Rating Prediction with Topic Gradient Descent Method for Matrix Factorization in Recommendation

Guan-Shen Fang, Sayaka Kamei, Satoshi Fujita
Department of Information Engineering
Hiroshima University
Hiroshima, Japan

***Abstract*—In many online review sites or social media, the users are encouraged to assign a numeric rating and write a textual review as feedback to each product that they have bought. Based on users' history of feedbacks, recommender systems predict how they assesses the unpurchased products to further discover the ones that they may like and buy in future. A traditional approach to predict the unknown ratings is matrix factorization, while it uses only the history of ratings included in the feedbacks. In recent researches, its ignorance of textual reviews is pointed out to be the drawback that brings mediocre performance. In order to solve such issue, we propose a method of rating prediction which uses both the ratings and reviews, including a new first-order gradient method for matrix factorization, named Topic Gradient Descent (TGD). The proposed method firstly derives the latent topics from the reviews via Latent Dirichlet Allocation. Each of the topics is characterized by a probability distribution of words and is assigned to correspond to a latent factor. Secondly, to predict the ratings of the users, it uses matrix factorizaiton which is trained by the proposed TGD method. In the training process, the updating step of each latent factor is dynamically assigned depending on the stochastic proportion of its corresponding topic in the review. In evaluation, we both use YELP challenge dataset and per-category Amazon review datasets. The experimental results show that the proposed method certainly converges the squared error of the prediction, and improves the performance of traditional matrix factorization up to 12.23%.**

Keywords*—Gradient descent; matrix factorization; Latent Dirichlet Allocation; information recommendation

I. Introduction

Nowadays, recommender systems play a significant role in online services and social networks to communicate with their users. In order to discover and provide the items (e.g. products or news, etc.) that users potentially are interested and buy in future, a consideration is to predict how they assess the unpurchased items based on their history of feedbacks. Such feedbacks are written by the users after their purchases, and each of them includes a numeric rating as the evaluation and a textual review. The most well-known approach to predict the ratings is Collaborative Filtering (CF) [1]. It assumes that users sharing similar evaluation to their common items in the past, are also likely to have similar evaluation for a certain item in the future. Among all the CF algorithms, latent factor-based matrix factorization (MF) is the most famous [2]–[4]. It characterizes both items and users by vectors of latent factors, which comprise computerized alternatives to the human created genres. The rating of a user to a specific item is modeled by the inner product of their factors. Using machine learning algorithm, the latent factors can be learnt based on the ratings in the history of feedbacks.

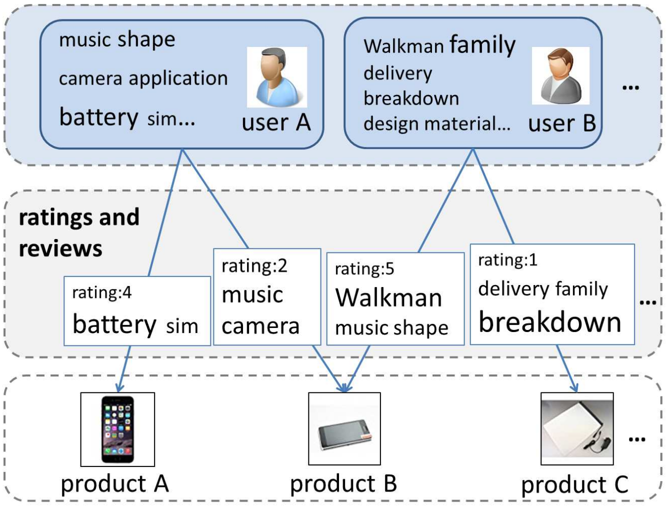

Fig. 1. A graph that characterizes the actual topics in reviews of two users to three products on Amazon. In reviews, each user mentioned many topics in which he is interested. For different products each user mentions different parts of interested topics in the reviews; for a specific product, different users focus on different associated topics.

However, recent researches pointed out that the ignorance of the reviews is the major shortcoming of MF and brings it mediocre performance [5]–[7]. Fig. 1 shows two users and their feedbacks to three products from Amazon. For product B (a music player), user A gives a low rating and points out the bad qualities of music's playing and camera; conversely, user B rates highly and his principal reason is that he is a fan of the player's maker. It represents the fact that although a user often has his own overall opinion (i.e. like or dislike) to obvious properties of product, he/she may focus only on a part of them in the evaluation. While the description of the properties are contained in the textual review, MF cannot realize such unequally treatment since the correspondence of them to latent factors are not defined. To bridge this gap, existing works [6], [7] model the latent topics of reviews with distributions of words, and transform them to latent factors. Unfortunately, the transformation is complicated and makes their methods time consuming in dealing with large-scale data.

In this paper, in order to solve the issues mentioned above, we propose a new method to predict the rating of user's unpurchased item for recommendation. In order to model the latent topics in the reviews, we train a Latent Dirichlet Allocation (LDA) [8] independently. Each of the topics is assigned to a latent factor. Our idea is to present a new first-order gradient descent method, called Topic Gradient Descent (TGD), which binds the latent topics to latent factors via the training process of MF instead of the transformation. Since a more mentioned topic in a review is considered more importantly when the user rates, its proportion to all topics represent the degree of importance. When iteratively updating its corresponding latent factor in the training, its updating step is dynamically fixed based on such importance. In other words, the importance of topics points out the direction to update the latent factor vectors of users and items. With these trained latent factor vectors, we use the biased MF to predict the ratings.

In our evaluation, we conducted a series of experiments using 11 datasets, including YELP challenge dataset and per-category reviews from Amazon. It evaluates not only the entire performance in the problem of missing rating's prediction, but also the convergence of the squared error of rating prediction in the training.

The contribution of this paper is as follows:

- It proposes a new gradient descent method named TGD, which provides a solution to introduce latent topics to MF, to release the unequally treatment of latent factors. The results of the experiment demonstrate that it certainly converges the objective function in MF's training. Furthermore, the speed of convergence depends on the dispersion of topics' proportions.
- TGD works well in the proposed method for rating prediction in recommendation. Comparing with simplex MF [2], the proposed system derives an improvement up to 9.03% in term of MAE, 12.23% in RMSE. It even outperforms state-of-the-art model [7] for recommendation in most of the datasets. Additionally, the proposed method is also demonstrated to have higher accuracy than simplex MF in the prediction of high-scored ratings.

The remainder of this paper is organized as follows: Section II overviews related works of latent factor models. Section III describes the problem that we focus and briefly reviews MF. Section IV describes the detail of proposed method. Section V represents the method of evaluation and shows its results. Finally, Section VI concludes the paper with future work.

II. Related Works

In recent years, latent factor-based approaches are popular for their efficiencies in handling large scale datasource. In order to model users' ratings for further prediction, they approximate user-item rating matrix by singular value decomposition [2], [9]. It decomposes the rating matrix into two orthogonal matrices, which represent the latent factors of users and items, respectively. In the training, the factors are often learnt via gradient descent method. For an unpurchased item of a given user, the rating prediction is made by calculating the inner product of their latent factors. Salakhutdinov *et al.* [9] proposed Probabilistic Matrix Factorization (PMF) and introduced Gaussian priors as hyperparameters to present latent factors. They noted that maximizing the log-posterior of the ratings over users' and items' latent factors is equivalent to minimizing the squared errors of the rating prediction. Hu *et al.* [10] applied latent factor model to the recommendation for implicit feedback datasets, which includes the indirectly reflect opinions, such as the purchase history of products and browsing history of the web pages. In the optimization of the model, they let the differentiation of user's and item's latent factors be zero, and recalculate the expression for them.Koren *et al.* [11] proposed a model for recommendation which integrates the latent factor model and traditional neighborhood model of CF. In their rating prediction, it directly sums up the predictions of these two models together. All these approaches ignore the reviews of users' feedbacks, which makes them scalability but mediocre accuracy in rating prediction.

To improve the performance of latent factor models, semantic analysis of textual summary of an item is introduced in many existing researches [5], [12], [13]. In the case of reviews of feedbacks, a common idea is to take the reviews correlated with individual item as one summary. Wang *et al.* [5] proposed a model named Collaborative Topic Regression (CTR), which combines PMF and LDA for recommendation of scientific articles. The latent topics of a given article are derived from its title and abstract. Their distribution and a set of parameters are together to form the latent factors of the article. Purushotham *et al.* [14] pointed out that CTR has poor performance in the case of few feedbacks contained in the dataset, the so-called sparsity of data. To solve the problem, they integrated follower relationship among users as assistant information. The social network structure was transformed into social matrix, and approximated by an additional MF model. On the other hand, Wang *et al.* [15] proposed Collaborative Deep Learning (CDL), a latent factor model based on CTR to improve its performance. Instead of LDA, they use stacked denoising autoencoder to infer the distribution of latent topics for scientific article. Different with other CTR-based method, CDL directly use the probabilistic distribution of topics to be the latent factors of the given article. The idea of these CTR-based method is to join the distribution of latent topics into users' or items' latent factors, or to replace them. Since the topics are not derived from the individual review, they cannot perceive the unequally treatment of factors in the user's evaluation to a specific item. In contrast, we learn the topics for each review by LDA independently, and use it to be the direction in updating of latent factors in their learning.

Another consideration to combine latent factor model and topic model is to define a transformation between the topic distribution of reviews and latent factor vector of the corresponding item. McAuley *et al.* [6] proposed a latent factor based model named Hidden Factors of Topics (HFT) which integrates LDA and MF. The two models are combined with a softmax function of item's latent factors, which transforms them into topic distribution of correlated reviews. Based on HFT, Bao *et al.* [7] proposed Topic-MF, in which they replace LDA with Non-negative Matrix Factorization model. Not only the items' latent factors, the users' factors are also introduced into the transformation of softmax function. Therefore, the topic distribution represents no longer the topics in the reviews that correlate with a specific item, but a single review in a given feedback. Although it is demonstrated that the performance of HFT and TopicMF outperforms the traditional models such as MF, both of them suffer the drawback of the complicate transformation of latent factor and topic distribution.

III. Preliminaries

A. Problem Definition

The problem we study is to accurately predict the ratings of unpurchased items for users based on their history of feedbacks. Each of the feedbacks includes a numerical rating in scale of $[a, b]$ (e.g. ratings of one to five stars on Amazon) and a textual correlated review. Suppose in the feedbacks we have I users and J items overall. The rating made by user u_i ($i \in \{1, \ldots, I\}$) to item v_j ($j \in \{1, \ldots, J\}$) is denoted as $r_{i,j}$. If $r_{i,j}$ exists, it must have a correlated review $d_{i,j}$ written by u_i. Therefore, feedback is a 4-tuple $< u_i, v_j, r_{i,j}, d_{i,j} >$. Note that for a given user u_i and his unpurchased item $v_{j'}$ ($j' \in \{1, \ldots, J\}$), we only predict the unknown rating as $\hat{r}_{i,j'}$ without the unknown review $d_{i,j'}$.

B. Matrix Factorization for Recommendation

Biased matrix factorization [2] is an influential method to predict the missing ratings in recommendation. It maps users and items into a joint latent factor space with K dimensions which is arbitrary predefined. Accordingly, each user u_i is associated with a vector $U_i \in \mathbb{R}^K$, whose elements measure his/her extent of interest to such factors. On the other hand, vector $V_j \in \mathbb{R}^K$ is associated with a given item v_j, and presents the positive or negative extent of those factors that v_j possesses. The inner product of U_i and V_j represents the interaction of u_i and v_j, and approximates the corresponding rating $r_{i,j}$:

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

where μ is the average of ratings over all users and items, b_i and b_j denote the observed biases of user u_i and item v_j, respectively. Normally, a bias of a given user or item is calculated as the result of subtraction of μ from the average of correlated ratings.

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

$$\begin{aligned}\mathcal{L} = \frac{1}{2} \sum_{i,j} [c_{i,j}(r_{i,j} - \hat{r}_{i,j})^2 \\ + \lambda(\|U_i\|^2 + \|V_j\|^2 + {b_i}^2 + {b_j}^2)],\end{aligned} \tag{2}$$

where λ is the parameter to control the regularization to avoid over-fitting in learning, and $\| \cdot \|^2$ denotes the L^2 norm. $c_{i,j}$ is the confidence parameter of rating $r_{i,j}$, which indicates how much we trust it. A large $c_{i,j}$ should be assigned for some deliberate ratings, and a small $c_{i,j}$ for the ones that do not deserve seriously treatment such as advertisings and fakes.

A typical way to minimize the objective function (2) is to use gradient descent algorithm [2], [9]. It calculates the gradients of U_i and V_j for every given rating $r_{i,j}$ as

$$\begin{aligned} g_{U_i} &= -c_{i,j}(r_{i,j} - \hat{r}_{i,j})V_j + \lambda \cdot U_i \\ g_{V_j} &= -c_{i,j}(r_{i,j} - \hat{r}_{i,j})U_i + \lambda \cdot V_j, \end{aligned} \tag{3}$$

and updates them to the inverse direction of gradients iteratively. The updating step is often unique and controlled by a constant learning rate. Since a big learning rate causes divergence of the objective function and a small one may result in slow learning, it is crucial to find a proper learning rate [16].

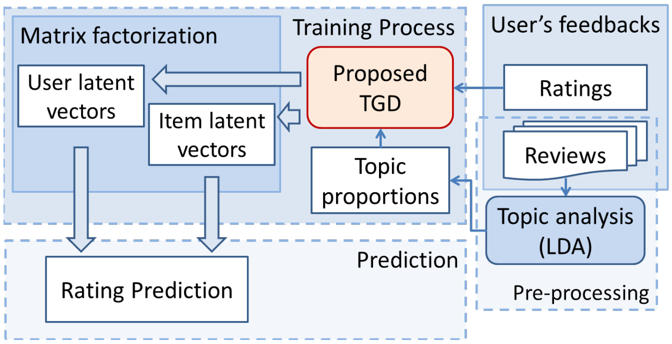

Fig. 2. The construction of proposed method.

IV. Proposed Method

In this section, we present our proposed method, whose structure is shown in Fig. 2. With a given set of history of feedbacks, the first task for us is to derive the topics from each review. As the pre-processing, we use LDA [8], which is a probabilistic generative latent topic model of a set of semantic documents called corpus. Its idea is that each latent topic is characterized by a distribution over words, and a document is a random mixture over such topics. We take each review in the feedback as a single document, and all reviews as the corpus D. Assume that there are K topics overall in D, which are shared by all documents. A topic is denoted by t_k with $k \in \{1, \ldots, K\}$. For a review $d_{i,j} \in D$, its topic distribution is denoted by $\theta_{i,j}$, which is a K-dimensional stochastic vector. Therefore, each of the elements $\theta_{i,j}^k$ represents the proportion of corresponding topic t_k having been mentioned in $d_{i,j}$. Following the method presented in [17], we independently train the LDA model for D and infer $\theta_{i,j}$ for each review $d_{i,j}$ by Gibbs Sampling.

Next, we use MF to model the ratings and further to predict the missing ones for users. The difficulty comes from the link of the topic distributions of reviews and latent factors without a complicated transformation between them. We propose a new first-order gradient descent method named Topic Gradient Descent (TGD), to correlate them through the training process of MF. Since the reviews provide an efficient tool for the users to explain their ratings, important topics are often mentioned much in the reviews. Therefore, the topic distribution $\theta_{i,j}$ represents the importance of degree of topics in the evaluation of user u_i to item v_j, rather than his/her preference on v_j. In other words, when $\theta_{i,j}^k = 0$, t_k is not worth to mention for u_i and have no impact on the evaluation of v_j. Assume that the number of latent factors is equal to the number of topics, and topic t_k corresponds and interprets the elements U_i^k and V_j^k of U_i and V_j. The key idea is to use $\theta_{i,j}^k$ to affect the learning of U_i^k and V_j^k in the training process of MF. To be more specific, a given error of the rating prediction $r_{i,j} - \hat{r}_{i,j}$ is a linear combination of $\theta_{i,j}$, U_i and V_j. With the denotation of gradients g_{U_i} and g_{V_j} in (3), we write the updating equation for U_i and V_j as

$$\begin{aligned} U_i &\leftarrow U_i - \gamma H_{i,j} \cdot g_{U_i} \\ V_j &\leftarrow V_j - \gamma H_{i,j} \cdot g_{V_j}, \end{aligned} \tag{4}$$

where γ is a pre-defined constant, and $H_{i,j}$ is a $K \times K$ diagonal matrix with $\theta_{i,j}$ as the diagonal elements. $H_{i,j}$ is together with γ to be the learning rate, which assigns various updating steps for each latent factor. For the topics which have high importance and generate much error, their corresponding latent factors are updated with large steps. In contrast, factors of unimportant topics are updated with small steps in every epoch of training. When U_i and V_j are intialized with vectors of extremely small constant, such factors will remain the initial values and further have little impact on the rating prediction.

Although we have correlated latent factors with topics and realized their unequal treatment, an issue remained to be solved is that the convergence of the objective function (2) may be slow. Since the average of $\theta^t_{i,j}$ is $1/K$, the average of updating step reduces to $1/K$ of the traditional gradient descent method [1]. Let $s \in [1, +\infty]$ be the timestamp that represents the epochs in training. Following the idea of previous effort [18], we introduce the timestamp into the learning rate. Instead of a constant, γ is re-defined with a function of the timestamp s:

$$\gamma = \frac{\alpha}{\sqrt{s}} \tag{5}$$

where α is an arbitrary predefined constant. γ is inverse to s so that it reduces following the growth of s. Therefore, U_i and V_j are updated with large steps at the beginning of training, and slightly adjusted to find the most proper values at last.

We present TGD method in Algorithm 1, where U^s_i and V^s_i denote the values of U_i and V_j in epoch s. Note that although the form of updating is similar to second-order Newton's method, we only use first-order information of U_i and V_j. Let $|D|$ denotes the number of reviews in corpus D. In each epoch, the time complexities for the calculation of gradients and update of U_i and V_j are $\mathcal{O}(|D| \cdot K)$. Also assume that in the epoch T the objective function converges. Therefore, the time complexity of TGD remains $\mathcal{O}(T \cdot |D| \cdot K)$, the same as existing first-order method.

With the MF model trained by TGD, for a given user u_i and an unpurchased item $v_{j'}$, we calculate the rating prediction $\hat{r}_{i,j'}$ following (1).

V. Evaluation

In this section, we conduct the evaluation with three perspectives: 1) whether the proposed TGD method makes the objective function (2) rapidly converge; 2) how the parameters impact the performance of the proposed method; 3) how is the performance of proposed method comparing with MF and the state-of-the-art model for recommendation.

A. Datasets and Implementation

In evaluation, we use several datasets have been driven from YELP [2] and Amazon [19]. They are filtered by the following constraints to have the feedbacks that: 1) their reviews have at least 10 words; 2) each of the users has at least 5 feedbacks; 3) each of the items concerns with at least 5 feedbacks.

Algorithm 1 Topic Gradient Descent

```
Require: $\theta_{i,j}$ for $d_{i,j} \in D$
Initialize $U_i$ and $V_j$ with vectors of unique value and $\alpha$ with constant, set $s = 1$.
while The objective function (2) has not converged do
    Calculate $\gamma \Leftarrow \alpha \cdot s^{-1/2}$
    for $d_{i,j} \in D$ do
        Compute gradients $g_{U_i}$ and $g_{V_j}$
        Apply update $U^{s+1}_i \Leftarrow U^s_i + \gamma H_{i,j} \cdot g_{U_i}$
        Apply update $V^{s+1}_j \Leftarrow V^s_j + \gamma H_{i,j} \cdot g_{V_j}$
    end for
    $t \Leftarrow t + 1$
end while
```

Additionally, since in the following comparison with existing method [2], [7] large datasets make the experiments time consuming, we cut each of them by the publishing date of the feedbacks. For *YELP challenge dataset*, we only utilize the feedbacks from State of Arizona and Nevada for the sparsity of data. Discard of the stop words and stemming are also conducted for each review. With these processes, Table I shows their statistic including the number of users, items and feedbacks contained. The third and seventh columns show the average of rating and number of words in a review in the datasets, respectively. The sparsity of a dataset is calculated as #feedbacks /(# users × # items).

For each dataset, we randomly take 80% of its feedbacks as training set, and the rest as testing set to conduct the experiments.

B. Convergence of Topic Gradient Descent

For each of the training sets, we train the proposed method to observe the sum of squared error of rating prediction in each epoch. Considering the total number of the reviews in datasets, parameters K and λ are fixed to 20 and 0.01 respectively. The latent factors in U_i and V_j are initialized to be unique values of 0.001. As comparison, we also train MF by the method presented in [2], with its K and λ fixed with the same values as our proposed method. Different with the proposed method, factors in U_i and V_j are initialized by randomly generated values following the zero mean Gaussian distribution of $\mathcal{N}(0, \lambda^2)$. In order to guarantee the fairness, we set the confidence parameter $c_{i,j}$ to 1 if $r_{i,j}$ exists, and 0 otherwise for both the proposed method and MF.

For the their typical results, we show the results in the first 500 epochs for dataset *Video Games*, and 150 epochs for *Movies and Videos* in Fig. 3. The parameter α in (5) is fixed from 1.0 to 1.3, and the learning rate of MF is set to 0.03. For both of the datasets, MF reaches lower levels of squared error than the proposed method. For *Video Games*, the proposed method reduces the squared error more slowly than MF, which is opposite to the result of *Movies and Videos*. Especially in *Movies and Videos*, $\alpha = 1.3$ is not a proper assignment since the squared error divergences early. Considering that for a given feedback, the updating steps of latent factors depend

[1] B. Webb, "Netflix update: Try this at home", http://sifter.org/∼simon/journal/20061211.html

[2] http://www.yelp.com/dataset_challenge

TABLE I. THE DESCRIPTION OF DATASETS USED IN EXPERIMENTS

dataset	# feedbacks	avg.rating	# users	# items	sparsity	avg.words	since
YELP	1933	4.059	341	286	0.019	66	Jul. 2014
Movies and Videos	1400	3.954	172	145	0.056	76	Sep. 2012
Tools and Home Improvement	1152	4.375	147	125	0.063	46	Sep. 2012
Baby	1096	4.187	130	112	0.075	39	Feb. 2013
Toys and Games	385	3.977	36	36	0.297	34	Oct. 2013
Cell Phones and Accessories	1084	4.565	103	94	0.112	45	Feb. 2014
Beauty	913	4.772	103	44	0.201	36	May. 2014
Video Games	949	4.080	137	78	0.089	71	May. 2013
Sports and Outdoors	1304	4.364	178	138	0.053	38	Jun. 2012
Grocery and Gourmet Food	647	4.136	87	66	0.113	36	Oct. 2013
Digital Music	1078	4.079	113	116	0.082	67	Oct. 2008

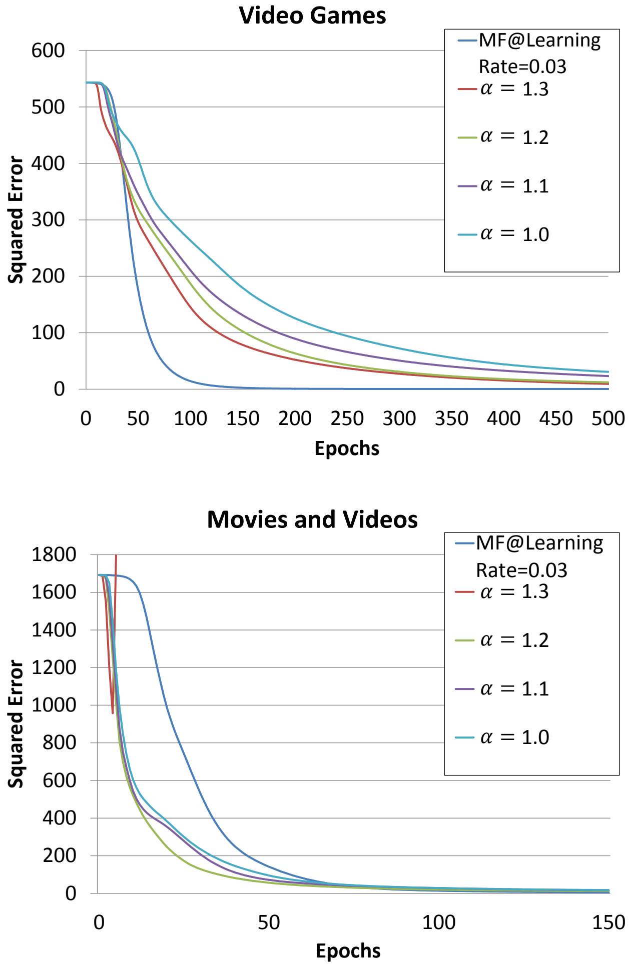

Fig. 3. The squared error of rating prediction in the training of MF with using proposed TGD and existing method. α is fixed to $1.0, 1.1, 1.2$ and 1.3. For existing method, the learning rate is set to a general value of 0.03.

on the topic distribution of its review, we calculated and observed the standard deviation (SD) of topic distribution for each review. As a consequence, the average SD of *Movies and Videos* (0.067) is much higher than *Video Games* (0.029). It indicates that the speed of convergence depends on the dispersion of the topics' proportions in the reviews.

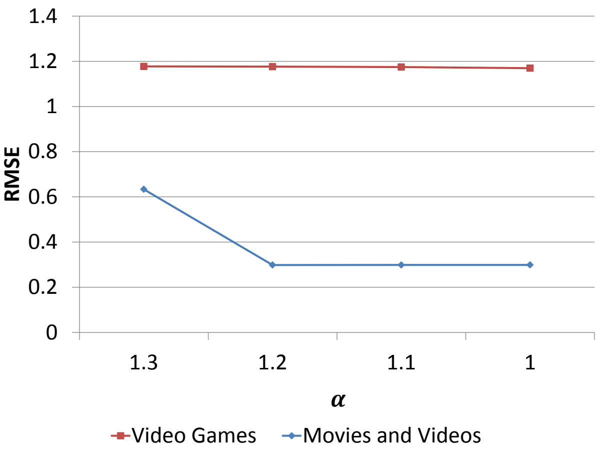

Fig. 4. The performance of *Video Games* and *Movies and Videos* in term of RMSE with α is fixed to 1.3, 1.2, 1.1 and 1.0.

C. Impact of Parameters

Since the problem is to predict the ratings of the users to their unpurchased items, the performance of the proposed method is evaluated by observing the accuracy of predictions. For a given feedback from the testing set, we compare the rating prediction $\hat{r}_{i,j}$ with its actual rating $r_{i,j}$. 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 the testing set, and $|\cdot|$ denotes the absolute value. In general, RMSE is more sensitive than MAE for large error of prediction. The assignment of parameters and initialization follows the previous experiment in Section V-B. For each of the training sets, the proposed method is trained until the objective function converges.

Fig. 4 shows the performance of the proposed method with α changed from 1.0 to 1.3. For *Video Games* RMSE is stable for all cases of α. On the other hand, RMSE of *Movies and Videos* is over 0.6 when $\alpha = 1.3$, and reduces to roughly 0.3 when $\alpha \in \{1.2, 1.1, 1.0\}$. Combining the results of the previous experiment, it is indicated that the divergence of

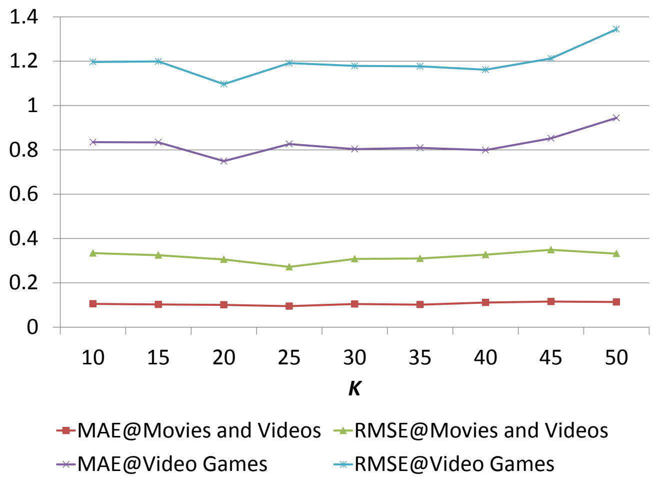

Fig. 5. The performance of *Video Games* and *Movies and Videos* with K fixed from 10 to 50.

objective function in learning further affects the performance. In other words, the performance of the proposed method is stable to small enough α. In order to avoid such affection, we fix α to 1.2 to conduct following experiments.

Fig. 5 shows the performance with K changed from 10 to 50. Recall that K denotes the number of overall topics, also the dimension of U_i and V_j. For *Video Games*, MAE and RMSE vary in parallel following K. When $K = 20$ the proposed method has the best performance and when $K \geq 40$ the performance declines. In the case of *Movies and Videos*, although MAE is stable, a trough of RMSE is observed when $K = 25$. Therefore, in order to achieve the best performance, K should be fixed into a proper range which depends on the dataset. An assignment of too small or too large values makes the performance declines.

D. Performance in Recommendation

According to the previous experimental results, we set K to 20 and 40 to conduct a detailed evaluation to the performance in rating prediction. Except MF, we also implemented TopicMF [7] which is an extension of HFT [6] as comparison. Following the setup of their experiments, we set $\lambda = 1$, $c_{i,j} = 1$ if $\exists r_{i,j}$ and $\lambda_u = \lambda_v = \lambda_B = 0.001$. Since the training of TopicMF is time consuming (3 to 5 minutes for a training set with a scale of 1,000 reviews for one epoch), we train it with 100 epochs and report its performance.

Table II summarizes the results of all datasets, with the best performance emphasized in boldface for each of the datasets. The last lines of the two tables present the average of MAE and RMSE. The improvement of the proposed method is calculated and presented in last four columns both for each dataset and average performance. When $K = 20$, the proposed method shows the best performance in term of RMSE on 10 datasets. Comparing with MF, the improvement of the proposed method is 3.77% in MAE, and 5.82% in RMSE in average. It indicates that the proposed method is effective in reducing the decisive failure of prediction. Especially on *YELP*, *Movies and Videos*, *Video Games* and *Digital Music*, the proposed method gains the improvement from 6.40% up to 9.03% in MAE. Also referring Table I, averages of words in one review of these four datasets are all more than 65. It represents that their reviews are written in more detail than other datasets. Therefore, the topics could be more clearly inferred to make the latent factors well trained in learning. On the other hand, the proposed method also outperforms TopicMF on 11 datasets, with the improvement of 4.87% in MAE and 3.99% in RMSE in average. When $K = 40$, the performance of the proposed method declines on most of the datasets except *Digital Music* and *Sports and Outdoors*. It represents that for such datasets, setting K to 40 makes the topics not clearly derived, further affects the performance. In term of RMSE, the improvement also reduces to -0.52% comparing with MF, and -3.40% with TopicMF in average.

Additionally, we underline the best performance among the approaches in both cases of K for each dataset. For example, the proposed method obtains the smallest RMSE for YELP dataset, which is underlined in the first table of $K = 20$. Overall, the proposed method obtains the best performance on 8 datasets in term of RMSE, 7 datasets in MAE. It is also observed that only two of them is in the case of $K = 40$. Therefore, proper assignment of K (20 for most of the datasets) guarantees the proposed method to gain better performance than two existing methods.

In practical application, if the predicted rating of an unpurchased item is high, such item may be a future recommendation to the given user. Therefore, we particularly evaluate the accuracy of predictions to the actual ratings with the highest score. Considering that both in YELP and Amazon a user evaluates an item up to 5 stars, we take the feedbacks in the testing set with 5 stars' ratings as objective ones. For such feedbacks, the prediction is successful if the predicted rating locates in $[4.5, \infty)$. The precision is calculated as the proportion of successful predictions to the objective feedbacks.

Table III shows the precision of the proposed method and MF on each dataset with $K = 20$. For example, to 5 stars' ratings in YELP dataset, 55.5% of them are predicted in the range of $[4.5, \infty)$ by MF, and 59.4% by the proposed method, respectively. Among all datasets the improvement of the proposed method is up to 6.977%. For *Movies and Videos*, since the precision of both MF and the proposed method is at a high level of more than 0.98, correspondingly the improvement is the slightest (0.334%). Also note that although the performance of the proposed method is worse than MF in *Sports and Outdoors* and *Grocery and Gourmet Food* (line 9 and line 10 in Table II), the precision is higher than MF. It demonstrates that the proposed method has higher accuracy in the prediction of such highest ratings than MF.

VI. Conclusion

In this paper, we propose a new method to predict ratings for recommendation, including a topic gradient descent method (TGD) for the MF model. From the given textual reviews, their topics are derived by Latent Dirichlet Allocation model. Using such topics, in the learning of the proposed method the latent factors of the users and items are iteratively updated by dynamically assigned updating steps. In the evaluation, we conduct a series of experiments utilizing 11 datasets, including

TABLE II. THE PERFORMANCE IN TERMS OF MAE AND RMSE OF MF, TOPICMF AND THE PROPOSED METHOD ON ALL DATASETS

$K = 20$										
	MAE			RMSE			Improvement in MAE		Improvement in RMSE	
Dataset	MF	TopicMF	Proposed Method	MF	TopicMF	Proposed Method	vs MF	vs TopicMF	vs MF	vs TopicMF
YELP	0.772	0.771	**0.723**	1.095	1.001	**0.969**	6.40%	6.25%	11.50%	3.19%
Movies and Videos	0.097	0.182	**0.088**	0.321	0.421	**0.306**	8.86%	51.45%	4.60%	27.18%
Tools and Home Improvement	0.648	0.648	**0.622**	0.934	0.925	**0.891**	3.96%	3.90%	4.66%	3.72%
Baby	0.693	0.703	**0.660**	0.876	0.892	**0.850**	4.72%	6.11%	2.95%	4.65%
Toys and Games	0.587	0.586	**0.568**	0.839	0.814	**0.758**	3.18%	3.00%	9.70%	6.93%
Cell Phones and Accessories	0.472	0.453	**0.444**	0.685	0.674	**0.638**	5.98%	2.03%	6.82%	5.32%
Beauty	0.238	0.225	**0.221**	0.401	0.385	**0.380**	7.48%	1.85%	5.28%	1.19%
Video Games	0.817	0.791	**0.760**	1.173	1.115	**1.061**	7.01%	4.01%	9.48%	4.77%
Sports and Outdoors	**0.607**	0.633	0.643	**0.844**	0.867	0.902	-5.87%	-1.50%	-6.86%	-3.99%
Grocery and Gourmet Food	**0.708**	0.738	0.739	**0.959**	0.986	0.970	-4.33%	-0.15%	-1.12%	1.64%
Digital Music	0.765	0.749	**0.696**	1.107	0.980	**0.971**	9.03%	7.14%	12.23%	0.83%
Average	0.582	0.589	**0.560**	0.839	0.824	**0.791**	3.77%	4.87%	5.82%	3.99%

$K = 40$										
	MAE			RMSE			Improvement of MAE		Improvement of RMSE	
Dataset	MF	TopicMF	Proposed Method	MF	TopicMF	Proposed Method	vs MF	vs TopicMF	vs MF	vs TopicMF
YELP	0.770	0.761	**0.722**	1.091	0.997	**0.970**	6.28%	5.13%	11.12%	2.77%
Movies and Videos	**0.101**	0.196	0.109	0.335	0.436	**0.323**	-7.36%	44.45%	3.52%	25.98%
Tools and Home Improvement	0.630	**0.622**	0.635	0.911	**0.891**	0.923	-0.83%	-2.06%	-1.35%	-3.68%
Baby	**0.667**	0.671	0.724	**0.851**	0.855	0.934	-8.55%	-7.80%	-9.68%	-9.19%
Toys and Games	**0.587**	0.613	0.611	0.839	**0.832**	0.766	-4.17%	0.29%	8.77%	7.95%
Cell Phones and Accessories	0.463	**0.471**	0.498	**0.672**	0.688	0.723	-7.70%	-5.71%	-7.51%	-5.12%
Beauty	0.226	**0.220**	0.250	0.388	**0.380**	0.444	-10.56%	-13.27%	-14.29%	-16.73%
Video Games	0.813	**0.760**	0.859	1.154	**1.061**	1.239	-5.72%	-13.12%	-7.31%	-16.71%
Sports and Outdoors	**0.605**	0.617	0.628	**0.830**	0.840	0.862	-3.81%	-1.82%	-3.87%	-2.66%
Grocery and Gourmet Food	0.681	**0.701**	0.760	**0.928**	0.946	1.019	-11.67%	-8.41%	-9.83%	-7.72%
Digital Music	0.771	0.713	**0.682**	1.113	**0.933**	0.958	11.58%	4.37%	13.90%	-2.74%
Average	**0.574**	0.577	0.589	0.828	**0.805**	0.833	-2.61%	-2.08%	-0.52%	-3.40%

TABLE III. THE PRECISION OF THE PROPOSED METHOD AND MF IN PREDICTION OF 5 STAR'S RATING

Dataset	MF	Proposed Method	Improvement
YELP	0.555	0.594	6.977%
Movies and Videos	0.984	0.987	0.334%
Tools and Home Improvement	0.6	0.622	3.704%
Baby	0.439	0.449	2.326%
Toys and Games	0.773	0.818	5.882%
Cell Phones and Accessories	0.734	0.741	0.98%
Beauty	0.845	0.865	2.29%
Video Games	0.545	0.571	4.918%
Sports and Outdoors	0.689	0.722	4.808%
Grocery and Gourmet Food	0.579	0.592	2.273%
Digital Music	0.559	0.581	3.846%

YELP challenge dataset and per-category Amazon reviews. Firstly, the experimental results verified that the TGD certainly converges the squared error of the rating prediction. Secondly, it also shows that the proposed method outperforms MF in the recommendation. The accuracy of rating prediction improves up to 12.23% in term of RMSE, and 5.82% on average in all datasets. Comparing with TopicMF which is a state-of-the-art recommendation model for recommendation, it also achieves a superiority of performance. Finally, the proposed method is demonstrated to have higher accuracy than MF in the prediction of high-scored ratings, which is considered as an ordinary scene of recommendation.

In the future, we intent to develop a mechanism to automatically search the proper assignment of parameters corresponding with the given dataset. On the other hand, we hope to evaluate the ability to describe the predicted ratings by the learnt latent factors and derived topics. Not only for MF, we also plan to apply the proposed TGD method to tensor factorization to extend it as an optimization to general latent factor based model.

REFERENCES

[1] D. Goldberg, D. Nichols, B. M. Oki, and 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. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, 2009.

[3] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using markov chain monte carlo," in *Proceedings of the 25th international conference on Machine learning*. ACM, July 2008, pp. 880–887.

[4] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, "Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering," in *Proceedings of the Fourth ACM Conference on Recommender Systems*. ACM, September 2010, pp. 79–86.

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

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

[7] Y. Bao, H. Fang, and J. Zhang, "Topicmf: Simultaneously exploiting ratings and reviews for recommendation." in *AAAI*, vol. 14, 2014, pp. 2–8.

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

[9] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proceedings of the 20th International Conference on Neural Information Processing Systems*, December 2007, pp. 1257–1264.

[10] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. Ieee, December 2008, pp. 263–272.

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

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

[13] D. Agarwal and B.-C. Chen, "flda: matrix factorization through latent dirichlet allocation," in *Proceedings of the third ACM international conference on Web search and data mining*. ACM, February 2010, pp. 91–100.

[14] S. Purushotham, Y. Liu, and C.-c. J. Kuo, "Collaborative topic regression with social matrix factorization for recommendation systems," in *Proceedings of the 29th International Conference on Machine Learning*, June 2012, pp. 759–766.

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

[16] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

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

[18] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ser. ICML'03, August 2003, pp. 928–935.

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