_{ij} \quad (20)$$

It means $[\frac{\partial f}{\partial H}]_{ij} = 0$. Therefore when the sequence (W, H) converges to (W^*, H^*) then H would be a limitation point for H^* from on step onwards [10][11]. On the other hand because the assumption of initial matrixes positiveness, for each i and j, $H_{ij} > 0$ and $[\frac{\partial f}{\partial H}]_{ij} = 0$. It means:

$$\frac{\partial f}{\partial H} = 0 \quad (21)$$

Obviously if the above partial derivations are zero, considering this fact that matrixes W and H are positive so the KKT conditions are satisfied. The local minimum is obtained according to W^* and H^*. Therefore if it is guaranteed that the initial matrixes are positive, the multiplicative update rules would converge to a local minimum.

2. EXPERIMENTAL RESULTS

Error evaluation factors are applied on non zero elements of matrix. In fact, non zero elements are the actual rating that should be analyzed and zero elements implies that the correspond item is not rated by the user. So error is computed based on non zero elements of the rating matrix.

Consider r_i is a non zero vector of the rating matrix and p_i is an approximation that the recommender system has computed and z is the number of the vector elements.

2.1. Mean Absolute Error

MAE is computed as follow [12]:

$$MAE_j = \frac{1}{j}\sum_i^j |r_i - p_i| \quad , j = 1,2,\dots,z. \quad (22)$$

2.2. Root Mean Square Error

RMSE is computed by the following formula:

$$RMSE_j = \sqrt{\frac{1}{j}\sum_i^j (r_i - p_i)^2}\ , j = 1,2,\dots,z. \quad (23)$$

2.3. Relative Error

RE is one of the popular factors to measure error which is computed by the following formula:

$$RE_i = \frac{|r_i - p_i|}{|r_i|} \quad (24)$$

2.4. Error Comparision

In this section, the proposed approch error is evaluated on Netflix data set [13]. RE diagram, as shown in Figures 2 and 3, represented as discrete points. In this diagram, the horizental and vertical axes represent index of data and

the error related to that index correspondingly. In MAE and RMSE diagrams, shown in Figures 4 and 5, the horizental and vertical axes represent the number of data and error value correspondingly.

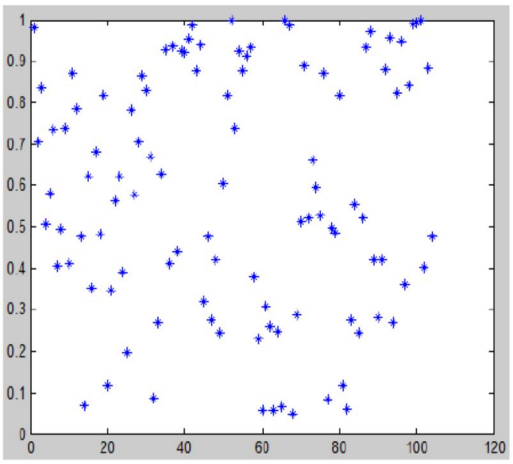


Figure 2. Relative error based on simple NMF

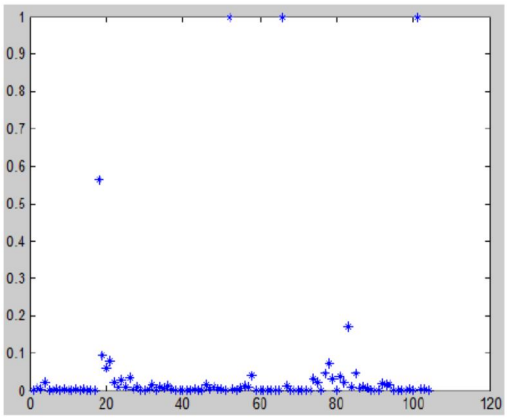


Figure 3. Relative error of the proposed approach

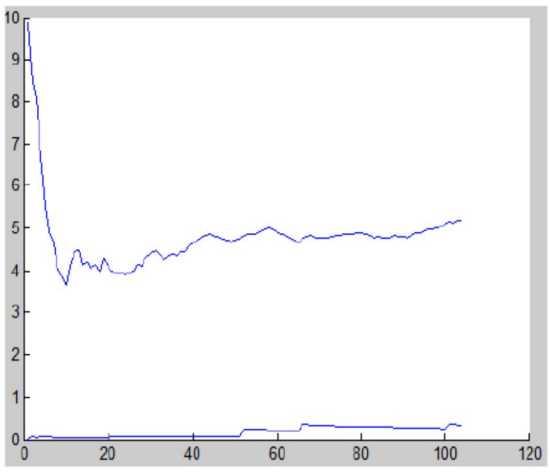


Figure 4. MAE comparison between the simple NMF and the proposed approach

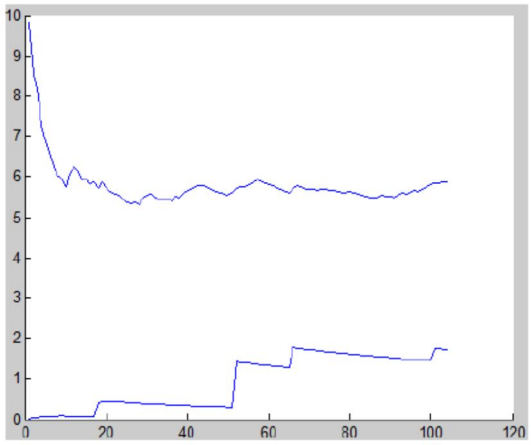


Figure 5. RMSE comparison between the simple NMF and the proposed approach

The results show that the proposed approach introduces less amount of error compared to the simple NMF factorization in recommender system.

3. CONCLUSION

Matrix factorization techniques have become a dominant methodology within collaborative filtering recommenders. Experience with datasets such as the Netflix Prize data has shown that they deliver accuracy superior to classical nearest-neighbor techniques. At the same time, they offer a compact memory-efficient model that systems can learn relatively easily. What makes these techniques even more convenient is that models can reveal the group of users that have similar interesting. This paper proposes a new update rule algorithm to compute matrix factorization. The proposed method guarantees the convergence of the algorithm to a local minimum which is the best factorization with the least estimation error.

REFERENCES

[1] Berry, M. W., Browne, M., Langville, A. N., Pauca, P., Plemmons, R. J. (2007). Algorithms and applications for approximate nonnegative matrix factorization, *Computational Statistics and Data Analysis*, 155-173.

[2] Bertsekas, D. P. (1999). *Nonlinear Programming*, 2nd. edn., Athena Scientific, Belmont, assachusetts.

[3] Kim, H., & Park, H. (2008). Nonnegative matrix factorization based on alternating non-negativity constrained least squares and the active set method, *SIAM J. Matrix Anal. Appl.*, *30*(2), 713-730.

[4] Kim, H., & Park, H. (2007). Sparse non-negative matrix factorization via automating non-negativity constrained least squares for microarray data analysis, *Bioinformatics*, 23, 1495-1502.

[5] Koren, Y., Bell, R., & Volinsky, Ch. (2009). Matrix factorization techniques for recommender system, *IEEE Computer*, 42, Issue 8, 30-37.

[6] Langville, A. N., Meyer, C. D., Albright, R., Cox, J., & Duling, D. (2014). Algorithms, initializations, and convergence for the nonnegative matrix factorization, *CoRR*.

[7] Lee, D. D., & Seung, H. S. (2011). Learning the parts of objects by non-negative matrix factorization, *Neural*, 401, 788-791.

[8] Lee, D. D., & Seung, H. S. (2001). Algorithms for non-negative matrix factorization, *Advanced in Neural Information Processing*, 13.
[9] Lin, C. J. (2005). On the convergence of multiplicative update algrithms for non-negative matrix factorization, *IEEE Transactions on Nueral Network*, 18(6), 1589-1596.
[10] Lin, C. J. (2005). Projected gradient methods for nonnegative matrix factorization, *Nueral Computation*, 19(10), 2756-2779.
[11] Paatero, P., Tapper, U. (1994). Positive matrix factorization: A non-negative factor model with optimal utilization for error estimates of data values, *J. Environ metrics*, vol. **5**, 111-126.
[12] Wang, Y. X., Zhang, Y. J. (2013). Nonnegative Matrix Factorization: A Comprehensive Review, *IEEE Transactions on Knowledge and Data Engineering*, Volume 25, Issue 6, 1336– 1353.
[13] Zhou, Y. et al. (2008). Large-Scale Parallel Collaborative Filtering for the Netflix Prize, *Proc. 4th Int'l Conf. Algorithmic Aspects in Information and Management*, LNCS 5034, Springer, 337-348.