Department of Industrial Engineering and Economics

Working Paper

No. 2017-5

Apparel Item Recommendation using Graph Regularized Nonnegative Tensor Factorization

Koji Tazawa, Kazuki Neichi, Yasuyuki Ohara,
Kazuki Chikuma, Shota Katayama,
Kazuhide Nakata

June 2017

Tokyo Institute of Technology

2-12-1 Ookayama, Meguro-ku, Tokyo 152-8552, JAPAN
http://educ.titech.ac.jp/iee/

Apparel Item Recommendation using Graph Regularized Nonnegative Tensor Factorization

Koji Tazawa[1], Kazuki Neichi[1], Yasuyuki Ohara[1], Kazuki Chikuma[2], Shota Katayama[1], and Kazuhide Nakata[1]

[1] *Dept. of Industrial Engineering and Economics, Tokyo Institute of Technology, Tokyo 152-8550, Japan*
[2] *UL Systems, Inc., Tokyo 104-0053, Japan*

Abstract

The number of items in e-commerce sites is continuously increasing, and attributes of those items are becoming more complicated. So it is crucial to find a way to deal with complicated data. A recommendation system using Nonnegative Tensor Factorization, NTF, is one way to tackle the problem. In NTF, we can deal with the data with 3 or more attributes like users, categories, and so on, as tensor's axes. In this paper, we proposed a new enhancement of NTF, named Graph Regularized NTF (G-NTF). The idea is based on one of the enhancements of Nonnegative Matrix Factorization, GWNMF [2]. Its factorization is regularized by graph networks which represent similarities between elements in each attribute respectively. We proved that the simple improvement of GWNMF's updating formula will converge in our method. G-NTF was applied to the real dataset of an apparel e-commerce site. This revealed that our method is successful and can recommend more suitable items to users than any other methods.

1 Introduction

Market size of e-commerce is expanding significantly, and the number of items in e-commerce sites is increasing. Therefore, for each user, to find and pick up items they really want are becoming difficult. Improving recommendation system is a fundamental problem in such a situation [4]. Users will purchase more if they can find items they want more easily, thereby e-commerce sites can increase their sales.

Nonnegative Matrix Factorization is popular way to recommend. If the training data is highly sparse because of too many items, however, the accuracy of recommendation tends to be low. There is a lot of research to tackle the problem [5]. However, as long as we try to recommend the specific item, it is not practical, and we cannot complement whole missing data by using the small amount of data. Instead of recommending specific item, we try to lower sparsity by recommending a set of items whose some of attributes match. We can use the tensor to deal with combination of some attributes. Each purchasement's log is stored in the element of the tensor. Each element corresponds to one user and one combination of some attributes. Nonnegative Tensor Factorization is one way to recommend by using the tennor.

In this paper, we propose the enhancement of NTF, denoted by G-NTF. Factorization of G-NTF is regularized by graph networks which represent similarities between elements in each attribute respectively. This is because that feature vectors of two users with similar purchasing tendencies should be similar.

We apply our method to the real dataset of order histories of the apparel e-commerce site, and evaluate our method. In the dataset, order histories are recorded with attributes of the apparel item, for example classification, price, color, and so on. Note that, the dataset was provided on Data Analysis Competition 2016 sponsored by Joint Association Study group of Management sCience (JASMAC).

2 Related Works

There are a lot of recommendation algorithms, but they would be classified into the two; the content-based filtering and the collaborative filtering. In this paper we focus on the latter. Intuitively, by using the collaborative filtering algorithm, we can recommend the user the item which was bought by other users who behaved like the user. Through this approach, we can recommend variety of items. Especially, we introduce NMF, GWNMF and NTF in this section.

Nonnegative Matrix Factorization, NMF, is one of the most popular collaborative filtering algorithm. NMF decomposes one nonnegative matrix $\boldsymbol{X} \in \mathbb{R}_+^{N\times M}$ into the two nonnegative matrices $\boldsymbol{A} \in \mathbb{R}_+^{N\times D}$ and $\boldsymbol{B} \in \mathbb{R}_+^{M\times D}$, so that the product $\boldsymbol{A}\boldsymbol{B}^T$ approximates $\boldsymbol{X}$. In the context of the recommendation, N is the number of users, M is the number items. The n-th row vector of $\boldsymbol{A}$, denoted by $\boldsymbol{A}_{n\cdot}$ can be regarded as the feature vector of the user n.

To find matrices $\boldsymbol{A}, \boldsymbol{B}$, NMF is formulated as follows;

$$\begin{array}{lll} \min & L = & \|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{B}^T\|^2 \\ \text{s.t.} & & \boldsymbol{A}, \boldsymbol{B} \geq 0 \end{array} \tag{1}$$

Gu et al. [2] proposed Graph Regularized Weighted Nonnegative Matrix Factorization, GWNMF, which works as a more precise recommendation system than NMF. The method uses the information of similarities between users and also between items in the process to find factor matrices.

We define user similarity matrix $\boldsymbol{W}^A = \{W_{ij}^A\}$, each element represents similarity between user i and j. It is calculated by using the information of users' ages, sex and so on. Similarly, it is needed to calculate item similarity matrix $\boldsymbol{W}^B = \{W^B_{ij}\}$ each element represents similarity between item i and j, by using the information of categories of items.

Two feature vectors corresponding two users who have high similarity score should be close. It is the same for items. Therefore, Gu et al. [2] formulated GWNMF as follows;

$$\begin{array}{llll} \min & L = & \|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{B}^T\|^2 \\ & & +\frac{\lambda_A}{2}\sum_{ij}\|\boldsymbol{A}_{i\cdot} - \boldsymbol{A}_{j\cdot}\|\boldsymbol{W}^A_{ij} \\ & & +\frac{\lambda_B}{2}\sum_{ij}\|\boldsymbol{B}_{i\cdot} - \boldsymbol{B}_{j\cdot}\|\boldsymbol{W}^B_{ij} \\ & = & \|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{B}^T\|^2 \\ & & +\lambda_A \mathrm{trace}\left(\boldsymbol{A}^T\boldsymbol{L}^A\boldsymbol{A}\right) + \lambda_B \mathrm{trace}\left(\boldsymbol{B}^T\boldsymbol{L}^B\boldsymbol{B}\right) \\ \text{s.t.} & & \boldsymbol{A}, \boldsymbol{B} \geq 0 \end{array} \tag{2}$$

where λ_A and λ_B are parameters of regularization, $\boldsymbol{D}^A$ is the diagonal matrix whose diagonal elements are $\boldsymbol{D}^A_{ii} = \sum_{j=1}^n W^A_{ij}$, $\boldsymbol{L}^A = \boldsymbol{D}^A - \boldsymbol{W}^A$ is the graph laplasian. Note that, the formulation is different from exact graph regularized WEIGHTED NMF which Gu et al. proposed. In this paper, for simplicity, we omit the weight matrix which defines how much to consider errors related to certain elements.

And also Gu et al. proposed the algorithm to find the solution of optimization problem below. The main step of the algorithm is composed by updating for each element. Updating formula of A's element is given as follows;

$$A_{nd} \leftarrow A_{nd}\sqrt{\frac{\left[\boldsymbol{X}\boldsymbol{B} + \lambda_A \boldsymbol{L}^-_A\boldsymbol{A}\right]_{nd}}{\left[(\boldsymbol{A}\boldsymbol{B}^T)\boldsymbol{B} + \lambda_A \boldsymbol{L}^+_A\boldsymbol{A}\right]_{nd}}}, \tag{3}$$

where $\boldsymbol{L}^A_+$ is the matrix which is replaced negative elements of matrix $\boldsymbol{L}^A$ by zero, and $\boldsymbol{L}^A_- = \boldsymbol{L}^A_+ - \boldsymbol{L}^A$. Similarly, we define $\boldsymbol{L}^B_+$ and $\boldsymbol{L}^B_-$ with updating formula of B's element is Given as;

$$B_{md} \leftarrow B_{md}\sqrt{\frac{\left[\boldsymbol{X}^T\boldsymbol{A} + \lambda_B \boldsymbol{L}^-_B\boldsymbol{B}\right]_{md}}{\left[(\boldsymbol{A}\boldsymbol{B}^T)^T\boldsymbol{A} + \lambda_B \boldsymbol{L}^+_B\boldsymbol{B}\right]_{md}}}. \tag{4}$$

Nonnegative Tensor Factorization [1], NTF, is simple extension of NMF. NTF decomposes one K-dimensional nonnegative tensor $\boldsymbol{X'} \in \mathbb{R}_+^{N_1 \times \cdots \times N_k}$ into K nonnegative matrices like $\boldsymbol{A}_{(1)} \in \mathbb{R}^{N_1 \times D}, \ldots, \boldsymbol{A}_{(K)} \in \mathbb{R}^{N_K \times D}$.

There are two approaches to formulate NTF, CP factorization and Tucker factorization, but we focus on the former 0in this research.

CP factorization is formulated as follows;

$$\begin{array}{lll} \min & L = & \|\boldsymbol{X'} - \sum_{d=1}^D \boldsymbol{A}_{(1)\cdot d} \circ \cdots \circ \boldsymbol{A}_{(K)\cdot d}\|^2 \\ \text{s.t.} & & \boldsymbol{A}_{(1)}, \cdots, \boldsymbol{A}_{(K)} \geq 0 \end{array} \tag{5}$$

where $\boldsymbol{A}_{(\boldsymbol{k})\cdot d}$ represents d-th column vector of $\boldsymbol{A}_{(\boldsymbol{k})}$ and the operator $\circ$ represents outer product of vectors.

3 Graph Regularized Nonnegative Tensor Factorization

In this section we propose Graph Regularized Nonnegative Tensor Factorization, G-NTF. It is enhanced NTF which introduces the idea of Graph Regularization. G-NTF decomposes the k-dimensional tensor formulated as follows;

$$\begin{array}{lll} \min & L = & \|\boldsymbol{X'} - \sum_{d=1}^{D} \boldsymbol{A}_{(1)\cdot d} \circ \cdots \circ \boldsymbol{A}_{(K)\cdot d}\|^2 \\ & & + \frac{\lambda_{(1)}}{2} \sum_{ij} \|\boldsymbol{A}_{(1)_{i\cdot}} - \boldsymbol{A}_{(1)_{j\cdot}}\| \boldsymbol{W}_{ij}^{(1)} \\ & & + \cdots + \frac{\lambda_{(K)}}{2} \sum_{ij} \|\boldsymbol{A}_{(K)_{i\cdot}} - \boldsymbol{A}_{(K)_{j\cdot}}\| \boldsymbol{W}_{ij}^{(K)} \\ \text{s.t.} & & \boldsymbol{A}_{(1)}, \cdots, \boldsymbol{A}_{(K)} \geq 0 \end{array} \tag{6}$$

When $k = 2$, the formulation is same as graph regularized NMF. And practically, k is assumed to be $k = 3, 4$ or so.

3.1 Updating Formula and Algorithm

To optimize Eq.(6), we propose the algorithm which updates each value of factor matrices repeatedly. Each value is updated as optimizing so that other values are fixed.

First, we focus on updating the values of matrix $\boldsymbol{A}_{(1)}$. If we pay attention to matrix $\boldsymbol{A}_{(1)}$, Eq.(6) is able to rewrite as follows;

$$\begin{array}{lll} \min & L = & \|\mathcal{X'}_{(1)} - \sum_{d=1}^{D} \boldsymbol{A}_{(1)\cdot d} \circ \boldsymbol{V}_{(1)\cdot d}^{T}\|^2 \\ & & + \lambda_{(1)} \text{trace}\left(\boldsymbol{A}_{(1)}^T \boldsymbol{L}^1 \boldsymbol{A}_{(1)}\right) \\ & = & \|\mathcal{X'}_{(1)} - \boldsymbol{A}_{(1)} \mathcal{V}_{(1)}^T\|^2 \\ & & + \lambda_{(1)} \text{trace}\left(\boldsymbol{A}_{(1)}^T \boldsymbol{L}^1 \boldsymbol{A}_{(1)}\right) \\ \text{s.t.} & & \boldsymbol{A}_{(1)} \geq 0 \end{array} \tag{7}$$

where $\mathcal{X'}_{(1)}$ is a mode-(1) flatten tensor, $\mathcal{X'}_{(1)} \in \mathbb{R}^{N_1 \times \prod_{i \neq 1} N_i}$, which refers to $\boldsymbol{X'}$ (General cases will be defined later), $\boldsymbol{V}_{(1)}$ is defined as

$$\boldsymbol{V}_{(1)\cdot d} = \left(\boldsymbol{A}_{(2)\cdot d} \otimes \left(\cdots \otimes \left(\boldsymbol{A}_{(K-1)\cdot d} \otimes \boldsymbol{A}_{(K)\cdot d}\right) \cdots\right)\right)$$

where the operator $\otimes$ represents Kronecker product, $\mathcal{V}_{(1)}$ is defined as

$$\mathcal{V}_{(1)} = \left(\left(\cdots \left(\boldsymbol{A}_{(K)} \odot \boldsymbol{A}_{(K-1)}\right) \odot \cdots\right) \odot \boldsymbol{A}_{(2)}\right),$$

where the operator $\odot$ represents Khatri-Rao product.

Now, objective function of Eq.(7) is considered as objective function of GWNMF (Eq.(2)) respect to A

$$\begin{array}{lll} \min & L = & \|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{B}^T\|^2 \\ & & + \lambda_{(1)} \text{trace}\left(\boldsymbol{A}^T \boldsymbol{L}^1 \boldsymbol{A}\right) \end{array} \tag{8}$$

Then we can obtain updating formula

$$A_{(1)_{n_1 d}} \leftarrow A_{(1)_{n_1 d}} \sqrt{\frac{\left[\mathcal{X'}_{(1)}\mathcal{V}_{(1)} + \lambda_{(1)}\boldsymbol{L}^{-}_{(1)}\boldsymbol{A}_{(1)}\right]_{n_1 d}}{\left[(\boldsymbol{A}_{(1)}\mathcal{V}^{T}_{(1)})\mathcal{V}_{(1)} + \lambda_{(1)}\boldsymbol{L}^{+}_{(1)}\boldsymbol{A}_{(1)}\right]_{n_1 d}}}. \tag{9}$$

Similarly, with respect to $A_{(k)}$, we can use updating formulas as follows;

$$A_{(k)_{n_k d}} \leftarrow A_{(k)_{n_k d}} \sqrt{\frac{\left[\mathcal{X'}_{(k)}\mathcal{V}_{(k)} + \lambda_{(k)}\boldsymbol{L}^{-}_{(k)}\boldsymbol{A}_{(k)}\right]_{n_k d}}{\left[(\boldsymbol{A}_{(k)}\mathcal{V}^{T}_{(k)})\mathcal{V}_{(k)} + \lambda_{(k)}\boldsymbol{L}^{+}_{(k)}\boldsymbol{A}_{(k)}\right]_{n_k d}}} \tag{10}$$

where

$$\{\mathcal{X'}_{(k)}\}_{i,j} = X'_{I_1,\cdots,I_{i-1},i,I_{i+1},\cdots,I_K},$$

$$I_s = \begin{cases} \left(j/\left(\prod_{l\neq 1,\ldots,s,k,\ldots,K} N_l\right) + 1\right)\% N_s & \text{for } s < k \\ \left(j/\left(\prod_{l\neq k,\ldots,s} N_l\right) + 1\right)\% N_s & \text{for } s > k \end{cases},$$

$$\mathcal{V}_{(k)} = \left(\left(\cdots\left(\left(\left(\cdots\left(\boldsymbol{A}_{(k-1)} \odot \boldsymbol{A}_{(k-2)}\right) \odot \cdots\right) \odot \boldsymbol{A}_{(1)}\right) \odot \boldsymbol{A}_{(K)}\right) \odot \cdots\right) \odot \boldsymbol{A}_{(k+1)}\right).$$

In the definition of I_s, operator/ represents a quotient, operator % represents a remainder, and if the remainder is equol to zero it is considered as $I_s = N_s$.

3.2 Convergence

As Gu et al. proved the convergence of the algorithm for GWNMF, the algorithm for G-NTF using updating formula, Eq.(10), converges.

Definition 3.1. *$G(h, h')$ is an auxiliary function for $F(h)$ if the conditions $G(h, h') \geq F(h), G(h, h) = F(f)$ are satisfied.*

Theorem 3.1. *Let*

$$F(\boldsymbol{A}_{(k)}) = \operatorname{trace}\Big(\lambda_{(k)}\boldsymbol{A}^{T}_{(k)}\boldsymbol{L}_{(k)}\boldsymbol{A}_{(k)} - 2\mathcal{X'}_{(k)}\mathcal{V}_{(k)}\boldsymbol{A}^{T}_{(k)} + (\boldsymbol{A}_{(k)}\mathcal{V}^{T}_{(k)})\mathcal{V}_{(k)}\boldsymbol{A}^{T}_{(k)}\Big)$$

Then the following function

$$\begin{aligned} &G(\boldsymbol{A}, \boldsymbol{A}') \\ &= \lambda_{(k)} \textstyle\sum_{ij} \frac{(\boldsymbol{L}^{+}_{(k)}\boldsymbol{A}'_{(k)})_{ij}\boldsymbol{A}^2_{ij}}{\boldsymbol{A}^2_{ij}} \\ &-\lambda_{(k)} \textstyle\sum_{ijl}(\boldsymbol{L}^{-}_{(k)})_{jl}\boldsymbol{A}_{(k)}{}'_{ji}\boldsymbol{A}_{(k)}{}'_{li}\left(1 + \log\frac{\boldsymbol{A}_{(k)}{}_{ji}\boldsymbol{A}_{(k)}{}_{li}}{\boldsymbol{A}_{(k)}{}'_{ji}\boldsymbol{A}_{(k)}{}'_{li}}\right) \\ &-2\textstyle\sum_{ij}(\mathcal{X'}_{(k)}\mathcal{V}_{(k)})_{ij}\boldsymbol{A}_{(k)}{}'_{ij}\left(1 + \log\frac{\boldsymbol{A}_{(k)}{}_{ij}}{\boldsymbol{A}_{(k)}{}'_{ij}}\right) \\ &+\textstyle\sum_{ij}\frac{\left((\boldsymbol{A}'_{(k)}\mathcal{V}^{T}_{(k)})\mathcal{V}_{(k)}\right)_{ij}\boldsymbol{A}_{(k)}{}^{2}_{ij}}{\boldsymbol{A}_{(k)}{}'_{ij}} \end{aligned}$$

is an auxiliary function for $F(\boldsymbol{A}_{(k)})$. Furthermor, it is a convex function in $\boldsymbol{A}_{(K)}$ and its global minimum is given by the right side of Eq.(10)

Proof. As against $\boldsymbol{A}_{(k)} \in \mathbb{R}^{N_k \times D}$, $\boldsymbol{L}^{+}_{(k)}$ is matrix $\mathbb{R}^{N_k \times N_k}$ and symmetric because of the definition. Also, $\mathcal{V}^T_{(k)}\mathcal{V}_{(k)}$ is matrix $\mathbb{R}^{d \times d}$ and clearly symmetric. Therefore, we can use the approach in [2]. □

Theorem 3.2. *For any k, updating $\boldsymbol{A}_{(k)}$ using Eq.(10) will monotonically decrease the value of the objective in Eq.(6), hence it converge.*

Proof. By theorem3.1 and lemma in [3] which is naturaly derived, we can get

$$F(\boldsymbol{A}^0_{(k)}) = G(\boldsymbol{A}^0_{(k)}, \boldsymbol{A}^0_{(k)}) \geq G(\boldsymbol{A}^1_{(k)}, \boldsymbol{A}^0_{(k)}) \geq F(\boldsymbol{A}^1_{(k)}) \geq \cdots$$

Therefore, $F(\boldsymbol{A}_{(k)})$ is monotonically decreasing. Since $F(\boldsymbol{A}_{(k)})$ is obviously bounded below, we prove the theorem. □

4 Experiments

We carried out three experiments with real apparel e-commerce data, and we confirmed that G-NTF was the practicality method.

4.1 Datasets

We used the order history data which was corrected on the one of the most famous fashion e-commerce sites in Japan. It is recorded from April 2015 to March 2016. The data consists of Order-table, Order-detail-table, User-table and Item table. Order-table and Order-detail-table have approximately 1M records of orders, who bought what for how much. User-table is a list of 100K users and including their age, sex, etc. Item table is a list of approximately 760K items and has large classification, small classification (we call it "Category", or later), color, etc.

In general, users of the e-commerce site have a price palatability. In other words, there are users who buy cheap items in any categories, and other users who buy expensive items in any categories. In the dataset, a similar behavior was observed. There are users who buy only items in high price range or who buy only items in low price range. Therefore, we try to recommend items which have suitable prices for users.

4.2 Pre-processing & Settings

Based on the idea in 4.1 we prepared the 3-dimensional tensor whose axes represent Users, Categories and PriceRanges respectively, and each element of it represents "who bought what for how much". We also prepared the 2-dimensional matrix to test NMF and GWNMF. Axes of the matrix represent Users and combinations of Categories and PriceRanges. Note that, it is same as a mode-User flatten tensor. Each element of this matrix also represents "who bought

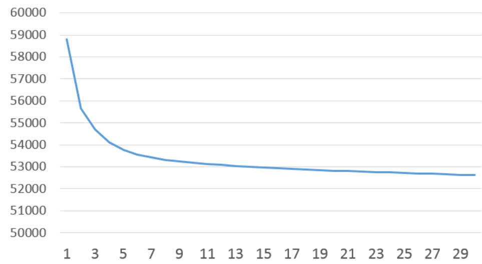

Figure 1: Convergence of G-NTF's objective function value

what for how much", but even if two columns represent same category, if they are different price ranges, they are considered as completely different categories. Therefore, they are thought to perform worse than methods which use the tensor directly.

To estimate methods, we recommend each user combinate units of the {Category-PriceRange} which weren't bought by the user in training term, but corresponding element in the reconstructed tensor or matrix has top-k high value in the part of the tensor or the matrix related to the user. Having high value means that corresponding unit is predicted likely to be bought.

To set similarity between users, we separated users into 12 groups according to sex and age. Then we calculated cosine similarity of each groups' purchase vector. Similarities between users were set by similarities of groups which they belong. Similarities between categories were set by they belong to the same large classification or not. Similarities between pricerange were set by how close they are.

We divided the data into training data and test data. Only training data was used in a training phase. To evaluate methods, we used annual average of F1 scores of top-k recommendation for test data. F1 score is harmonic mean precision and recall. Precision is a measure of how many recommended unit was bought in fact in test term. Recall is a measure of how many units which are bought in fact in test term was recommended.

4.3 Results

First, Figure 1 shows the result of experiment 1, a convergence of G-NTF's objective function value. The training term was set from April to June. The objective value converged when the number of iteration is 25 or more. Therefore, in following experiments, we set the number of iteration to 50 with margin.

Table 1 shows the result of experiment 2 to compare methods. Training term was set at 3 months, and test term was set at 1 month. We focus on the presence of graph regularization. Because of GWNMF and G-NTF showed higher performance than NMF and NTF respectively, graph regularization has a good impact on F1 scores. Next we focus on form of data. Because of NTF and G-NTF showed higher performance than NMF and GWNMF respectively, using

Table 1: F1 scores of methods with a different number to recommend and execution time for train

Method	top-1	top-3	top-5	time(s)
NMF	0.041	0.076	0.088	519
GWNMF	0.046	0.085	0.097	581
NTF	0.044	0.078	0.091	1247
G-NTF	0.052	0.093	0.107	1276

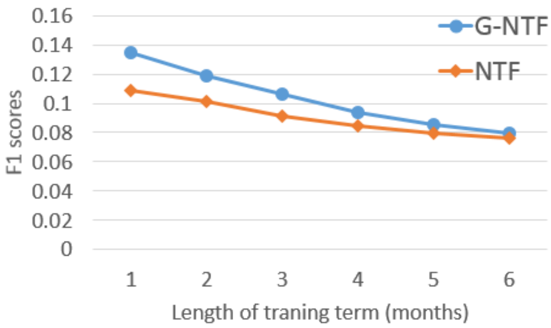

Figure 2: F1 scores of G-NTF and NTF with a different length of training terms.

tensors is meaningful in general. In conclusion, our method G-NTF showed the highest performance. Note that, methods using tensors took more time than methods using matrices respectively. However, the execution time of G-NTF is feasible.

Figure 2 shows the result of experiment 3. We changed training terms from 1 month to 6 months, and test term was set at 1 month. With the shorter training term, both NTF and G-NTF performed better than the case with the longer training term. Trends and especially seasonality are reasons of this. Apparel items have those, so it is important to recommend fashionable items. The shorter the training term, G-NTF outdo NTF. It's mean that the smaller the data is, the stronger similarity affects. From these results, we can know that regularizing by similarity is meaningful to recommend with high performance.

5 Conclusions

In this paper, we have proposed the new recommendation algorithm, G-NTF. It is the enhancement of NTF with idea of Graph regularization [2]. By using flatten tensors, we gave the algorithm updating formula which was proved convergence. And the algorithm shows high recommendation performance than NMF, GWNMF or neutral NTF, in the experiment with real order history data of the apparel e-commerce site. Especially, the less training data it has, the higher performance it shows. It is practically useful well.

Acknowledgment

Thanks to the JASMAC and the firm which is operating the e-commerce site for preparing the data.

References

[1] Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari. *Non-negative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

[2] Quanquan Gu, Jie Zhou, and Chris Ding. Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs. In *Proceedings of the 2010 SIAM International Conference on Data Mining*, pages 199–210. Society for Industrial and Applied Mathematics, April 2010.

[3] Daniel D. Lee and H.S. Seung. Algorithms for non-negative matrix factorizatiojn. In *Advances in neural information procesing system*, pages 556–562, 2001.

[4] Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80, 2003.

[5] Xin Luo, Mengchu Zhou, Yunni Xia, and Zhu Qingsheng. An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. *IEEE Transactions on Industrial Informatics*, 10(2):1273–1284, 2014.