International Journal of Networking and Computing – www.ijnc.org, ISSN 2185-2847
Volume 11, Number 2, pages 198–214, July 2021

Matrix Factorization with Topic and Sentiment Analysis for Rating Prediction

Ben Wang
Graduate School of Advanced Science and Engineering, Hiroshima University,
1-7-1 Kagamiyama, Higashi Hiroshima, Hiroshima, 739-8521 JAPAN

Guan-Shen Fang
National Institute of Technology, Tsuyama College,
624-1, Numa, Tsuyama-City, Okayama, 708-8509 JAPAN

and

Sayaka Kamei
Graduate School of Advanced Science and Engineering, Hiroshima University,
1-7-1 Kagamiyama, Higashi Hiroshima, Hiroshima, 739-8521 JAPAN

Received: February 15, 2021
Revised: May 4, 2021
Accepted: June 6, 2021
Communicated by Yasuaki Ito

Abstract

In today's online services, users' feedback such as numerical rating, textual review, time of purchase, and so on for each item is often encouraged to provide. Managers of online services utilize the feedback to improve the quality of their services, or user experience. For example, many recommender systems predict the items that the users may like and purchase in the future using users' historical ratings. With the increase of user data in the systems, more detailed and interpretable information about item features and user sentiments can be extracted from textual reviews that are relative to ratings. In this paper, we propose a novel topic and sentiment matrix factorization model, which leverages both topic and sentiment drawn from the reviews simultaneously. First, we conduct topic analysis and sentiment analysis of reviews using Latent Dirichlet Allocation (LDA) and lexicon construction technique, respectively. Second, we combine the user consistency, which is calculated from his/her reviews and ratings, and helpful votes from other users of reviews to obtain a reliability measure to weight the ratings. Third, we integrate these three parts into the matrix factorization framework for the prediction of ratings. Our experimental comparison using Amazon datasets indicates that the proposed method significantly improves performance compared to traditional matrix factorization up to 14.12%.

Keywords: Rating prediction, Matrix Factorization, Topic model, Sentiment analysis, Recommender system

1 Introduction

Recommender systems play a significant role in today's online services and business systems. Their main goal is to help users discover items that they are interested in purchasing from large-scale of items. The history feedback provided by users after their purchase is the basis of the recommender systems, mainly including digital ratings and textual reviews. As the most effective algorithm for predicting rating, Collaborative Filtering (CF) [5] assumes that users who are interested in the same items share similar interests. Matrix Factorization (MF) [10, 13] is the ideal approach among CF algorithms, which is based on the latent factor model. It characterizes both users and items by vectors of latent factors that are inferred from user ratings. For a user and an unpurchased item, it calculates the inner product of their latent vectors as the predicted rating.

However, recent research [4, 2] pointed out the mediocre performance of MF caused by its ignorance of the textual reviews, which have users' detailed opinions about items. In order to solve this problem, efforts are made to recognize and characterize such opinions into sentiments and topics, to enhance the performance of MF. Existing studies include the applications of the topic model [2, 3, 19], sentiment analysis [21, 17], and their combination [14, 22, 20].

In this paper, through effective utilization of users' feedback, we propose a new approach to predict the missing ratings of given items and users for the recommender systems. Our idea is to replace the latent feature matrices of MF with two new fixed matrices, and assign weights for them to predict rating based on reliability measures. Firstly, we train Latent Dirichlet Allocation (LDA) [6] model with reviews of users' historical feedback. For each item, we infer topic probability distribution for each of its relevant reviews and summarize them as its topic distribution vector. By gathering all items' topic distribution vectors, we fix *item topic distribution matrix.* On the other hand, we apply sentiment analysis to each review to derive sentiment intensity via Valence Aware Dictionary and Sentiment Reasoner (VADER). For each user, we combine the topic distribution vectors and the sentiment intensity of his/her reviews to construct a preference vector. Similarly to items, all users' preference vectors are gathered and constitute the fixed *user preference distribution matrix*. Secondly, we introduce reliability measures both for users and items, which indicate the trustworthiness of their reviews and ratings. They are calculated by the sentiment intensity of relevant feedback and the helpfulness indicator, namely the helpful votes given by other users. User reliability measure is used as weights of item topic distribution matrix and user preference distribution matrix both in the training phase and prediction phase. Item reliability measure is used as parameters to adjust the learning rate in Stochastic Gradient Descent (SGD) process.

In the evaluation, we perform the experiments with Amazon review dataset, to compare the overall performance of missing rating prediction under various values of parameters. Particularly, the main contributions of this paper are as follows:

- We simultaneously introduce the topic model, the sentiment analysis, and the reliability measure into the traditional MF method for better performance.
- Comparing with the other five existing methods for rating prediction, the proposed models SCMF and SCMFP outperform all other methods in most of the datasets, and SCMFP (resp. SCMF) derives an improvement up to 14.11% (resp. 14.12%) in terms of RMSE compared with traditional MF.

The remainder of this paper is organized as follows: Section II overviews related works of latent factor models and the review extraction. Section III simply describes the fundamental of the basic latent factor models. Section IV describes the existing methods, i.e., SBMF+R and STMF, because we utilize the part of the main idea of these methods. Section V describes the detail of our approaches. Section VI represents the experimental methodologies of the proposed methods and the results. Finally, section VII gives conclusions and outlines future works.

2 Related Work

With the increase in feedback to published items, researchers are increasingly focusing on how to integrate the topic model and sentiment analysis of reviews in feedback into recommendation. First, researchers have tried to use the topic model to directly impact the generation process of the latent factors of MF methods [11, 2, 3, 15, 18]. The methods of [11, 2] transform the topic distribution of reviews by LDA to latent factors of MF, while the method of [3] aligns learning rates of MF by using the topic distribution. The method proposed by Peña et al. [15] uses the topic distribution of reviews for the initialization of the latent factors of MF. The method proposed by Shoja et al. [18] uses the topic distribution by LDA to extract user attributes related to each item category, and construct the user attributes matrix separately from the user-item matrix. In these methods, they do not consider the sentiment intensity of textual reviews.

Another consideration is to take the sentiment intensity derived from the reviews as the virtual rating to augment recommendations. Zhang et al. [21] suggested that combining real ratings with inferred ratings extracted from emoticons and opinion words of reviews is indicated to return better recommendations. Hyun et al. [8] proposed a CNN-based recommendation method that is guided to incorporate the sentiments when modeling the users and items. Shen et al. [17] presented SBMF+R model based on the probability matrix factorization, incorporated the ratings, sentiment intensities, and helpful votes from other users for prediction simultaneously.

Since the item features can be shown by the topic model and the user sentiments can be estimated from sentiment analysis, the combination of the topic model and sentiment analysis becomes popular. Wang et al. [20] considered the sentiment and topics involved in the reviews and proposed a novel interpretable model called STMF, especially in explaining user preference. Zhang et al. [22] proposed a method that combines the topics in reviews via LDA and the emotion of each topic with the item-based collaborative filtering recommendation (Note that, their method is not the model-based method.). Although these approaches mainly rely on the use of topic and sentiment analysis of textual reviews, they lack a measurement of reliability and deep use of the sentiment intensity.

3 Preliminaries

3.1 Problem Definition

The problem that we study is to accurately predict the ratings of unpurchased items based on the users' historical feedback, i.e., our purpose is to predict missing values in the user-item rating matrix. Normally, each feedback includes a rating in the range of [1, 5] and a related textual review. Suppose there are N users and M items. The rating evaluated by user u_i ($i \in \{1, \dots, N\}$) to item v_j ($j \in \{1, \dots, M\}$) is denoted as r^5_{ij} and r^1_{ij}, where r^5_{ij} is the observed rating in the scale of [1, 5] and r^1_{ij} is considered as the converted rating in the scale of [-1, 1] obtained from r^5_{ij} as following:

$$r^1_{ij} = \frac{1}{2}(r^5_{ij} - 3) \tag{1}$$

Therefore, for the given user u_i, the prediction of missing rating $\hat{r}^5_{ij}$ on the given item v_j is the problem that we consider. Let R^5 and R^1 be $N \times M$ user-item rating matrices such that $r^5_{ij} \in R^5$ and $r^1_{ij} \in R^1$ respectively.

Also, we denote the textual review of user u_i on item v_j as d_{ij}, and the sentiment intensity of d_{ij} extracted by VADER [7] method in the third-party toolkit NLTK or any method based on lexicon [17, 1, 9] as s^5_{ij} and s^1_{ij}, where s^1_{ij} is original sentiment intensity in the scale of [-1, 1] and s^5_{ij} is in the scale of [1, 5] converted from s^1_{ij} according to the following formula:

$$s^5_{ij} = 2 \times s^1_{ij} + 3 \tag{2}$$

Let S^5 and S^1 be $N \times M$ user-item sentiment intensity matrices in which $s^5_{ij} \in S^5$ and $s^1_{ij} \in S^1$ respectively.

Additionally, there are other users' helpful votes on the authenticity of each user's historical feedback (r_{ij}, d_{ij}). To be more specific, (r_{ij}, d_{ij}) can be upvoted/downvoted as positive/negative by other users, so the positive votes number for (r_{ij}, d_{ij}) and total votes number for (r_{ij}, d_{ij}) are denoted as f^P_{ij} and f_{ij} respectively.

3.2 Matrix Factorization Model

Matrix Factorization (MF) [10] is an effective method to predict the missing ratings for the recommender systems, which has two common versions—basic MF and biased MF. At first, the biased MF will initialize two predefined matrices—user latent feature matrix U and item latent feature matrix V using K-dimensional latent factor space. The vector $U_i \in \mathbb{R}^{\mathbb{K}}$ of U is assumed to be associated with user u_i while the vector $V_j \in \mathbb{R}^{\mathbb{K}}$ of V is assumed to be associated with item v_j, in which the elements of U_i measure the extent of the interest of u_i to such factors and V_j 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^5_{ij} as follows:

$$r^5_{ij} \sim \hat{r}^5_{ij} = \mu + b_i + b_j + U_i^T V_j$$

where μ is the global bias, i.e., the average of all observed ratings, b_i and b_j are the user bias for u_i and the item bias for v_j, respectively. Therefore, the objective is to learn U_i and V_j through a given training set, by minimizing the sum-of-squared-error as shown in the following:

$$\zeta = \frac{1}{2}\sum_{i,j}[(r^5_{ij} - \hat{r}^5_{ij})^2 + \lambda(\|U_i\|^2 + \|V_j\|^2 + \|b_i\|^2 + \|b_j\|^2)] \tag{3}$$

where λ is the regularization parameter which can avoid overfitting in learning, and $\|\cdot\|$ represents the L^2 norm. A typical way to minimize the objective function (3) is to use the SGD algorithm, which calculates the gradients of U_i and V_j for each observed rating r^5_{ij} as follows:

$$\begin{aligned} gU_i &= -(r^5_{ij} - \hat{r}^5_{ij})V_j + \lambda U_i \\ gV_j &= -(r^5_{ij} - \hat{r}^5_{ij})U_i + \lambda V_j \\ gb_i &= -(r^5_{ij} - \hat{r}^5_{ij}) + \lambda b_i \\ gb_j &= -(r^5_{ij} - \hat{r}^5_{ij}) + \lambda b_j \end{aligned} \tag{4}$$

The basic MF can be obtained by deleting biases μ, b_i, and b_j together.

3.3 Probabilistic Matrix Factorization Model

Probabilistic Matrix Factorization (PMF) [13] is introduced as a further optimized model, which is a probability understanding of the basic MF. The user factors and item factors are modeled by the Gaussian hypothesis as the latent feature matrices U and V, respectively. The conditional distribution over the observed ratings is defined as follows:

$$p(R^5 \mid U, V, \sigma_R^2) = \prod_{i=1}^{N}\prod_{j=1}^{M}[\mathcal{N}(r^5_{ij} \mid U_i^T V_j, \sigma_R^2)]^{I^R_{ij}} \tag{5}$$

where $\mathcal{N}(x \mid \mu, \sigma^2)$ is the probability density function of the Gaussian distribution with mean μ and variance σ^2. σ_R^2 is regarded as the variance of r^5_{ij}, and I^R_{ij} is the indicator function that is equal to 1 if user u_i evaluated item v_j or 0 otherwise. The zero-mean spherical Gaussian priors are also placed on user and item feature vectors:

$$\begin{aligned} p(U \mid \sigma_U^2) &= \prod_{i=1}^{N}[\mathcal{N}(U_i \mid 0, \sigma_U^2\mathbf{I})] \\ p(V \mid \sigma_V^2) &= \prod_{j=1}^{M}[\mathcal{N}(V_j \mid 0, \sigma_V^2\mathbf{I})] \end{aligned} \tag{6}$$

where $\mathbf{I}$ is the identity matrix of size K. Therefore, through simple Bayesian inference, we can know the following inference:

$$\begin{aligned} &p(U, V \mid R^5, \sigma_R^2, \sigma_U^2, \sigma_V^2) \propto p(R^5 \mid U, V, \sigma_R^2)p(U \mid \sigma_U^2)p(V \mid \sigma_V^2) \\ &= \prod_{i=1}^{N}\prod_{j=1}^{M}[\mathcal{N}(r^5_{ij} \mid U_i^T V_j, \sigma_R^2)]^{I^R_{ij}}\prod_{i=1}^{N}[\mathcal{N}(U_i \mid 0, \sigma_U^2\mathbf{I})]\prod_{j=1}^{M}[\mathcal{N}(V_j \mid 0, \sigma_V^2\mathbf{I})] \end{aligned}$$

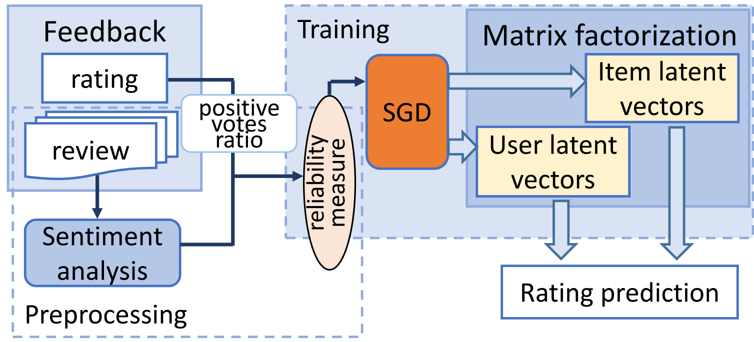


Figure 1: Construction of SBMF+R.

4 Existing Methods

In this section, we introduce two existing methods, SBMF+R and STMF, on which our proposed methods are based. They are novel methods for rating prediction which use sentiment value derived from reviews.

4.1 SBMF+R Model

SBMF+R [17] is an improved model based on PMF, which adds the sentiment intensity extracted from user reviews, and takes the reliability measure into account simultaneously as shown in Figure 1. We can clearly see in the middle of the figure that the reliability measure is extracted from the user's feedback and used in the SGD training.

For this model, with the given historical reviews, the first task is to derive the sentiment intensity of each review by using a method based on the original lexicon. Based on the sentiment intensity, the reliability measure is calculated by the way that we explain in section 4.1.1.

Then, SBMF+R adds the sentiment intensity to latent feature matrices of users and items (See section 4.1.2), and uses the objective function using the reliability measure (See section 4.1.3).

4.1.1 Calculation of the reliability measure

The helpful votes from other users are considered as the helpfulness of the feedback, which reflects the validity of the feedback. Thus, with the user consistency and positive votes ratio by other users on feedback, the reliability measure of each rating can be made for assigning its weight. For each user u_i, M_i is denoted as the number of feedbacks published by u_i. Thus, the sentiment intensity of u_i is s^1_{ij} $(j \in \{1, \ldots, M_i\})$ inferred from d_{ij} via a method based on lexicon [1, 9]. In order to align s^1_{ij} with r^5_{ij}, the formula in Eq.(2) is used to get s^5_{ij}. Therefore, the user consistency c_i of u_i is calculated by the Euclidean distance between the corresponding rating r^5_{ij} and sentiment intensity s^5_{ij}:

$$c_i = \sqrt{\sum_{j}^{M_i} (r^5_{ij} - s^5_{ij})^2}.$$

Then the reliability wu_{ij} of rating r^5_{ij} is defined as follows:

$$wu_{ij} = \frac{f^P_{ij}/f_{ij}}{1 - c_i} \tag{7}$$

where f^P_{ij}/f_{ij} represents the positive votes rate of (r_{ij}, d_{ij}). Then, the denominator of wu_{ij} will not be 0 since the sentiment intensities of users are all decimals. Similarly, the reliability of sentiment intensity s^5_{ij} is $1 - wu_{ij}$. Finally, the interval of reliability factors is normalized into $[0, 1]$.

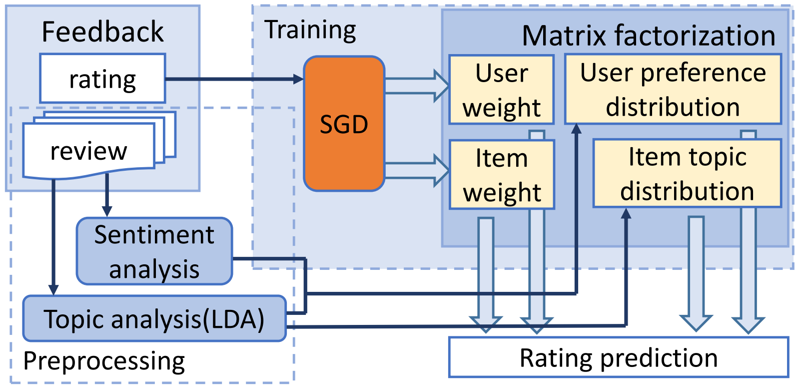


Figure 2: Construction of STMF.

4.1.2 Building conditional distribution

In PMF, the user factor and item factor are modeled by the Gaussian hypothesis as latent feature matrices U and V, respectively. The difference between SBMF+R and PMF is that the conditional distribution over the sentiment intensity is also defined for fitting the sentiment intensity similar to Eq.(5) as follows:

$$p(S^5 \mid U, V, \sigma_S^2) = \prod_{i=1}^{N}\prod_{j=1}^{M}[\mathcal{N}(s_{ij}^5 \mid U_i^T V_j, \sigma_S^2)]^{I_{ij}^S}$$

where σ_S^2 is regarded as the variance of s_{ij}^5, I_{ij}^S is the indicator function that is equal to 1 if user u_i evaluated item v_j or 0 otherwise. Also, the zero-mean spherical Gaussian priors are placed on the user and item feature vectors as in inference Eq.(6), so the inference can be derived as follows:

$$\begin{aligned} &p(U, V \mid S^5, \sigma_S^2, \sigma_U^2, \sigma_V^2) \propto p(S^5 \mid U, V, \sigma_S^2)p(U \mid \sigma_U^2)p(V \mid \sigma_V^2) \\ &= \prod_{i=1}^{N}\prod_{j=1}^{M}[\mathcal{N}(s_{ij}^5 \mid U_i^T V_j, \sigma_S^2)]^{I_{ij}^S} \prod_{i=1}^{N}[\mathcal{N}(U_i \mid 0, \sigma_U^2\mathbf{I})] \prod_{j=1}^{M}[\mathcal{N}(V_j \mid 0, \sigma_V^2\mathbf{I})] \end{aligned}$$

4.1.3 Objective function of SBMF+R

The log of the posterior distribution over the user and item features matrices is given by $\ln p\left(U, V \mid R^5, S^5, \sigma_R^2, \sigma_S^2, \sigma_U^2, \sigma_V^2\right)$, if hyper-parameters $(\sigma_R^2, \sigma_S^2, \sigma_U^2, \sigma_V^2)$ kept fixed, then maximizing the log-posterior is equivalent to minimizing the sum-of-squared-error as shown in the following:

$$\zeta = \frac{1}{2}\sum_{i,j}\{I_{ij}[wu_{ij}(r_{ij}^5 - \hat{r}_{ij}^5)^2] + I_{ij}[(1 - wu_{ij})(s_{ij}^5 - \hat{r}_{ij}^5)^2] + \lambda_U\|U_i\|^2 + \lambda_V\|V_j\|^2\}$$

where I_{ij} is the indicator function that is equal to 1 if user u_i evaluated item v_j or 0 otherwise, $\lambda_U = \sigma_R^2/\sigma_U^2$ and $\lambda_V = \sigma_R^2/\sigma_V^2$ are the regularization parameters.

4.2 STMF Model

As shown in Figure 2, STMF model [20] initializes two predefined matrices using K-dimensional space similarly to the latent factor models (e.g., MF, PMF, SBMF+R). However, the difference is that the item topic distribution Y is a fixed matrix to replace the item latent feature matrix V, and the user preference distribution X is a fixed one to replace the user latent feature matrix U.

4.2.1 Calculation of the fixed matrices

First, the item topic distribution is constructed from historical reviews via LDA. LDA assumes that each document is a mixture of several topics, and the presence of each word can be attributed to one topic of the document. All reviews to the item v_j in feedback are regarded as the overall "review" d_j of v_j. Suppose there are K topics overall in d_j, its topic distribution proportion is denoted by θ_j, which is a K-dimensional stochastic vector. To be more specific, a topic is denoted by t_k with $k \in \{1, \ldots, K\}$, and each element θ_j^k indicates the proportion of corresponding topic t_k which have been mentioned in d_j. So the topic distribution matrix for all items is represented as $Y = [\theta_1, \ldots, \theta_M]$.

Unlike the item topic distribution, the user preference distribution comes from the users' opinions and preferences via sentiment analysis. Note that STMF model utilizes the sentiment intensity rather than the result of sentiment classification. Let M_i be the number of feedback of u_i. As we know, the rating is in the scale of [1, 5], while the sentiment intensity s_{ij}^1 in section 4.1.1 falls into the range of [-1, 1]. In order to obtain the converted rating r_{ij}^1, the operation in Eq.(1) is necessary for aligning r_{ij}^5 with s_{ij}^1. So the user preference vector of u_i denoted by ρ_i is calculated as follows:

$$\rho_i = \frac{1}{M_i} \sum_{j}^{M_i} [\frac{1}{2}(s_{ij}^1 + r_{ij}^1)\theta_j]$$

where $\theta_j \in Y$ is the topic distribution corresponding to each item v_j of u_i. Therefore, the preference distribution matrix for all users is represented as $X = [\rho_1, \ldots, \rho_N]$.

4.2.2 Objective function of STMF

Since the relative sizes of X and Y in the model need to be kept, two weight vectors are introduced as w_i and w_j. The new rating prediction function is as shown in the following:

$$r_{ij}^5 \sim \hat{r}_{ij}^5 = \mu + b_i + b_j + w_i X_i^T \cdot w_j Y_j \tag{8}$$

where μ is the global bias, i.e., the average of all observed ratings, and b_i and b_j are the user bias for u_i and item bias for v_j, respectively. Thus, the new function of sum-of-squared-error is shown as follows:

$$\zeta = \frac{1}{2} \sum_{i,j} [(r_{ij}^5 - \hat{r}_{ij}^5)^2 + \lambda(\|w_i\|^2 + \|w_j\|^2 + \|b_i\|^2 + \|b_j\|^2)]$$

5 Proposed Methods

In this section, we propose Sentiment Combination Matrix Factorization (SCMF) and its upgraded version (SCMFP) to predict the missing ratings. The structure of SCMF (resp. SCMFP) is shown in Figure 3 (resp. Figure 4).

For SCMF, first, in the preprocessing of data, we use the methods provided in section 4.2.1 of STMF to establish the user preference distribution X and item topic distribution Y via topic analysis and sentiment analysis techniques. Then, the user reliability measure wu_{ij}, which uses the rating r_{ij}^5 and positive votes ratio f_{ij}^P / f_{ij} is established in the way of section 4.1.1 of SBMF+R. After that, based on the distribution matrices, we utilize the rating prediction function as Eq.(8) in section 4.2.2 and propose a new objective function by adding the reliability measure.

As the upgraded version SCMFP of SCMF, we add the item reliability measure to SCMF as an adjustment parameter of the learning rate during training.

5.1 SCMF

With the given set of historical feedback, X and Y are trained from the user reviews with LDA and VADER independently. As the first step of LDA, the text preprocessing operations like stemming, lemmatization, word segmentation, stop-word filtering, and number filtering on the original review data are performed. In order to obtain more explicit and interpretable sentiment intensity, our model differs from STMF and SBMF+R in which a sentiment processing module VADER is applied to

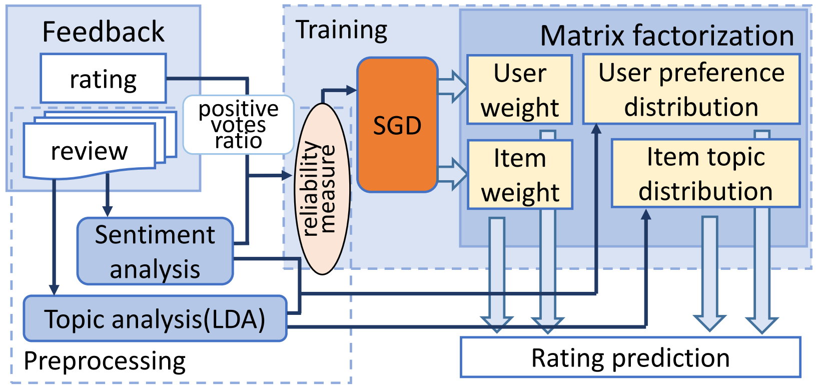


Figure 3: Construction for SCMF.

get s^1_{ij}. In the text preprocessing of VADER, we only normalize the text to get accurate intensities without removing the numbers and stop-words. To get s^5_{ij} from s^1_{ij}, the formula of Eq.(2) is used.

As the weights assigned for X and Y which have been fixed in the model, we make use of the rating prediction function shown in Eq.(8) in section 4.2.2. In the next step, we find that not only the rating needs to be fit, but the user's sentiment also needs to be fit. Thus, we use the reliability measure of users to separately fit the rating and sentiment to obtain a new objective function. With the acquisition of the reliability measure wu_{ij} according to the method in section 4.1.1, we assign the weights to each rating r^5_{ij} and each sentiment intensity s^5_{ij}.

Therefore, the new objective function in order to model X and Y is proposed as follows:

$$\begin{aligned}\zeta =&\frac{1}{2}\sum_{i,j}\{[wu_{ij}(r^5_{ij}-\hat{r}^5_{ij})^2]+[(1-wu_{ij})(s^5_{ij}-\hat{r}^5_{ij})^2]\\ &+\lambda(\|w_i\|^2+\|w_j\|^2+\|b_i\|^2+\|b_j\|^2)]\},\end{aligned}$$

where wu_{ij}, w_i, and w_j represent the reliability factor, user weight, and item weight, respectively. b_i and b_j denote the user bias and item bias, respectively.

A typical way to minimize the objective function is to use the SGD algorithm similar to Eq.(4), which calculates the gradients of w_i, w_j, b_i and b_j for each observed rating r^5_{ij} as follows:

$$\begin{aligned}gw_i &= -[wu_{ij}(r^5_{ij}-\hat{r}^5_{ij})+(1-wu_{ij})(s^5_{ij}-\hat{r}^5_{ij})]X_i^T\cdot w_jY_j+\lambda w_i\\ gw_j &= -[wu_{ij}(r^5_{ij}-\hat{r}^5_{ij})+(1-wu_{ij})(s^5_{ij}-\hat{r}^5_{ij})]w_iX_i^T\cdot Y_j+\lambda w_j\\ gb_i &= -[wu_{ij}(r^5_{ij}-\hat{r}^5_{ij})+(1-wu_{ij})(s^5_{ij}-\hat{r}^5_{ij})]+\lambda b_i\\ gb_j &= -[wu_{ij}(r^5_{ij}-\hat{r}^5_{ij})+(1-wu_{ij})(s^5_{ij}-\hat{r}^5_{ij})]+\lambda b_j\end{aligned}\tag{9}$$

and iteratively updates them in the opposite direction of the gradients.

5.2 SCMFP

In addition to user reliability measure used in SCMF, we establish review reliability measure for each item based on the user reviews and their helpful votes related to the item, which can be seen as the usefulness of feedback. When the review reliability measure is high, the feedback of the item is worth referring into the training of the model. Correspondingly, in our online style of training, we increase the learning rate for a large updating step for the item. Otherwise, i.e, when the review reliability measure is low, we reduce the learning rate for a slight one. As shown in Figure 4, the users' feedback provides reliability measures both for users and items. They will affect the generation of w_i and w_j in matrix decomposition together.

For an item v_j, its item consistency t_j is calculated as the Euclidean distance between the rating r^5_{ij} and the sentiment intensity s^5_{ij} of all feedback for v_j. Where N_j is the number of users who

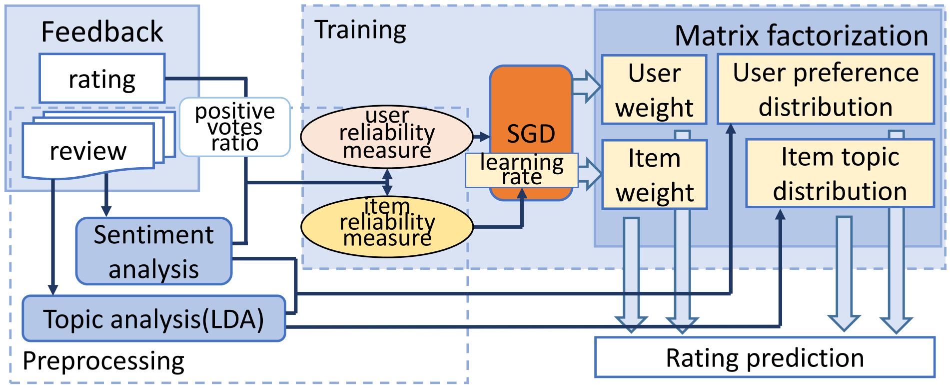


Figure 4: Construction for SCMFP.

posted feedback for v_j, we write the equation for t_j as:

$$t_j = \sqrt{\sum_{i}^{N_j} (r_{ij}^5 - s_{ij}^5)^2}$$

Further, we introduce the reliability measure for rating r_{ij}^5, based on the helpful votes of its corresponding review d_{ij} and item consistency t_j. Inspired by previous study [17], we define it as wv_{ij} as follows:

$$wv_{ij} = \frac{f_{ij}^P / f_{ij}}{med - t_j}$$

where f_{ij}^P / f_{ij} represents the positive votes rate of d_{ij}, and med represents the median value of consistency t_j among all items. The denominator translates t_j into the deviation from med. At last, wv_{ij} is normalized into $[0, 1]$ for the convenience of calculation.

Finally, in the SGD training of SCMFP, in order to adjust the updating step of w_i and w_j, we take place the original constant of learning rate α with wv_{ij}. With the denotation of gradients gw_i, gw_j, gb_i and gb_j following Eq.(9), the updating equations for w_i, w_j, b_i and b_j are written as:

$$\begin{aligned}
w_i &\leftarrow w_i - \alpha \cdot wv_{ij} \cdot gw_i \\
w_j &\leftarrow w_j - \alpha \cdot wv_{ij} \cdot gw_j \\
b_j &\leftarrow b_j - \alpha \cdot wv_{ij} \cdot gb_j \\
b_j &\leftarrow b_j - \alpha \cdot wv_{ij} \cdot gb_j
\end{aligned}$$

where α is a pre-defined constant for SCMFP model. Thus, a trustworthy rating which is with high wv_{ij} brings w_i and w_j significant updates. As a result, the weights of matrices will finally be fine-tuned to find the most suitable value.

6 Evaluation

6.1 Datasets

In the evaluation for the model's performance, we select ten categories of 5-core Amazon review datasets [12] to conduct experiments: "Musical Instruments", "Patio Lawn and Garden", "Automotive", "Instant Video", "Tools and Home Improvement", "Office Products", "Digital Music", "Baby", "Grocery and Gourmet Food", and "Pet Supplies". The datasets are extremely helpful to test the performance of the recommender systems in different scenarios. Each of 5-core datasets contains reviews, ratings, helpful votes, item metadata, links, and so on. We filter out users and items with constraints such that each user and each item have at least five feedback respectively.

Table 1: Statistics of the Amazon datasets.

Dataset	#users	#items	#reviews	avg.ratings	var.rating	avg.sentiments	avg.words	#pos	#total	sparsity	K
Musical	1,429	900	10,261	4.4887	0.8003	4.1650	91.1	16,119	19,066	0.0080	15
Patio	1,686	963	13,272	4.1865	1.1752	3.9332	159.2	42,914	49,859	0.0082	10
Automotive	2,928	1,835	20,473	4.4718	0.8842	4.0500	86.0	31,612	38,603	0.0038	5
Instant	5,130	1,685	37,126	4.2095	1.2511	3.9737	92.0	48,024	74,958	0.0043	30
Tools	16,638	10,217	134,476	4.3654	1.0724	4.0318	110.9	407,895	472,891	0.0008	5
Office	4,905	2,420	53,258	4.3460	0.8653	4.1073	147.5	162,510	183,894	0.0045	15
Digital	5,541	3,568	64,706	4.2225	1.1796	4.0992	200.0	239,161	342,510	0.0033	15
Baby	19,445	7,050	160,792	4.2141	1.3095	4.0867	99.6	285,670	345,537	0.0012	10
Grocery	14,681	8,713	151,254	4.2430	1.1881	4.1102	94.2	237,201	302,126	0.0012	5
Pet	19,856	8,510	157,836	4.2297	1.3825	4.0195	88.8	216,011	250,124	0.0009	10

Table 2: Statistics of the Amazon datasets (continued).

Dataset	avg.wu_{ij}	avg.wv_{ij}	avg.f^P_{ij}/f_{ij}	var.f^P_{ij}/f_{ij}	avg.$(1-c_i)$	var.$(1-c_i)$	avg.$(med-t_j)$	var.$(med-t_j)$
Musical	3.0425	-0.1345	0.2638	0.1764	-0.9674	0.9392	1.4971	4.2425
Patio	-0.4137	0.0967	0.3662	0.2005	-1.3448	1.3080	3.0599	6.8721
Automotive	0.5815	0.0500	0.2782	0.1780	-1.2923	1.0816	3.4105	4.3309
Instant	-0.4390	0.0330	0.2033	0.1297	-1.1451	1.5811	3.1993	17.5988
Tools	-0.3776	0.0995	0.3654	0.2039	-1.5104	1.5758	4.0129	6.1041
Office	-0.1116	0.0618	0.2921	0.1826	-1.4241	1.2537	2.0768	4.3757
Digital	-0.2099	0.0827	0.5187	0.1794	-2.2332	9.9010	5.7494	7.9775
Baby	-0.2484	4.0193	0.2522	0.1629	-1.1577	1.2820	4.5734	11.7065
Grocery	-0.1641	0.0499	0.3028	0.1822	-1.5882	2.3697	5.2009	13.5151
Pet	-0.8776	0.0560	0.2652	0.1796	-1.4295	1.5316	5.3596	14.2718

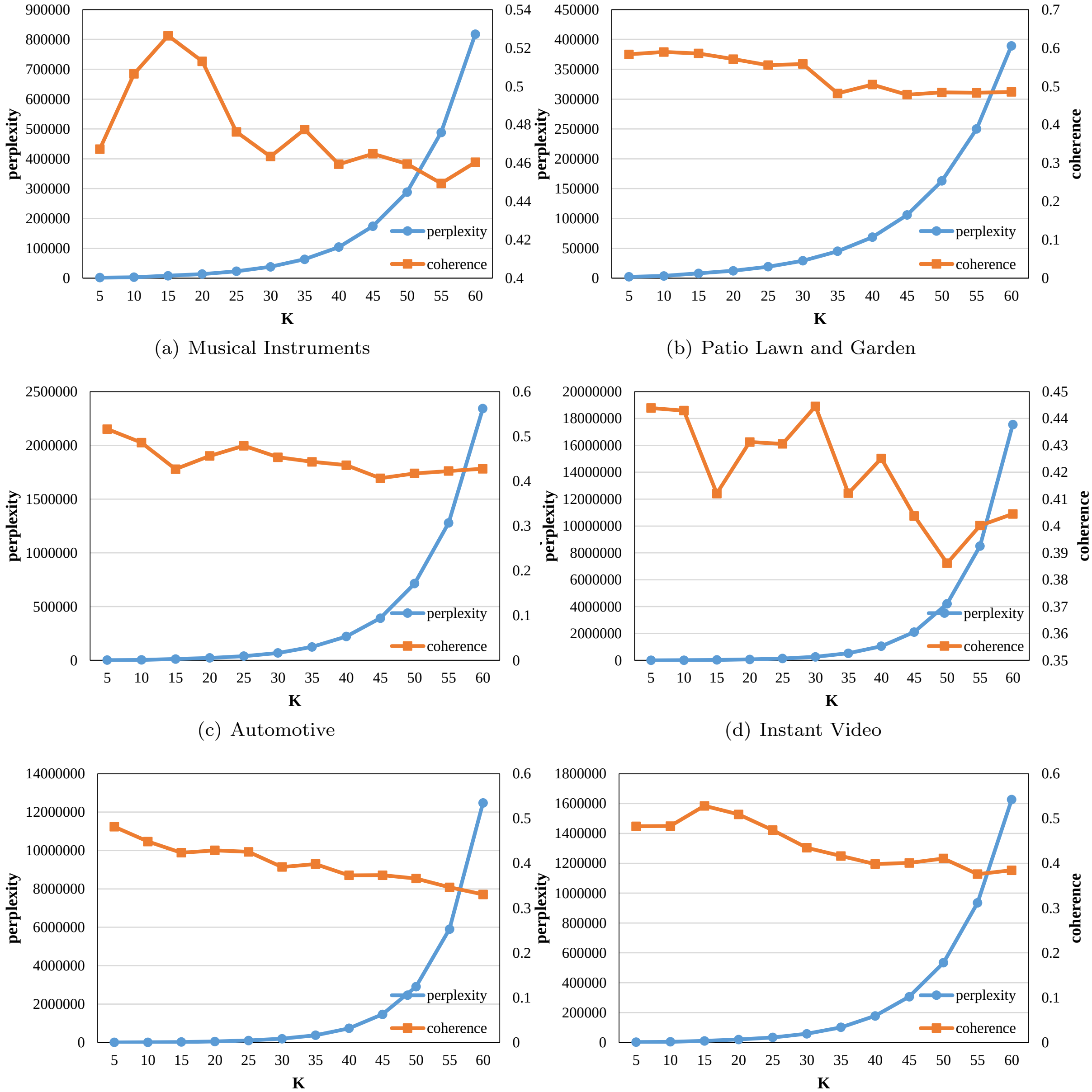
Tables 1 and 2 show the statistics for the datasets. For simplicity, the dataset name is represented as the first word of the name in the following tables. In Table 1, the average of ratings (resp. the average number of words, the sparsity) of a dataset is calculated as #ratings/#reviews (resp. #words/#reviews, #reviews/(#users×#items)). The value of "avg.sentiments" means the average of sentiment intensities of reviews. In addition, the positive votes number and the total votes number for each dataset are shown as #pos and #total, respectively. In Table 2, the average values of wu_{ij} and wv_{ij} and the average and variance values of f^P_{ij}/f_{ij}, $1-c_i$, and $med-t_j$ are shown, respectively.

6.2 Implementation

We compare the proposed models (i.e. SCMF and SCMFP) with the following existing models in our experiment: basic MF, biased MF, PMF, SBMF+R, and STMF. In the experiment, 80% of each dataset is regarded as a training set and 20% as a testing set. We conduct 5-fold cross-validation in training.

In order to implement LDA, we use gensim library in sklearn of Python. Also, the parameter settings for the method described in Table 3 are used to get more accurate training results. In addition, we calculate the perplexity score and coherence score of LDA with topic dimension K varies from 5 to 60. The perplexity [6] is a measure of how well a probability model predicts a sample while the coherence [16] is a measure of topic quality. The smaller the perplexity, the larger the coherence, the better the performance of LDA. More specifically, the perplexity score keeps getting larger as K keeps increasing. However, the maximum coherence scores are mostly different, focusing on 5 to 30. In Figures 5 and 6, we show the scores for each dataset, and the best value of K for each dataset is shown in Table 1. For fairness, we do comparison experiments of the dimensions K of topics, where K is set to 10, 20, and 30 for each method.

For the comparison of methods, first the regularization term and learning rate are fixed as $\lambda = 0.06$ and $\alpha = 0.0002$, respectively, which are decided by experiments. Concretely speaking, we tried various pairs of values $\alpha = 0.0001, 0.0002, \cdots, 0.0007$ and $\lambda = 0.01, 0.02, \cdots, 0.07$ for each dataset and each method, and chose the average of the best values. For all methods, we set the number of epochs of each model to 2000. The weight vectors w_i and w_j are initialized by randomly generated values following uniform distribution over $[0, 1]$.



(a) Musical Instruments

(b) Patio Lawn and Garden

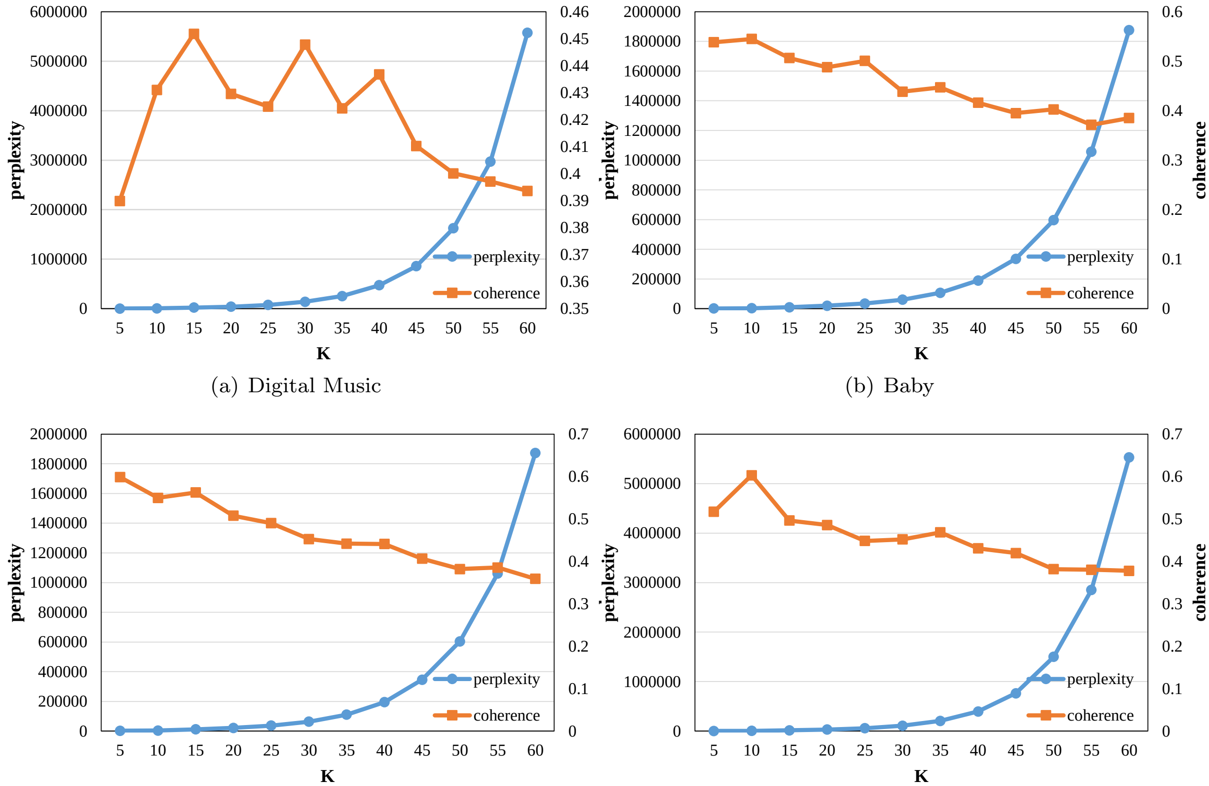
(c) Automotive

(d) Instant Video

(e) Tools and Home Improvement

(f) Office Products

Figure 5: Perplexity and Coherence.



(a) Digital Music (b) Baby

(c) Grocery and Gourmet Food (d) Pet Supplies

Figure 6: Perplexity and Coherence (continued).

6.3 Evaluation Metric

With the problem we have defined, the performance of each model can be measured by observing the accuracy of the prediction, that is, for the ratings in the test set, the difference between the predicted value $\hat{r}_{ij}$ and the real rating value r_{ij} can be evaluated. Thus, we use the commonly used Root Mean Square Error (RMSE) as an indicator, which is calculated as follows:

$$RMSE = \sqrt{\frac{\sum_{i,j}(r_{ij} - \hat{r}_{ij})^2}{T}}$$

where T is the number of feedback in the testing set. The model is considered as better as the obtained RMSE value is getting smaller.

Additionally, in order to further investigate the performance of SBMF+R, STMF and our proposed methods in detail, we re-define the five-level rating values of 1, 2 and 3 as negative, 4 and 5 as positive. Based on this definition, we calculate the accuracy rate of each method for the binary prediction, i.e., polarity (positive or negative) prediction. We consider that, in some cases of applications, the accuracy of the polarity prediction may be more important than the accuracy of the prediction of the exact ratings.

6.4 Results

Table 4 summarizes the results of all datasets with $K = 10$, $\alpha = 0.0002$ and $\lambda = 0.06$, where the best performance of each dataset is emphasized in bold. Table 5 (resp. 6) summarizes the improvement of SCMFP (resp. SCMF) for each dataset. The improvement from each existing method is calculated by $(B - A)/B$, where A is the result of SCMFP (resp. SCMF) and B is the existing method. When

Table 3: Parameter setting for LDA.

Parameter	Value
learning_method	online (EM algorithm)
max_iter	500
learning_offset	50
random_state	0
learning_decay	0.7
batch_size	128

$K = 10$, both SCMFP and SCMF show the best improvement in terms of RMSE on ten datasets as almost 14.1% compared with MF, 7.13% compared with SBMF+R, and 0.69% compared with STMF on average.

In order to ensure there is a statistical significance between the results of SCMF and existing methods (resp. SCMFP and other methods including SCMF) at $K = 10$ respectively, we performed a t-test on the results for each dataset. In Tables 5 and 6, the symbol † means that $p \leq 0.01$. For almost of all cases, the p-values are less than 0.01. That is, comparing with existing methods, SCMF and SCMFP show statistical significance in each dataset. There is also a statistical significance between our proposed methods SCMF and SCMFP as well.

As shown in these results, SCMFP method outperforms other methods including SCMF on the datasets except "Automotive", "Digital Music" and "Baby". If we exclude "Baby" dataset, the average improvement of SCMFP against SCMF (resp. STMF) is 0.63% (resp. 1.25%). Additionally, if we exclude SCMFP, SCMF outperforms the existing methods on the datasets expect "Automotive" dataset. A close analysis against the results of these datasets remains as a future work. In the statistics shown in Table 2, for dataset "Baby", the average value of wv_{ij} is higher than other datasets. In such a case, our method may update w_i and w_j in too large steps in each learning epoch. Thus, depending on item reliability measure wv_{ij}, a dynamic adjustment of its influence on the training process may be needed. For dataset "Digital Music", the average values of f_{ij}^P/f_{ij} and $(med - t_j)$ are higher than other datasets. It means that the obtained $wv_{ij} = (f_{ij}^P/f_{ij})/(med - t_j)$ will become very large or small at some point, which may cause a great impact on the dynamic adjustment.

To further confirm and determine whether there is a statistical significance between the results of SCMFP (resp. SCMF) with different K, we performed a t-test on them, introducing p-value as the lowest level in the observed values of the test statistic. However, we found that the RMSE results of SCMFP (resp. SCMF) lack significant differences, so the results of $K = 20$ and 30 are omitted in the table.

The accuracy of the polarity prediction is shown in Table 7. On each dataset, we can see that SCMFP has achieved the highest accuracy rate and shows the best performance on average. Also, in Table 8 (resp. 9), we use the same calculation method as Table 5 (resp. 6), to summarize the improvement of SCMFP (resp. SCMF) for each dataset, where SCMFP (resp. SCMF) shows the improvement in terms of accuracy of polarity prediction as 1.30% (resp. 0.88%) compared with STMF on average. In addition, we performed a t-test on the results for each dataset. The symbol † in Tables 8 and 9 represents that $p \leq 0.01$, which means that comparing with existing methods, SCMF and SCMFP show statistical significance in each dataset.

7 Conclusion

In this paper, we propose SCMF and SCMFP methods to predict the missing ratings for the recommender systems. From the given textual reviews, the topic distribution and sentiment value are extracted by LDA and VADER, respectively. They are used to directly construct the fixed user preference distribution and item topic distribution matrices instead of the latent factor matrices. Also, in SGD process, the weights for the fixed matrices are iteratively updated by adjusting the ratio between the user reliability factors of each rating and each sentiment intensity. In SCMFP,

Table 4: Performance in terms of RMSE of different methods at $K = 10$, $\lambda = 0.06$ and $\alpha = 0.0002$.

Dataset	MF	PMF	Baised MF	SBMF+R	STMF	SCMF	SCMFP
Musical	1.0639	0.9219	0.9985	0.9168	0.9239	0.9169	**0.9045**
Patio	1.1027	1.0794	1.0514	1.0668	0.9762	0.9692	**0.9614**
Automotive	1.0880	0.9512	0.9955	0.9485	**0.9154**	0.9271	0.9209
Instant	1.1321	1.0923	1.0286	1.0889	0.9612	0.9530	**0.9437**
Tools	1.1574	1.0563	1.0641	1.0443	0.9878	0.9769	**0.9721**
Office	0.9961	0.9481	0.9197	0.9408	0.8648	0.8555	**0.8529**
Digital	1.0917	1.0425	0.9899	1.0406	0.9233	**0.9190**	0.9229
Baby	1.2383	1.1880	1.1748	1.1525	1.0915	**1.0773**	1.1391
Grocery	1.1451	1.0887	1.0828	1.0773	1.0007	0.9964	**0.9930**
Pet	1.2886	1.2150	1.2207	1.1931	1.1356	1.1191	**1.1061**
Average	1.1304	1.0583	1.0526	1.0469	0.9780	**0.9710**	0.9717

Table 5: The improvement in terms of RMSE of SCMFP on all datasets (%). The symbol † means that $p \leq 0.01$.

Dataset	vs MF	vs PMF	vs Baised MF	vs SBMF+R	vs STMF	vs SCMF
Musical	14.98†	1.89†	9.42†	1.33†	2.10†	1.35†
Patio	12.81†	10.93†	8.56†	9.88†	1.52†	0.80†
Automotive	15.37†	3.19†	7.50†	2.91†	−0.59†	0.68†
Instant	16.65†	13.61†	8.26†	13.34†	1.82†	0.98†
Tools	16.01†	7.97†	8.65†	6.91†	1.59†	0.49†
Office	14.38†	10.04†	7.26†	9.34†	1.37†	0.29†
Digital	15.46†	11.47†	6.77†	11.31†	0.05†	−0.42†
Grocery	13.29†	8.80†	8.29†	7.83†	0.77†	0.34†
Pet	14.16†	8.96†	9.38†	7.29†	2.59†	1.15†
Average	**14.79**	**8.54**	**8.23**	**7.79**	**1.25**	**0.63**
Baby	8.01†	4.12†	3.03†	1.16†	−4.36†	−5.74†
Average	**14.11**	**8.10**	**7.71**	**7.13**	**0.69**	**-0.01**

the review reliability factors are used for the adjustment of the learning rate.

In our evaluation, we perform the experiments with ten Amazon review datasets. The results show that the RMSE of rating prediction by our SCMF and SCMFP methods improve significantly comparing to traditional MF methods on average. Additionally, the proposed methods can predict the polarity of ratings more accurately.

In the future, we plan to apply other methods to analyze reviews to build the item topic distribution matrix and the user preference distribution matrix to get better performance.

References

[1] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. SentiWordNet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In Proceedings of the International Conference on Language Resources and Evaluation, 2010.

[2] Yang Bao, Hui Fang, and Jie Zhang. TopicMF: Simultaneously exploiting ratings and reviews for recommendation. In Proceedings of the 28th AAAI conference on artificial intelligence, pages 2–8, 2014.

[3] Guan-Shen Fang, Sayaka Kamei, and Satoshi Fujita. Rating prediction with topic gradient descent method for matrix factorization in recommendation. International Journal of Advanced Computer Science and Applications, 8(12):469–476, 2017.

Table 6: The improvement in terms of RMSE of SCMF on all datasets (%). The symbol † means that $p \leq 0.01$.

Dataset	vs MF	vs PMF	vs Baised MF	vs SBMF+R	vs STMF
Musical	13.82†	0.55†	8.18†	-0.01	0.77†
Patio	12.11†	10.21†	7.82†	9.15†	0.72†
Automotive	14.79†	2.53†	6.87†	2.25†	−1.28†
Instant	15.82†	12.75†	7.34†	12.47†	0.84†
Tools	15.60†	7.51†	8.19†	6.45†	1.11†
Office	14.12†	9.77†	6.99†	9.07†	1.08†
Digital	15.82†	11.84†	7.16†	11.68†	0.47†
Baby	13.00†	9.32†	8.30†	6.52†	1.31†
Grocery	12.99†	8.48†	7.98†	7.51†	0.43†
Pet	13.16†	7.89†	8.33†	6.21†	1.45†
Average	**14.12**	**8.09**	**7.72**	**7.13**	**0.69**

Table 7: Accuracy of polarity prediction in terms of different methods on all datasets.

Dataset	SBMF+R	STMF	SCMF	SCMFP
Musical	0.8520	0.8528	0.8713	**0.8743**
Patio	0.7859	0.7901	0.8044	**0.8067**
Automotive	0.8611	0.8662	0.8723	**0.8757**
Instant	0.8058	0.8166	0.8180	**0.8205**
Tools	0.8263	0.8341	0.8423	**0.8478**
Office	0.8254	0.8410	0.8444	**0.8469**
Digital	0.8126	0.8261	0.8288	**0.8313**
Baby	0.7682	0.7783	**0.7870**	0.7855
Grocery	0.7849	0.7940	0.7959	**0.7970**
Pet	0.7667	0.7713	0.7782	**0.7908**
Average	0.8089	0.8171	0.8243	**0.8276**

[4] Gayatree Ganu, Noemie Elhadad, and Amélie Marian. Beyond the stars: improving rating predictions using review text content. In Proceedings of the 12th International Workshop on the Web and Databases, pages 1–6, 2009.

[5] David Goldberg, David Nichols, Brian M Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. Communications of the ACM, 35(12):61–70, 1992.

[6] Matthew Hoffman, Francis R Bach, and David M Blei. Online learning for latent dirichlet allocation. In Proceedings of the 23rd International Conference on Neural Information Processing Systems, pages 856–864, 2010.

[7] Clayton J Hutto and Eric Gilbert. VADER: A parsimonious rule-based model for sentiment analysis of social media text. In Proceedings of the 8th International AAAI conference on weblogs and social media, 2014.

[8] Dongmin Hyun, Chanyoung Park, Min-Chul Yang, Ilhyeon Song, Jung-Tae Lee, and Hwanjo Yu. Review sentiment–guided scalable deep recommender system. In Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 965–968, 2018.

[9] Olga Kolchyna, Thársis T. P. Souza, Philip C. Treleaven, and Tomaso Aste. Twitter sentiment analysis: Lexicon method, machine learning method and their combination. CoRR, abs/1507.00955, 2015.

Table 8: The improvement in terms of accuracy of polarity prediction of SCMFP on all datasets (%). The symbol † means that $p \leq 0.01$.

Dataset	vs SBMF+R	vs STMF	vs SCMF
Musical	2.61†	2.51†	0.34†
Patio	2.65†	2.10†	0.28†
Automotive	1.70†	1.10†	0.39†
Instant	1.82†	0.48†	0.30†
Tools	2.60†	1.64†	0.66†
Office	2.60†	0.70†	0.29†
Digital	2.30†	0.64†	0.30†
Baby	2.25†	0.93†	−0.19†
Grocery	1.55†	0.38†	0.15†
Pet	3.14†	2.52†	1.61†
Average	**2.32**	**1.30**	**0.41**

Table 9: The improvement in terms of accuracy of polarity prediction of SCMF on all datasets (%). The symbol † means that $p \leq 0.01$.

Dataset	vs SBMF+R	vs STMF
Musical	2.27†	2.17†
Patio	2.36†	1.81†
Automotive	1.30†	0.70†
Instant	1.52†	0.18†
Tools	1.93†	0.98†
Office	2.31†	0.41†
Digital	2.00†	0.34†
Baby	2.45†	1.12†
Grocery	1.40†	0.23†
Pet	1.51†	0.90†
Average	**1.90**	**0.88**

[10] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. Computer, 42(8):30–37, 2009.

[11] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In Proceedings of the 7th ACM conference on Recommender systems, pages 165–172, 2013.

[12] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton 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, pages 43–52, 2015.

[13] Andriy Mnih and Russ R Salakhutdinov. Probabilistic matrix factorization. In Proceedings of the 20th International Conference on Neural Information Processing Systems, pages 1257–1264, 2008.

[14] Yashar Moshfeghi, Benjamin Piwowarski, and Joemon M Jose. Handling data sparsity in collaborative filtering using emotion and semantic based features. In Proceedings of the 34th International ACM SIGIR conference on Research and development in Information Retrieval, pages 625–634, 2011.

[15] Francisco J. Peña, Diarmuid O'Reilly-Morgan, Elias Z. Tragos, Neil Hurley, Erika Duriakova, Barry Smyth, and Aonghus Lawlor. Combining rating and review data by initializing latent

factor models with topic models for top-n recommendation. In Proceedings of the 14th ACM Conference on Recommender Systems, page 438–443, 2020.

[16] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In Proceedings of the 8th ACM International conference on Web search and data mining, pages 399–408, 2015.

[17] Rong-Ping Shen, Heng-Ru Zhang, Hong Yu, and Fan Min. Sentiment based matrix factorization with reliability for recommendation. Expert Systems with Applications, 135:249–258, 2019.

[18] Babak Maleki Shoja and Nasseh Tabrizi. Customer reviews analysis with deep neural networks for e-commerce recommender systems. IEEE Access, 7:119121–119130, 2019.

[19] Yang Sun, Guan-Shen Fang, and Sayaka Kamei. A tensor factorization on rating prediction for recommendation by feature extraction from reviews. In Proceedings of the 7th International Symposium on Computing and Networking, pages 218–224, 2019.

[20] Xiaoteng Wang and Bo Yang. STMF: A sentiment topic matrix factorization model for recommendation. In Proceedings of the 3rd International Conference on Computer and Communication Systems, pages 444–447, 2018.

[21] Weishi Zhang, Guiguang Ding, Li Chen, Chunping Li, and Chengbo Zhang. Generating virtual ratings from Chinese reviews to augment online recommendations. ACM Transactions on intelligent systems and technology, 4(1):1–17, 2013.

[22] Zhan Zhang and Yasuhiko Morimoto. Collaborative hotel recommendation based on topic and sentiment of review comments. In Proceeding of the 9th Forum on Data Engineering and Information Management, 2017.