DEIM Forum 2017 G5-3

A Gradient Descent Method for Matrix Factorization Using Latent Dirichlet Allocation

Guan-shen FANG†, Kamei SAYAKA††, and Satoshi FUJITA††

† Information Engineering, Graduate School of Engineering, Hiroshima University,
1-3-2 Kagamiyama, Higashi-Hiroshima City Hiroshima, Japan
†† Department of Information Engineering, Faculty of Engineering, Hiroshima University
1-3-2 Kagamiyama, Higashi-Hiroshima City, Hiroshima, Japan
E-mail: †{bou,s-kamei,fujita}@se.hiroshima-u.ac.jp

Abstract Recommendation of products plays a significant role in online service and social network nowadays. In order to discover unpurchased ones that a user may like, a consideration is to predict how he evaluates them after his purchasing according to his history feedbacks. Such feedback often includes a couple of numeric rating and textual review for a specific purchased product. A traditional technique to model user's rating is matrix factorization, which uses latent factors to characterize both user's interests and product's properties. However, its ignorance of review brings some drawbacks, including incorrect latent factors learnt and mediocre performance. In this paper, we aim to combine the analysis of review into traditional matrix factorization and improve the learning of latent factors. The experimental results show that by using proposed method in matrix factorization model, latent factors are more accurately learnt and a more accurate prediction of rating for unpurchased products is realized. In the scene of prediction for unpurchased item's rating, the improvement in term of Mean Absolute Error is up to 3.756%.

Key words gradient descent, recommender system, topic model, Matrix Factorization

1. Introduction

As an efficient way to communicate with their users, recommender system plays a significant role in online service and social network nowadays. In order to discover and provide users the products that they potentially be interested and buy in future, a consideration is to predict how they assess unpurchased products. The most well-known approach is Collaborative Filtering (CF) [1]. It assumes that users sharing similar opinions to their common products in the past, are also likely to have similar evaluation for a certain product in the future. Among all the CF algorithms, latent factor models [2], [5] are the most successful ones. It characterizes both products and users by vectors of latent factors, which comprise computerized alternatives to the human created genres. In order to infer such factors by machine learning algorithms, it uses the ratings in users' feedbacks, which typically include couples of rating and review. High correspondence between user and new product leads to a high predicted rating — a recommendation.

However, recent researches pointed out that the ignorance of textual reviews in the feedbacks is the major shortcoming of traditional latent factor models [6], [7], [9]. A rating in a user's feedback only represents whether he likes or dislikes

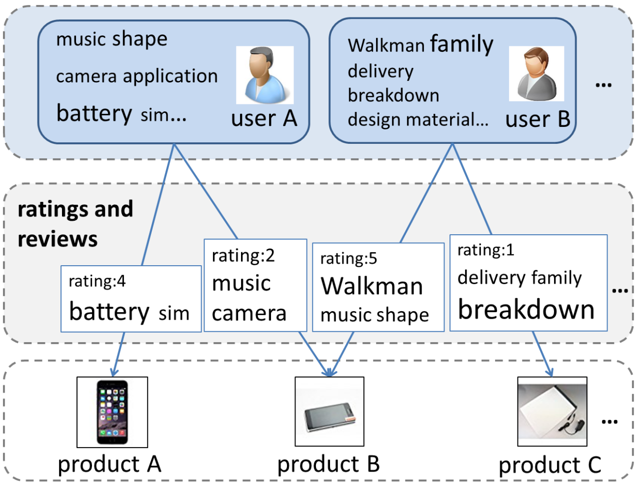

Figure 1 A graph characterized the actual topics in reviews of two users to three products on Amazon. For a review, the size of characters illustrates the proportion of the topic; for a user, it represents how much the user mentions it in all reviews.

the product, but cannot further explain the reasons in which he rates. This is the meanning of "latent": only using history ratings we cannot establish the definit correspondance between latent factors and actual topics which represent obvious genres, or properties of products. Therefore, it forces traditional models to treat each of latent factors equally in their learning phase. In contrast, there is a gap with the actual case that users give their ratings. Users often has their own overall opinion (i.e. like or dislike) to each factor, but for a specifc product, they may only consider a part of them when they rate and write the reviews. Examples are shown in Figure 1, which represents two users and the topics in their reviews to three products. For product B, user A made a low rating and pointed out the bad qualities of music's playing and camera; conversely, user B rated highly and his important reason is that he is a fan of the product's maker. It seems like there is a degree of importance for each of the topics, which is often adjusted case by case. Due to such gap, the latent factors learnt only by ratings are not correct. More seriously, such incorrect latent factors lead to mediocre performance in missing rating's prediction for user. Therefore, it is necessary to understand the latent factors and realize such adjustments in their learning.

In this paper, in order to solve the issue mentioned above, we aim to use analysis of reviews to enhance traditional matrix factorization (MF). Our objective is to relax the constraint that equally treats latent factors, and further improve the performance of MF. We use Latent Dirichlet Allocation (LDA) to model the topics that users summarized in their reviews. Such topics always represent the reasons why they gave the corresponding rating. Therefore, the topic proportion in a review represents its importance of degree aforementioned in user's consideration of rating. Rather than integrating LDA and transforming such topics into latent factors in existing research [9], we propose a new gradient descent method via LDA. Our idea is that the error of prediction with actual rating is composited by portions of topics, the more important the user considers a topic and mentioned in review, the more error the topic "contributed". Therefore, we relate each latent factor with a topic, and individually assign the learning rate in its learning. In other words, we simply let each latent factor characterize the interest of user/quality of item for one topic, and accurately update its value in learning process. Such method has various advantages: 1) one topic corresponds with one factor and interpret it; 2) with such relation each factor is accurately learnt, and further provide accurate prediction of rating, and 3) comparing with the approach which integrate topic model into MF, it is easy to implement and time saving in learning process.

In evaluation, we conducte a series of experiments using real-world actual datasets. The results show that in the problem of missing rating's prediction, when using proposed method in optimization of objective function and update the latent factor vectors both for user and item, MF gains an improvement up to 3.756% in terms of Mean Absolute Error. Comparing with existing approach [6] which both model the rating and topics in the same time, proposed method make MF outperform it with an improvement up to 28.442%.

The remainder of this paper is organized as follows. Section 2. overviews related works of latent factor models. Section 3. describes the problem that we focus and briefly reviews LDA and MF. Section 4. describes the detail of proposed method. Section 5. represents the method of evaluation and shows its results. Finally, Section 6. concludes the paper with future work.

2. Related Work

In recent years, latent factor-based approaches are popular for their efficiencies in dealing large scale datasource. In order to model users' ratings for further prediction, they transfer the approximation of user-item rating matrix into minimizing the sum-squared errors between actual and predicted ratings, and solve it using Singular Value Decomposition [4]. Salakhutdinov *et al.* [5] proposed Probabilistic Matrix Factorization model (PMF) and introduces Gaussian priors as hyperparameters to present latent factor vectors. They also pointed out that maximizing the log-posterior over items and user's latent factors with hyperparameters kept fixed is equivalent to minimizing the sum-of-squared-errors objective function. Both of these approaches ignore the reviews in users feedback, which is pointed out as a significant drawback.

In order to improve the interpretive ability of recommendation result, efforts are made in combining semantic analysis of reviews or introductory essays of products. Wang *et al.* [6] proposed Collaborative Topic Regression (CTR), a latent factor-based model combining PMF and LDA for scientific articles recommendation. For each article, its topic proportion is infered as a stochastic vector using its title and abstract. Each of the elements in such vector is added a latent variable, to form the article's latent vector. In their learning algorithm, the latent vectors for user and article and its topic proportion are learnt dependently. To solve the problem of many users have few feedbacks, namely data sparsity of users, Purushotham *et al.* [7] improved CTR with integration social network structure, i.e. follower relationship of users. They transform the corelation of users into social network matrix. For such matrix is approximated by the inner product of user latent matrix and social factor feature matrix by matrix factorization. In other words, user latent matrix is shared in decomposition of social network matrix and rating matrix. On the other hand, Wang *et al.* [8] proposed Collaborative Deep Learning based on CTR, to improve its performance. Instead of LDA, they use Stacked Denoising Autoencoder to infer the topic proportion of scientific article or semantic introduction of item. These CTR-based method directly use topic proportion of item as stochastic vector instead of latent factor vector, or to combine with latent variables to form such vector in rating matrix's approximation.

Another consideration in combining latent factor model and topic model is to transform the topic proportion of item into its latent factor vector. McAuley *et al.* [9] developed a statistical model named 'Hidden Factors of Topics', trying to use topics to interpret latent factors in MF. For a specific item, its related reviews are defined as a single document, which is represented by topics having the same size to latent factors. Therefore, the latent factors having been infered are normalized by introducing a parameter as the stochastic topic proportion directly. Zhang *et al.* [10] further analyzed sentiment contained in reviews in their explicit factor model. Addition to implicit (latent) factors, they extract the nouns represent properties of item, as explicit feature. After that, the polarities of such explicit features are extracted from reviews to form two explicit feature matrics for both size of user and item. In their model's learning, the user and item latent factors are not only to fit the rating matrix, but also have to fit such sentiment explicit feature matrics.

Although in state-of-the-art approaches semantic analysis of review are integrated to gain better performance and recommendation's interpretive ability than traditional MF, they focus on the "document" formed by reviews published by individual user or related with single item. In other words, they can not cope with the case that individual user may adjusts the importances of factors when he faces to different items.

3. Preliminaries

3.1 Problem Definition

The problem we study is to accurately learn latent factor vector for user and item, to provide the prediction to missing ratings for users by their history feedbacks. Such rating is often in scale of $[a, b]$ (e.g. ratings of one to five stars on Amazon), and couples with a semantic review. Suppose we have I users and J items, and the rating made by user u_i to item v_j is denoted as $r_{i,j}$, its corresponding review is denoted by $d_{i,j}$ respectively. We assume that $r_{i,j}$ must have a $d_{i,j}$ having been written. Therefore, a feedback is a 4-tuple $(u_i, v_j, r_{i,j}, d_{i,j})$. For user u_i, U_i denotes its latent factor vector, and V_j denotes the one for item v_j.

3.2 Matrix Factorization for Recommendation

Matrix Factorization [2] is a basic but successful method to predict the missing rating in recommendation. It mapps users and items into a joint latent factor space with K dimensions. Accordingly, each user u_i is associated with a vector $U_i \in \mathbb{R}^K$, and each item v_j is associated with a vector $V_j \in \mathbb{R}^K$. The elements in U_i measure the extent of interest of u_i of such factors, on the other hand, the elements of V_j measure the extent to which the item possesses those factors with positive or negative. Therefore, the product of U_i and V_j represents the interaction of u_i and v_j, further approximates the corresponding rating $r_{i,j}$, leading to the estimate:

$$\hat{r}_{i,j} = U_i^T V_j + \mu + b_i + b_j \tag{1}$$

where b_i and b_j denote the observed bias of rating of user u_i and item v_j respectively, and μ is the average overall rating.

The objective is to learn U_i and V_j by given training set including observed ratings, by minimizing the regularized squared error loss:

$$\begin{aligned}\mathcal{L} = \min_{U,V} \sum_{i,j} &[c_{i,j}(r_{i,j} - \hat{r}_{i,j})^2 \\ &+\lambda(\|U_i\|^2 + \|V_j\|^2 + {b_i}^2 + {b_j}^2)]\end{aligned} \tag{2}$$

where λ is the parameter to controll the regularization to avoid over-fitting in learning, and $\|\cdot\|^2$ denotes the L^2 norm. $c_{i,j}$ is a confidence parameter of rating $r_{i,j}$, which indicated how much we trust it. In other words, a large $c_{i,j}$ means that $r_{i,j}$ is a deliberate rating so that we trust it more; a small $c_{i,j}$ should be assigned to some observed rating that do not deserve seriously treatment such as advertisings.

In order to optimize Equation (2), gradient descent algorithm [11] is used to loop through all ratings in training set. U_i and V_j are updated by modifing them to the opposite direction of the gradient of error:

$$\begin{aligned}U_i &\leftarrow U_i + \gamma[c_{i,j}(r_{i,j} - \hat{r}_{i,j})V_j - \lambda U_i] \\ V_j &\leftarrow V_j + \gamma[c_{i,j}(r_{i,j} - \hat{r}_{i,j})U_i - \lambda V_j]\end{aligned} \tag{3}$$

where γ is the learning rate. With U_i and V_j updated alternately, a missing rating can be predicted following Equation (1).

3.3 Latent Dirichlet Allocation (LDA)

Topic model is the algorithm used to discover topics in a large set of documents, i.e., corpus. Such topic is a probability distribution over all words. The words associate with a single theme make their corresonding elements bias in the distribution of a topic. Therefore, topic model provides an interpretable dimension reduction of corpus.

LDA [3] is a generative probabilistic topic model of a set of semantic documents. Its basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. Assume there are K topics in corpus, and one topic t_k ($k \in [1, \ldots, K]$) is a probability distribution of all words. For each document $\mathbf{w}_j$ in corpus, the generative process is as follows:

- Choose topic proportion θ_j of $\mathbf{w}_j \sim \mathrm{Dir}(\alpha)$

- For each of the words w_n:
  – Choose a topic assignment $z_{j,n} \sim$ Multinomial(θ_j)
  – Choose a word $w_n \sim$ Multinomial$(\beta_{z_{j,n}})$

The process represents the assumption how a document is generated. Overall topic size K is assumed to be known and t_k is shared by corpus. For each document $\mathbf{w}_j$, a specific topic proportion θ_j is provide as its representation.

The objective is to estimate the maximun likelihood of β_k and α to generate the documents of corpus. We approximate them by using an variational EM algorithm [3] which maximizes their lower bound. Further, the parameters θ_j and $z_{j,n}$ can be updated via Gibbs sampling [12] iteratively.

4. Proposed Method

In our method, we use LDA to uncover the latent topics in user's review text. Different with existing method [8], [9], we propose a new gradient descent method to correlate the topics to latent factors. Our idea is to adjust the learning rate with topics' proportions, to unequally treat each latent factor for a specific rating.

Firstly, each review in users' feedbacks is taken as a single document and all reviews form the corpus. With such corpus, we independently train LDA to infer the topic distribution $\theta_{i,j}$ for review $d_{i,j}$, which is published by user u_i to item v_j. Suppose we have K topics overall, and let $\theta_{i,j}^k$ denote the proportion of k-th topic in $\theta_{i,j}$. Thus $\theta_{i,j}^k \in [0,1]$ and $k \in 1,\ldots,K$. Since $d_{i,j}$ summarizes the reasons that u_i gives the corresponding rating $r_{i,j}$, $\theta_{i,j}^k$ represents the importance of topic t_k in $r_{i,j}$. In other words, for specific v_j, the topics that u_i does not care or considers it normal are not mentioned in $d_{i,j}$. On the other hand, if u_i considers t_k very important for v_j, he will mention it more in $d_{i,j}$ and adjust $r_{i,j}$ intenser from his base line than other topics.

Although we can simply introduce the importance of topics into rating's modeling following the estimation

$$\hat{r}_{i,j} = U_i^T H_{i,j} V_j + \mu + b_i + b_j$$

where $H_{i,j}$ is a $K \times K$ diagonal matrix with $\theta_{i,j}$ as its diagonal elements, for unpurchased item v_j', $H_{i,j'}$ is unknown and hard to predict. As the solution, we re-define the learning rate γ in Equation (3) as

$$\gamma = \alpha\theta_{i,j}^k$$

where α is a constant. Therefore, for the K-dimensional latent factor vectors U_i and V_j, the update equation (3) of the elements U_i^k and V_j^k are re-writed as

$$\begin{aligned} U_i^k &\leftarrow U_i^k + \alpha\theta_{i,j}^k[c_{i,j}(r_{i,j} - \hat{r}_{i,j})V_j^k - \lambda U_i^k] \\ V_j^k &\leftarrow V_j^k + \alpha\theta_{i,j}^k[c_{i,j}(r_{i,j} - \hat{r}_{i,j})U_i^k - \lambda V_j^k] \end{aligned} \tag{4}$$

where for each pair of elements U_i^k and V_j^k we individually determine their learning rates. Note that $\hat{r}_{i,j}$ is calculated following Equation (1). It follows the idea that 1) since latent factors has no definit correspondance with actual topics, we force each of its elements to relate with a topic; 2) the rating error $r_{i,j} - \hat{r}_{i,j}$ is composited by the portions of topics with importances $\theta_{i,j}^k$. In other words, the important the topic considered in $r_{i,j}$, the more its corresponding factors are updated. For the omitted topics in $d_{i,j}$ since they do not produce error, we do not update their corresponding factors respectively. Another issues for us is to determine a proper value for constant α. Comparing with ordinary MF which only uses a constant learning rate, since the average of $\theta_{i,j}^k$ is $1/K$, in proposed method the step of updating in each loop reduces to $1/K$ overall. Thus, we assign $K\gamma$ for α, where γ is the learning rate used in MF. For example, if the learning rate γ was 0.01, for proposed method when $K = 5$, we set $\alpha = 0.05$.

After the optimization of each U_i and V_j, we can use them to predict the missing ratings of users following Equation (1) of MF.

Table 1 The description of dataset used in experiments.

dataset	Yelp	automotive	digital music
♯users	1114	369	4202
♯items	5781	369	3279
♯ratings	16014	2549	52500
average ratings	3.81	4.33	4.216
average ♯words of review	134.8	188	234

5. Evaluation

In order to evaluate the performance of proposed method, we conduct several experiments to compare with existing methods by using real-world datasets. We aim to answer the questions of 1) whether the loss (i.e. square error in Equation (2)) rapidly converge in learning process by using proposed method; 2) whether proposed method can improve MF to gain a better performance in the scene of rating's prediction.

5.1 Datasets

As shown in Table 1, we use three datasets provided by Yelp and Amazon in our experiments. For Yelp dataset(注1), we focus on the feedbacks that relate with the POIs in Arizona, USA, and are published in 2014. For Amazon datasets provided by McAuley [13], we use the feedbacks of categories of "automotive" and "digital music". All datasets are filtered by constrains of 1) users who at least have 5 feedbacks; 2)

(注1)：http://www.yelp.com/dataset_challenge

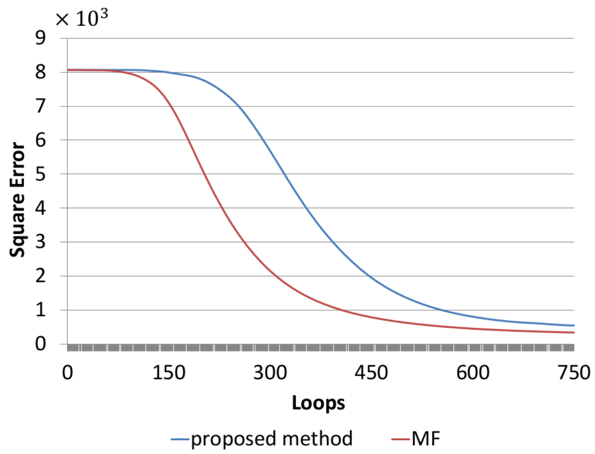

Figure 2 The square error in learning process on dataset.

items that is reviewed by at least 3 users; 3) the feedbacks including a review that has at least 10 nouns. In the last line of Table 1, we list the average size of words included in reviews. It indicates that reviews are well summarized by users, and will not affect the performance of inference of topics. For each dataset, we randomly take 80% of its feedbacks as training set, and the rest as testing set. In order to independently train LDA, for each review in the training set, we extract the nouns as words, and conduct a stemming to each words.

5.2 Convergence of Square Error

Firstly, we simply conduct learning processes both using proposed method and the method [11] in traditional MF by using YELP dataset, with $K = 5$ and $\lambda = 0.01$. In each loop to update U_i and V_j we observe the square error. For traditional MF, we set U_i and V_j with randomly generated values following the zero mean gaussian distribution of $\mathcal{N}(0, \lambda^2)$. For proposed method, we initially set a extremely small unique value of 0.001 for the elements in U_i and V_j. The reason is that such assignment makes the initially prediction of rating be the average of ratings, and minimize the effect of initialization. The learning rate for traditional MF is set to 0.01, and 0.05 for proposed method since $K = 5$.

Figure 2 summarized the square error of traditional MF and proposed method in learning process. It shows that comparing with existing gradient descent method [11], the convergence of square error of proposed method is more slower, but the step of updating is almost same. The reason is considered that the initial value is small and learning rate is set conservatively. On the other hand, the minimum of square error is also higher than existing method. This is because that for the topics having been seldom mentioned in reviews, their corresponding factors are also seldom updated so that they remain the initial values and hard to reduce.

5.3 Impact to Recommendation

In this section, we evaluate the performance of proposed method when it is used as optimization method for latent factor model.

5.3.1 Comparisons and Implementation

We select traditional MF [4] and Collaborative Topic Regression (CTR) [6] as baselines. For both of them, latent factors are optimized by the gradient descent methods following their researches. Additionally, in order to evaluate the performance of proposed method, we use proposed method to learn latent factors of MF instead of [11], as the improvement of traditional MF. Especially, the initial values for elements in U_i and V_j are fixed to 0.001. For traditional MF we tried several values for λ and finally fix it to 0.001. On the other hand, we set $c_{i,j} = 1$ if u_i have reviewed and rated v_j, and $c_{i,j} = 0$ otherwise. For CTR, since it analysis the topics in textual introduction essay of item instead of the individual review, we take the reviews related to an item as a single document as its introduction essay. Following the parameters' setting in their research, we set λ_u equals to 0.01. At last, we fix K to 5 and 10 overall to observe their performances. Also note that both for YELP and Amazon datasets the ratings is in scale of $[1, 5]$, thus we limit the predicted rating in this range.

5.3.2 Evaluation Methodologies

For each dataset, we form and use a common training set for both proposed method and existing models in various K values. In training phase, the parameters U_i and V_j are iteratively learnt until the loss calculated by objective function convergences. After that, we evaluate the performances by observing the accuracy of prediction of the ratings in test set. As quantification, we use mean absolute error (MAE) and root mean square error (RMSE) which are calculated as follows:

$$MAE = \frac{1}{N}\sum_{i,j}(|r_{i,j} - \hat{r}_{i,j}|),$$
$$RMSE = \sqrt{\frac{1}{N}\sum_{i,j}(r_{i,j} - \hat{r}_{i,j})^2}$$

where N denotes the number of feedbacks in test set, and $|\cdot|$ denotes the absolute value. In general, RMSE is more sensitive for large prediction's errors than MAE.

5.4 Experimental Results

Table 2 shows the results of MAE and RMSE of traditional MF, CTR and MF using proposed method with K equals to 5 and 10. The emphasised numbers represents the lowest MAE and RMSE for each dataset. When $K = 5$, MF using proposed method shows the best performance for all datasets. Comparing with traditional MF, our proposed method improves it from 0.743% to 3.756% in terms of MAE, 1.727% to 3.361% in RMSE. When $K = 10$, for dataset 'digital music' traditional MF outperforms porposed method. It represents

Table 2 The performances of MF, CTR and proposed method in $K = 10$ and $K = 50$.

dataset	MF		CTR		proposed method		improvement of MAE		improvement of RMSE	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	vs. MF	vs.CTR	vs. MF	vs.CTR
$K = 5$										
YELP	1.0280	1.3649	1.3293	1.6910	**0.9970**	**1.3221**	3.015%	24.998%	3.135%	21.815%
digital music	0.8829	1.3027	0.9348	1.3673	**0.8497**	**1.2589**	3.756%	9.102%	3.361%	7.925%
automotive	0.7404	1.0712	1.027	1.3794	**0.7349**	**1.0527**	0.743%	28.442%	1.727%	23.684%
$K = 10$										
YELP	0.9734	1.2992	1.3920	1.7363	**0.9634**	**1.2918**	1.03%	30.79%	0.57%	-25.96%
digital music	**0.7935**	**1.1435**	0.8813	1.2276	0.8528	1.2305	-7.472%	3.228%	-7.604%	-0.236%
automotive	0.7363	1.0452	0.9749	1.2930	**0.7114**	**1.0299**	3.382%	27.028%	1.464%	20.348%

that for MF, our method are more affected by parameter K than existing gradient descent method [11]. Comparing with CTR, which is a model that integrates LDA into MF to improve the explanation of latent factor, by using our proposed method MF also shows better performances in most of the experiments. It indicates that our proposed method improves MF in its factor's learning process, and enhances it in the scene of rating's prediction.

Combining with the experimental result in Section 5. 2, in optimization of objective function Equation (2), although proposed method cannot reaches the same level of loss to existing gradient descent [11], it trains U_i and V_j more accurately.

6. Concluding Remarks

In this paper, we propose a gradient descent for the optimization of tradional latent factor model like MF. By using LDA to model the topics in user's review, we relate latent factors with topics and individually determine the learning rates via proportions of topics. The performance of optimization of proposed method is experimentally evaluated by a scene to use MF in prediction of unknown ratings of user. The results show that proposed method provides more accurately learnt latent factors. Overall, it indicates that our proposed method achieves the objectives to release the limination of equally treatment of latent factors, and improve the performance of MF.

In future work, we want to further improve the speed of optimization of proposed method. By considering the epoch in learning process, we can further adjust the learning rate to fasten the convergence of loss. On the other hand, since the running time of proposed method has not analyzed, we will calculate it and make the comparison with existing method.

References

[1] D. Goldberg, D. Nichols, B. M. Oki, D. Terry, "Using collaborative filtering to weave an information tapestry," Communications of the ACM, Vol. 35, No. 12, pp.61–70, 1992.

[2] Y. Koren, R. M.Bell and C.Volinsky, "Matrix factorization techniques for recommender systems," IEEE Computer, Vol. 42, No. 8, pp. 30–37. 2009.

[3] D. M. Blei, A. Y. Ng, M. I. Jordan, "Latent dirichlet allocation," Journal of machine Learning research, Vol. 3, No. Jan, pp. 993–1022, 2003.

[4] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 426–434, 2008.

[5] R. salakhutdinov, A. Mnih, "Probabilistic matrix factorization," NIPS, Vol. 20, pp. 1–8, 2011.

[6] C. Wang, D. M. Blei, "Collaborative topic modeling for recommending scientific articles," Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 448–456, 2011.

[7] S. Purushotham, Y. Liu, C. C. J. Kuo, "Collaborative Topic Regression with Social Matrix Factorization for Recommendation Systems," CoRR, Vol. abs/1206.4684, 2012.

[8] H. Wang, N. Wang, D. Y. Yeung, "Collaborative deep learning for recommender systems," Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1235–1244, 2015.

[9] J. McAuley, J. Leskovec, "Hidden factors and hidden topics: understanding rating dimensions with review text," Proceedings of the 7th ACM conference on Recommender systems, pp. 165–172, 2013

[10] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, S. Ma, "Explicit factor models for explainable recommendation based on phrase-level sentiment analysis," Proceedings of the 37th international ACM SIGIR conference on Research and development in information retrieval, pp. 83-92, 2014.

[11] S. Funk, "Netflix Update: Try This at Home," http://sifter.org/ simon/journal/20061211.html, 2006

[12] G. Heinrich, "Parameter estimation for text analysis," University of Leipzig, Tech. Rep, 2008

[13] J. McAuley, C. Targett, Q. Shi, A. van den Hengel, "Image-based recommendations on styles and substitutes," Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 43–52, 2015