A Novel Multidimensional Comments and Co-preference Aware Recommendation Algorithm

Yanmei Zhang(✉), Nana Song, Xiaoyi Tang, and Huaihu Cao

Central University of Finance and Economics, Beijing 100081, China
zhangym@cufe.edu.com

Abstract. A recommendation system creates a personalized experience for each customer, which helps companies boost the average order value and the amount of revenue generated from each customer. In a typical recommendation system, comments typify the group wisdom of users, which can reflect their feelings toward the product in multiple dimensions. Co-preference mirrors common preference of a group of users. By mining the multidimensional comments and co-preference relationship comprehensively, it is justifiable to recommend products that both have a good reputation and conform to users' interests. However, the existing related methods have two problems. Firstly, there is lack of further consideration on how to fully utilize comments of products from multiple dimensions for recommendation. Secondly, how to mine co-preference relationship and combine it with multidimensional comments for recommendation is seldom considered. Therefore, a novel recommendation algorithm is proposed, which mines the comments from multiple dimensions and then converges it with co-preference relationship for recommendation. Experiments conducted on two real-world datasets reveal that our proposed method improves the accuracy in terms of MAE and RMSE, compared with state-of-the-art algorithms.

Keywords: E-commerce · Personalized recommendation · Comment mining · Natural language processing · Co-preference relationship

1 Introduction

Recommendation systems in the e-commerce platform, which work as intelligent agents to assist people to locate what they need, are become more and more indispensable. The data that are relatively easy to acquire are ratings [1, 2] and comments [3–5], but the limitations of both are also obvious. First, product comments and ratings are arbitrary, and the actual ratings for products are generally higher [3]. For example, a seller may gain a good reputation because of a particularly high rating of one of the products; therefore, users believe that other products sold by the seller are also good, but the actual situation may differ. Second, ratings and comments are relatively simple and cannot represent users' real concerns about the product. Third, some e-commerce platforms

H. Gao et al. (Eds.): CollaborateCom 2020, LNICST 349, pp. 530–549, 2021.
https://doi.org/10.1007/978-3-030-67537-0_32

(e.g., Taobao) give high ratings by default to users who forget to give ratings, which do not represent the real experiences and feelings of users. Therefore, how to effectively utilize the available information to recommend high-quality products that meets users' demand is the key for recommendation systems.

At present, the related work of recommendation algorithms is mainly divided into two categories: one is the recommendation algorithm based on comments, which digs out the users' concerns from the comments and adds them to the preference model to achieve product recommendation that meets the characteristics of different users. There are still problems such as insufficient reflection of user interest differences. The other is the recommendation algorithm based on co-preference (common preference) relationship, which adds the co-preference factors into the recommendation model and uses the behavioral preference information of the co-preference users to achieve the recommendation for the target users. However, there are still personalized recommendations such as the single emotion of the users in the co-preference relationship.

In response to the above problems, a novel recommendation algorithm named MCCA (Multidimensional Comments and Co-preference Aware) algorithm framework is proposed. From the perspective of two kinds of information, based on the rating matrix, multi-dimensional comments and Co-preference relationships are integrated into the recommendation model to achieve the purpose of accurately positioning users' consumption tendency. In order to further reflect the user's emotional feelings for different products, we have also improved the rating prediction model, which also considers the user's historical ratings, product ratings based on comments, and ratings of co-preference user groups. It has been verified that the algorithm proposed in this paper has improved the prediction effect.

The main contributions of our proposed algorithm are as follows:

First, the MCCA recommendation algorithm is proposed. It mines users' comments from multiple dimensions and co-preference relationship so as to get effectively recommendation results.

Second, experiments were conducted on two real-world datasets of different sizes, and the results showed that our proposed algorithm could achieve more accurate prediction results. Compared with the state-of-the-art algorithms, the average effect improved by 7.2% in terms of mean absolute error (MAE) and root mean square error (RMSE).

2 Related Work

In general, the related work mainly includes two types: recommendation algorithms based on comments and those based on co-preference relationship.

2.1 Recommendation Algorithms Based on Comment Mining

As users usually focus more on comments on products, it is important to mine those comments. Research has been conducted in the field of natural language processing. For example, Turney [4] examined comments at the sentence level and proposed a simple unsupervised learning approach. Zhuang et al. [5] disposed comments at the word level and obtained more personalized user information through fine-grained mining. Studies at the word level include two basic tasks: domain word clustering and emotion analysis.

Domain word clustering refers to extraction from innumerable texts of words that belong to specific domains and automatically grouping them into semantic categories. For example, Zhang et al. [3] clustered words with LDA, and Luo et al. [6] used Word2Vec for clustering. However, LDA differs from Word2Vec [6] in that LDA is based on document information and learns semantic relevance, while Word2Vec is based on adjacent words information and learns semantic similarity.

In terms of emotion analysis, the two main methods are statistical and semantic. The statistical method implies determining the emotional tendency of words with statistical models. For example, Turney [4] established a database of positive and negative seed words. Semantic methods mainly use the existing knowledge base for analysis. For example, Li et al. [7] used WordNet to judge the emotional tendency. Other emotional dictionaries such as SentiWordNet [8] and HowNet [9] also exist.

With regard to recommendation based on comments, Zhang et al. [4] proposed CommTrust, which generated dependency relations of comments and clustered words with LDA to rank sellers. Wei et al. [10] extracted the tag information from the interaction between users and movies, constructed users' preferred theme model, combined users' ratings of movies with the tag, and used the improved SVD algorithm to make personalized recommendations for users. Kharrat et al. [11] proposed a recommendation algorithm based on the semantics of comments with SlopeOne and SVD. Liu [9] constructed an algorithm with collaborative filtering recommendation by considering users' interests and contextual constraints. Ma et al. [12] suggested that users' nonthemed content is helpful in showing the similarity of users' interests, proposed a framework to extract features, and introduced it into the algorithm based on matrix decomposition (MF).

Some studies merge the text information of comments with the collaborative filtering algorithm and obtain better recommendation effects. For example, McAuley et al. [13] believed that the themes of comments reflect the features of users and products, and proposed the Hidden Factors as Topics (HFT). They considered the dimensions of ratings and subjects of comments, and combined the latent factor model (LFM) and LDA. Zhang et al. [14] proposed the explicit factor model (EFM), which extracted the features of products and users through a periodic sentiment analysis of comments and merges them with the hybrid matrix decomposition framework. Zhang [15] focused on the word-level affective analysis of comments and emphasized its key role in cold start, interpretability of recommendation results, and generation of features of products and users. Bao [16] proposed a matrix decomposition model TopicMF that considers ratings and comments simultaneously. The topic is derived from the comments through non-negative matrix decomposition such that the topic distribution parameters are consistent with the corresponding potential user factors and product factors. Ling [17] suggested ratings meet reviews (RMR), which combines the theme model of comments with the mixed Gaussian function model based on ratings, and uses Gibbs sampling to learn the model parameters to improve the accuracy of the recommendation model.

However, recommendation algorithms based on comments make limited use of the multidimensional nature of comments. Moreover, one of the most important differences between comments and ratings is that comments reflect users' feeling about products from multiple dimensions such as quality, price, and logistics. Therefore, it is necessary to examine the comments from multiple dimensions and word levels for personalized recommendations.

2.2 Recommendation Algorithms Based on Co-preference Relationship

Users' preference are different in different cases, so scholars try to model users' preference to achieve more accurate recommendations. For example, Liu [18] designed a three-layer model to consider users' preference, examined the hidden preference layer between users and products, and proposed a model-based collaborative filtering algorithm. A few studies have added co-preference factors into preference modeling for users. For example, Chen et al. considered the co-preference relationship between users in the context of social network [19].

Golbeck [15] combined social networking and online scoring to isolate several features that demonstrated the co-preference characteristics of users. It was found that the co-preference relationship based on the overall similarity could predict user ratings more accurately by experimenting on filmTrust; Georgoulas and Vlachou et al. [20] introduced a user-centered similarity calculation method in the consideration of user's preference. Guo and Xu et al. [21] considered the recommendation of friends, the proposed framework can model user relationships and learn the strength of user relationships to better infer potential co-preference relationships between users.

Guo and Zhang et al. [22] proposed the TrustSVD algorithm that used matrix decomposition and considered the co-preference between users. They think that the co-preference relationship had a very important influence on the user's score. Co-preference users may have similar preference, so the influence of the co-preference user should be taken into account in the user's scoring prediction of the item; Deng et al. [23] used the deep learning for automatic feature selection, introduced the impact of social networks, and proposed a co-preference perception recommendation method based on matrix decomposition (DLMF).

At present, although the recommendation algorithms proposed in some related work do not involve direct co-preference relationships, it is related to the research of our work. Yang and Lei et al. [24] considered the trust factor in the process of matrix decomposition, and proposed the Trust MF model. In the model, trust relationship and trusted relationship were used for user' recommendation. Wei and He et al. [25] proposed the IRCD-CCS and IRCD-ICS models to solve the problem of cold start based on the SDAE model [26] that extracted information features and improved the SVD model [27] (timeSVD++); Dang and Ignat [21] proposed dTrust to avoid simple social recommendation methods using personal information; Golbeck's TidalTrust algorithm [16] provided a way to calculate the weight of users' direct neighbors in the network.

Although there are some related studies, few focuses on mining comments from multiple dimensions. Moreover, since the social network is difficult to obtain on e-commerce platforms, it is more feasible to mine the co-preference relationship hidden between users. As far as we know, there is few work integrating multidimensional comments and co-preference relationship to support recommendation.

3 MCCA Recommendation Method

Users' comments contain a significant amount of information, which mirror their experience and feelings about products from multiple dimensions, including quality, price, experience, and logistics. Therefore, it is necessary to comprehensively mine and utilize users' comments from multiple dimensions for recommendations. Moreover, co-preference relationship based on product interest among users should also be considered. Co-preference relationship represents the users' relationship with the similar interests and preference. In general, users are more willing to purchase the same products with their co-preference friends. If the comments and co-preference relationship can be considered comprehensively, the users' feelings and experiences regarding products can be revealed from multiple aspects, so as to make more accurate recommendations.

Therefore, the MCCA recommendation algorithm is proposed and the framework can be seen in Fig. 1. The upper and bottom rectangles with black borders are the offline and online processes, respectively. The arrows represent the algorithm flow, and the hollow arrow on the left represents the updating process of user-product rating matrix and comments. In the offline process, dependency relations in the form of (head, dependent) are mined from comments. Next, the "head" words are clustered and the rating of every cluster with "dependent" words is calculated. Finally, a comment rating of each product is generated based on the comment. In the online process, when users enter the system, the co-preference relationship set of the users is obtained bas ed on the Pearson similarity of their historical ratings. Finally, users' ratings for products are predicted in order to recommend for users.

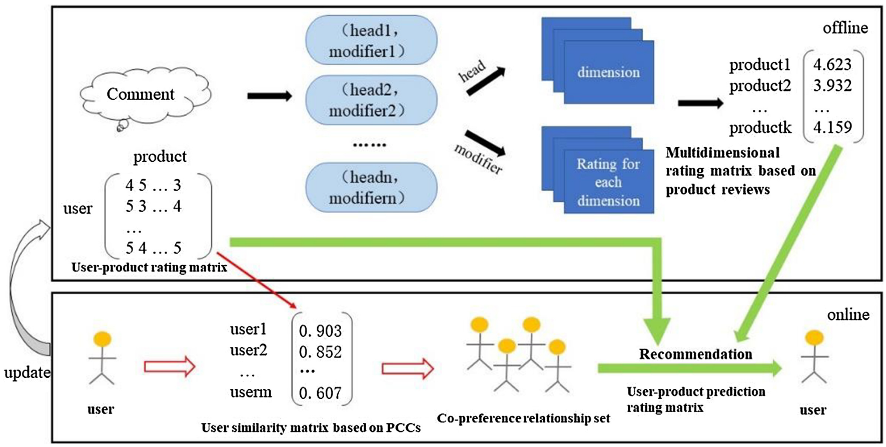

Fig. 1. The framework of MCCA recommendation algorithm

3.1 Comment Mining

Generation of Dependency Relations. Unlike ratings, a user's comments are usually subjective and unstructured; therefore, natural language processing techniques are used to mine and process comments. The Stanford Parser Syntax analysis is a two-step tool used to analyze English comment data. The first step is to use the syntactic analysis tool to address the commodity review corpus and excavate the syntactic dependencies in comments. The second step helps choose the dependency relations, which can reflect the modification of emotional words ("dependent" words) to subject words ("head" words) in the forms of (head, dependent). Figure 2 shows the structure and relationship pairs of a simple English comment sentence "I like it very much" by Stanford Parser.

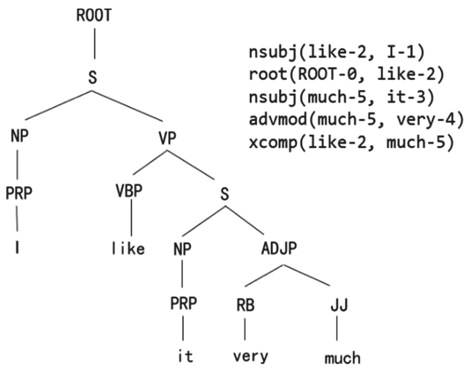

Fig. 2. Analysis result of Stanford Parser tool (*ROOT- the statement to process text; S- phrase combination; NP- noun phrase; VP- verb phrase; PRP- personal pronoun; VBP- link verb; ADJP- adjective phrase; RB - adverb; JJ- adjective; NN - noun; VB - verb; amod - adjective modification; advmod - adverb modification; nsubj - noun subject; acomp - adjective complement; xcomp - lack of subject clause complement*)

In the comments, "dependent" words are used to modify the "head" words, and there is a modified dependency between the "head" words and "dependent" words. The Stanford Parser syntactic analysis tool expresses the dependency between words in the sentence and generates different kinds of dependency pairs. Zhang et al. [5] suggested that the dependency relations in the forms of (head, dependent) can be obtained from four dependency relations: amod (NN, JJ), advmod (VB, RB), nsubj (JJ, NN), and acomp (VB, JJ).

Clustering of Dependency Relations. As users usually focus on different aspects of products, the subject words ("head" words) should be clustered after generating (head, dependent) relations to examine the comments multidimensionally. K-means is used to cluster subject words.

Distance in K-means. Given that the objects of K-means clustering are not numerical values but subject words, Word2Vec is used to represent each word as a vector corresponding to the hidden space. Word2Vec is a tool based on deep learning, which could convert words into vectors and calculate the similarity of the words by measuring the distance of their corresponding vectors. Table 1 describes the words and their similarity that are semantically similar to "man" calculated by the Word2Vec.

Table 1. Semantically similar words and similarity of “man”

Semantically Similar Word	Similarity
Spider	0.60792303
Men	0.5677488
Progs	0.53041357
Henchmen	0.527988

The Selection of K. The cluster number K of each product can be identified by the line graph method. After the completion of clustering, the central point M_i of K clusters, the corresponding cluster C_i of each word, and the distance d between two words are obtained. The sum of the distance from all words to the center point of the cluster is considered as the measurement of the model, which is denoted as D_k:

$$D_k = \sum_{i=1}^{K} \sum_{X \in C_i} d_{x,M_i} \tag{1}$$

Let different K values be the horizontal coordinate and D_k be the vertical coordinate as follows:

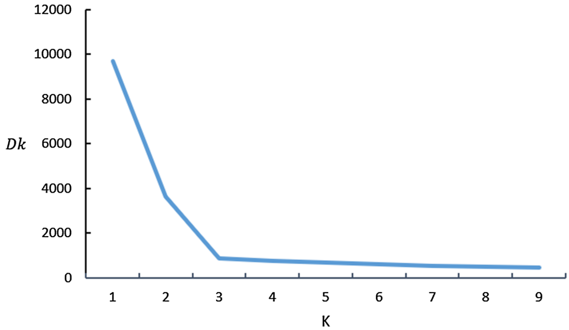

Fig. 3. Cluster number K and corresponding $\boldsymbol{D_k}$

Figure 3 illustrates that the bigger the K, the smaller the D_k, and K = 3 is an inflection point, which indicates D_k goes down very fast when K = 1 to 3 and then stabilizes, so this inflection point is the best K value.

Sentiment Analysis. Sentiment analysis is to examine users’ emotional tendencies in comments that include positive, negative, and neutral. In the MCCA algorithm, SentiWordNet is used to assess emotion words, and positive, negative, and neutral emotion words correspond to +1, –1, and 0 ratings, respectively. Meanwhile, when negative relation is detected, the polarity of emotion words should be inverted.

Comment Ratings. Users usually concentrate on different aspects, and their interest can be found with the word frequency of dimensions. Let w_i denote the weight of each dimension, which is the ratio of the number of (modifier, head) dependency relations in the dimension to those in total. The recommendation index R is defined as follows:

$$\mathrm{R} = 5 * \sum_{i}^{m} w_i * \left(\frac{p_i - n_i}{p_i + n_i}\right), \tag{2}$$

where R denotes the comment rating of product, and m denotes dimension numbers of the product. In addition, w_i represents the weight of each dimension as above. p_i and n_i represent the number of positive and negative emotional words of a dimension, respectively.

3.2 Co-preference Relationship

Co-preference relationship is usually obtained through users' past ratings or interaction, and its value is [0, 1], where 0 indicates no co-preference and 1 indicates complete co-preference. An algorithm based on Pearson similarity is introduced to obtain co-preference relationship. The similarity between user u and user v is defined as follows:

$$S_{u,v} = \frac{\sum_i \left(r_{u,i} - \bar{r}_u\right)\left(r_{v,i} - \bar{r}_v\right)}{\sqrt{\sum_i \left(r_{u,i} - \bar{r}_u\right)^2}\sqrt{\sum_i \left(r_{v,i} - \bar{r}_v\right)^2}}, \tag{3}$$

where $r_{u,i}$ denotes the rating of user u to product i, and $\overline{r_u}$ denotes the average rating of user $u.r_{v,i}$ denotes the rating of user v to product i, and $\overline{r_v}$ denotes the average rating of user v.

Papagelis et al. [28] defined co-preference relationship degree with Pearson similarity. When the Pearson similarity between users is greater than a given threshold, the co-preference relationship degree is $S_{u,v}$; otherwise, the co-preference relationship degree is 0.

$$t_{u,v} = \begin{cases} s_{u,v},\, if s_{u,v} > \theta_s,\, \left|I_{u,v}\right| > \theta_I \\ 0,\, otherwise \end{cases}, \tag{4}$$

where $I_{u,v}$ represents a set of products that both user u and user v rated. θ_s is the threshold of Pearson similarity, which should be determined by experimenting, while θ_I is threshold of $I_{u,v}$ and the value of θ_I is 2.

3.3 Rating Prediction

The MCCA algorithm is proposed based on the TrustSVD algorithm. $\mu + \mathrm{b_j}$ represents the overall rating bias of product j in the TrustSVD algorithm, while it is replaced by c_j, which denotes the comment rating of product j in the MCCA algorithm. The proposed rating prediction model is as follows:

$$\hat{r}_{u,j} = b_u + c_j + q_j^T\left(p_u + |I_u|^{-\frac{1}{2}}\sum_{i \in I_u} y_i + |T_u|^{-\frac{1}{2}}\sum_{v \in T_u} w_v\right), \tag{5}$$

where q_j^T and p_u are the product feature matrix and the user feature matrix, respectively; c_j denotes comment rating of product j ; b_u denotes user bias; I_u is the collection of products rated by user u , and $q_j^T|I_u|^{-\frac{1}{2}}\sum_{i\in I_u} y_i$ represents the change in user factors based on implicit feedback. In addition, T_u denotes the set of co-preference relationship users of user u, and $q_j^T|T_u|^{-\frac{1}{2}}\sum_{v\in T_u} w_v$ denotes the influence of co-preference relationship users.

To avoid the issue of overfitting, the loss function of the model presented adopts the regularization strategy. The objective function is as follows:

$$
\begin{aligned}
\mathrm{L} = {} & \frac{1}{2}\sum_{u}\sum_{j\in I_u}(\hat{r}_{u,j}-r_{u,j})^2+\frac{\lambda_t^I}{2}\sum_{u}\sum_{v\in T_u'}(\hat{t}_{u,v}-t_{u,v}^I)^2+\frac{\lambda}{2}\sum_{u}|I_u|^{-\frac{1}{2}}b_u^2+\frac{\lambda}{2}\sum_{j}|U_j|^{-\frac{1}{2}}c_j^2 \\
& +\sum_{u}(\frac{\lambda}{2}|I_u|^{-\frac{1}{2}}+\frac{\lambda_t}{2}|T_u|^{-\frac{1}{2}})||p_u||_F^2+\frac{\lambda}{2}\sum_{j}|U_j|^{-\frac{1}{2}}||q_j||_F^2+\frac{\lambda}{2}\sum_{i}|U_i|^{-\frac{1}{2}}||y_i||_F^2 \\
& +\frac{\lambda}{2}|T_v|^{-\frac{1}{2}}||w_v||_F^2
\end{aligned}
$$

The model is solved by stochastic gradient descent method, and the Pseudo code is as follows:

Algorithm 1: The MCCA Algorithm

Input: user-product rating matrix R, co-preference relationship matrix T, comment rating matrix C, learning rate a, maximum number of iterations K, λ and λ_t

Output: predicted user-product rating matrix $\hat{R}$

1 Initialize P, Q, W, B. Let *Integer* = 0

2 Do

3 For each $r_{u,j}$ in R, each $e_{u,v}$ in T

4 Calculate the rating of user u to product j, $\hat{r}_{u,j}$

5 Calculate $e_{u,j}=\hat{r}_{u,j}-r_{u,j}$

6 Calculate $e_{u,v}=\hat{t}_{u,v}-t_{u,v}^I$

7 $b_u=b_u-a(e_{u,j}+\lambda|I_u|^{-\frac{1}{2}}b_u)$

8 $p_u=p_u-a(e_{u,j}q_j+\lambda_t e_{u,v}w_v+(\lambda|I_u|^{-\frac{1}{2}}+\lambda_t|T_u^I|^{-\frac{1}{2}})p_u)$

9 $q_j=q_j-a(e_{u,j}(p_u+|I_u|^{-\frac{1}{2}}\sum_{i\in I_u}y_i+|T_u^I|^{-\frac{1}{2}}\sum_{v\in T_u}w_v)+\lambda|U_j|^{-\frac{1}{2}}q_j)$

10 $\forall \mathrm{i}\in I_u, y_i=y_i-\mathrm{a}(e_{u,j}|I_u|^{-\frac{1}{2}}q_j+\lambda|U_i|^{-\frac{1}{2}}y_i)$

11 $\forall \mathrm{v}\in T_u, w_v=w_v-\mathrm{a}(e_{u,j}|T_u|^{-\frac{1}{2}}q_j+\lambda_t e_{u,v}p_u+\lambda|T_v^I|^{-\frac{1}{2}}w_v)$

12 End for

13 Integer = Integer + 1

14 While (Integer < K)

15 Calculate $\hat{R}$ with the latest parameters

16 Return $\hat{R}$

The steps of the MCCA algorithm are as follows: First, input user-product rating matrix R, co-preference relationship matrix T, comment rating matrix C, learning rate a, maximum number of iterations K, λ, and λ_t, and initialize user feature matrix P, product feature matrix Q, co-preference feature matrix W, rating bias matrix B, and iteration number *Integer* (line 1). Next, go into the iteration loop. In each iteration, calculate

$\hat{r}_{u,j}$, $e_{u,j}$, and $e_{u,v}$ with the parameters obtained from the previous iteration, while the initial values are used in the first iteration (lines 4–6). Next, the updated parameters are obtained with Stochastic Gradient Descent (lines 7–11). End the loop when *Integer* up to K. Finally, measure the predicted user-product rating matrix $\hat{R}$ with the latest parameters and return it (lines 15–16).

4 Experiments

In this section, several experiments are conducted to select parameters and compare MCCA to some classical algorithms and state-of-the-art algorithms. The experiments are aimed at addressing the following research questions:

RQ1: How to select the cluster number K of K-means?
RQ2: How to select the threshold of co-preference relationship?
RQ3: How much dose the ratios of training sets and test sets affects the algorithm performance?
RQ4: How does MCCA perform when it runs on two data sets of different sizes and sparsity?

4.1 Dataset

The experiments in the present study use a real-world dataset, that is, a comment dataset of Amazon audio equipment, which includes 1,428 users, 846 products, 10,250 ratings and comments, and Amazon book comment dataset, which includes 68,218 users, 31,191 products, 10,05,011 ratings and comments. Table 2 shows specific information of the datasets.

Table 2. Dataset statistics

Dataset	User	Product	Rating (comment)	Density
Audio equipment	1428	846	10250	0.85%
Book	68218	31191	1005011	0.05%

The co-preference relationship among users is calculated by Pearson similarity. The optimal threshold is determined by the grid search approach. In the experiment, different thresholds were taken, and different sets of co-preference users were obtained. Table 3 describes the number of co-preference users under different thresholds in the two data sets.

Table 3. Co-preference relationship statistics

Threshold	Co-preference (audio equipment)	Co-preference (book, *10^5)
0.1	5998	29.74
0.2	5686	26.28
0.3	5466	21.03
0.4	5027	19.72
0.5	4614	17.91
0.6	4285	14.69
0.7	3843	11.15
0.8	2709	8.42
0.9	1869	6.95
1.0	1050	5.38

4.2 Evaluation Metrics

In the experiments, common metrics, including MAE and RMSE, were used, which could reflect the recommendation accuracy.

1) MAE [6]: It is the average of the absolute error, which can reflect the prediction error by calculating the difference between predicted ratings and actual ratings. The smaller the value, the higher is the recommendation accuracy.

$$\mathrm{MAE} = \frac{\sum_{u,j}\left|\hat{r}_{u,j} - r_{u,j}\right|}{N}, \tag{7}$$

where $\hat{r}_{u,j}$ denotes the predicted rating of user u to product j, and $r_{u,j}$ denotes the actual rating of user u to product j, and N is the total number of ratings in the test set.

2) RMSE [6]: It is the standard deviation between predicted ratings and actual ratings, which measures more rigorously. The smaller the value, the higher is the recommendation accuracy.

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{u,j}\left(\hat{r}_{u,j} - r_{u,j}\right)^2}{N}}, \tag{8}$$

where $\hat{r}_{u,j}$ denotes the predicted rating of user u to product j, and $r_{u,j}$ denotes the actual rating of user u to product j, and N is the total number of ratings in the test set.

4.3 Experiment Algorithms

The comparison algorithm selected in the present study includes a classical random algorithm and state-of-the-art algorithms closely related to MCCA, including CommTrust [3] (the algorithm considering comments), HFT [25] (the algorithm considering comments and ratings), and the algorithms based on singular value decomposition or matrix decomposition such as SVD++ [29], TrustSVD [21], and TrustMF [24]).

1) Random: The algorithm generates user ratings for products randomly, and it is a basic algorithm of the recommendation system.
2) CommTrust [3]: The algorithm fetches dependency relations of comments and clustered words with LDA in order to rank sellers, which represents the group wisdom of users.
3) HFT [13]: Hidden Factors as Topics (HFT) considers the dimensions of ratings and themes of comments together, and combines LFM and LDA. In the experiment, the parameters are set as follows: the learning rate is 0.005, and the number of iterations is 100.
4) SVD++ [29]: The SVD++ considers implicit feedback based on singular value decomposition (SVD) [29]. In the experiment, the parameters are set as follows: $\lambda = 0.1$, the learning rate is 0.005, and the number of iterations is 100.
5) TrustSVD [30]: This algorithm considered trust based on SVD++. In the experiment, the parameters are set as follows: $\lambda = 1.2$, $\lambda t = 0.9$, the learning rate is 0.005, and the number of iterations is 100.
6) TrustMF [23]: Based on matrix decomposition, this algorithm makes recommendations from the viewpoint of trust and being trusted. In the experiment, the parameters are set as follows: $\lambda = 0.001$,$\lambda t = 1$, the learning rate is 0.005, and the number of iterations is 100.

4.4 Experiment Result Analysis

1) Experiment 1: This experiment answers RQ1. For each product i, the best cluster number K_i can be determined by the line graph method. However, for repeatability of the algorithm, a uniform cluster number K should be selected. In this experiment, the best cluster numbers of 100 products K_1~K_{100} are generated first to determine the range of the uniform cluster number K, shown in Fig. 4.

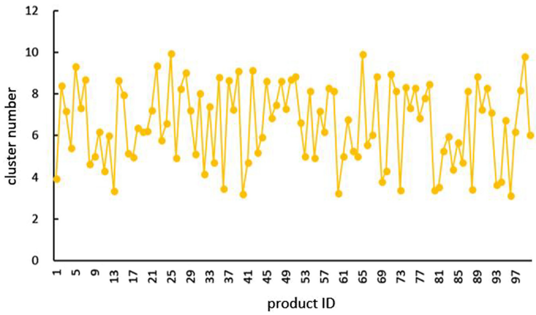

Fig. 4. Distribution of best cluster numbers of 100 products

Figure 4 shows that the range of the best cluster numbers of 100 products is between 3 and 10. Different K values in the range are taken in the following experiments. We choose 80% of the data as a training set and the remaining 20% as a test set. The results are fundamentally the same in other ratios of training set to test set. The parameters in the experiment are set as follows: $\lambda = 1.2$, $\lambda t = 0.9$, the learning rate is 0.005, and the number of iterations is 100.

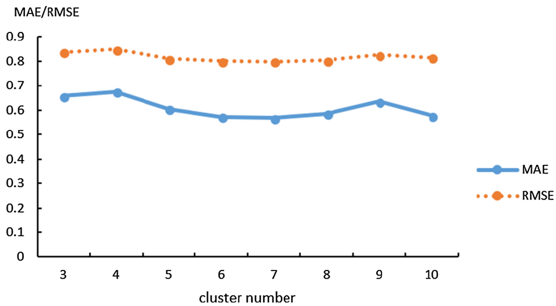

Fig. 5. MAE and RMSE of different cluster numbers

Figure 5 shows that the distribution of MAE and RMSE in the number of clusters is approximately the same. The MAE and RMSE of MCCA are minimum when $K = 7$, which implies that $K = 7$ is the optimal uniform cluster number and should be used in subsequent experiments.

2) Experiment 2: This experiment answers RQ2. Different similarity thresholds are set as 0.7, 0.8, 0.9, and 1. If the Pearson similarity between a target user and other users is larger than the threshold, co-preference relationship will be constructed between them. The MAE and RMSE of MCCA with co-preference relationship in different similarity thresholds are shown in Tables 4 and 5. The settings of ratio of training set to test set and parameters are the same as in Experiment 1.

Table 4. MAE and RMSE in different thresholds (audio equipment)

Threshold	MAE	RMSE
0.1	0.571858	0.809445
0.2	0.571855	0.809465
0.3	0.571846	0.809469
0.4	0.571806	0.809264
0.5	0.571798	0.809257
0.6	0.571802	0.809272
0.7	0.571787	0.809215
0.8	0.571703	0.809068
0.9	**0.571629**	**0.808931**
1.0	0.571706	0.809076

Table 5. MAE and RMSE in different thresholds (book)

Threshold	MAE	RMSE
0.1	0.773954	0.982867
0.2	0.773876	0.982693
0.3	0.773838	0.982625
0.4	0.773812	0.982603
0.5	0.773769	0.982481
0.6	0.773733	0.982441
0.7	0.773791	0.982502
0.8	0.773702	0.982387
0.9	**0.773647**	**0.982277**
1	0.773723	0.982406

Tables 4 and 5 show that the MAE and RMSE are minimum when the threshold is 0.9, which implies that the experiment performance is best when Pearson similarity of at least 0.9 is considered as co-preference relationship.

3) Experiment 3: This experiment answers RQ3. In the experiment, different ratios of training sets and test sets were selected for the experiment. In the algorithm considering co-preference, co-preference relationship was used. A value of 0.9 with better performance was selected as the similarity threshold. In the selection of clustering number in MCCA algorithm, K = 7 is selected as the optimal clustering number. The performance of MCCA algorithm in terms of MAE and RMSE under the two data sets is shown in Figs. 6 and 7.

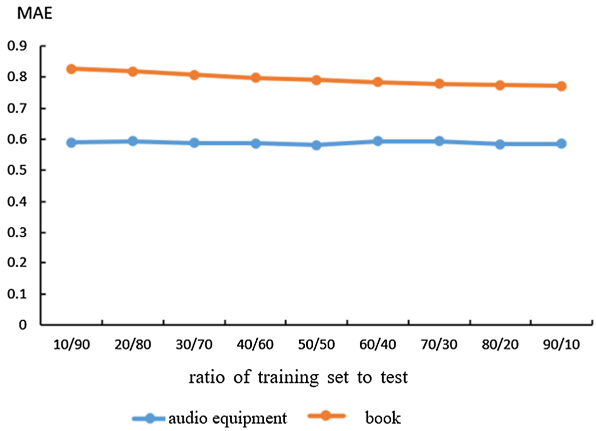

Fig. 6. MAE of different ratio of training set to test

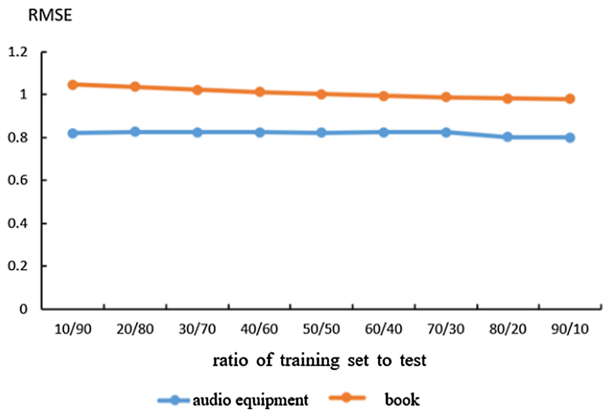

Fig. 7. RMSE of different ratio of training set to test

4) Experiment 4: This experiment answers RQ4. The performance of the CommTrustSVD algorithm is compared with classical Random algorithm as well as state-of-the-art algorithms, including CommTrust, HFT, SVD++, TrustSVD, and TrustMF by calculating the MAE and RMSE. We set K = 7 and threshold of Pearson similarity as 0.9; other settings parameters are the same as in Experiment 1. The results are shown in Figs. 8, 9, 10 and 11.

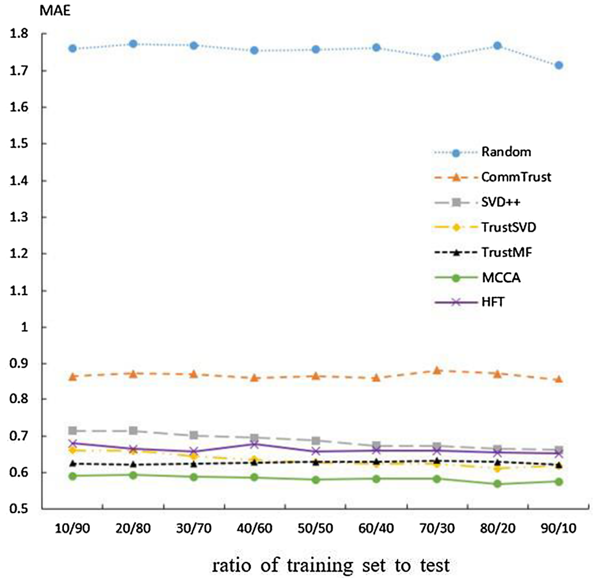

Fig. 8. MAE of different algorithms in audio equipment dataset

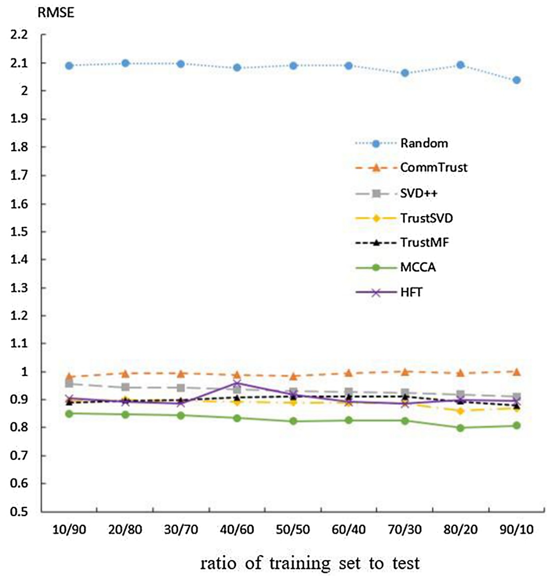

Fig. 9. RMSE of different algorithms in audio equipment dataset

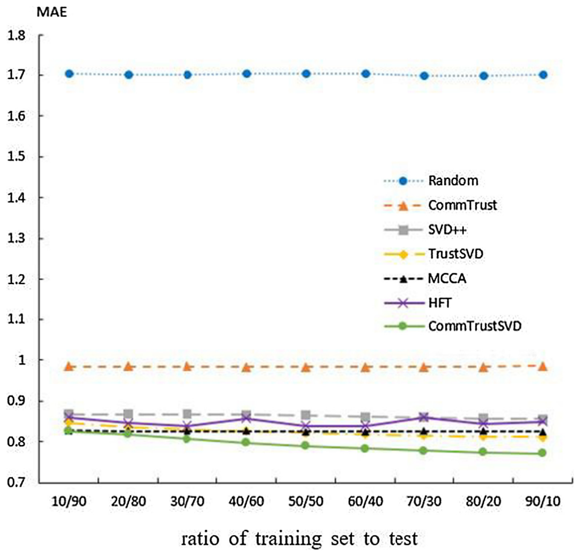

Fig. 10. MAE of different algorithms in book dataset

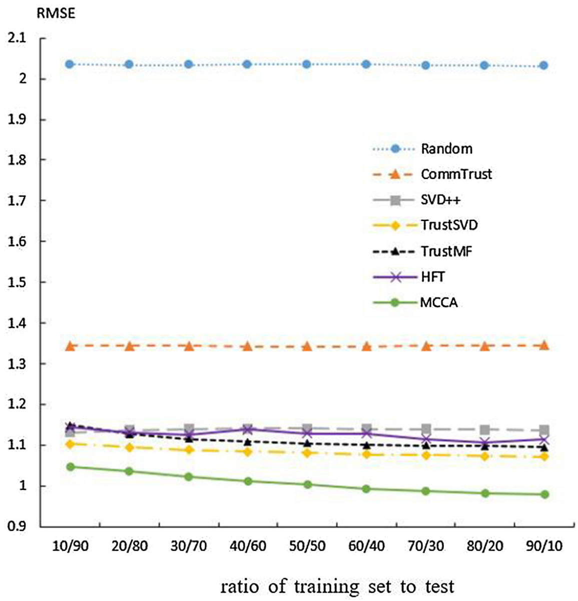

Fig. 11. RMSE of different algorithms in book dataset

Figures 8, 9, 10 and 11 show that in terms of all ratios of training set to test of the two datasets, the proposed algorithm MCCA slightly improved in terms of MAE and RMSE compared with these baseline algorithms. For comparison algorithms, HFT, TrustSVD, and TrustMF are better than Random, CommTrust, and SVD++.

Random is based on neither products nor users' interest, so the effect is the worst. CommTrust is a ranking algorithm that only considers comments and not users' preference. Although its performance is better than the Random algorithm, it does not reflect users' personalization, so the recommendation effect is not satisfactory. The traditional SVD++ algorithm adds implicit feedback information; however, compared with TrustSVD, TrustMF, and MCCA, it does not consider users' co-preference or other information.

HFT considers comments based on collaborative filtering, and the prediction effect is better than SVD++. Notably, in the case of sparse data, the effect of HFT is better than SVD++. This indicates that adding comments into the matrix decomposition algorithm can fairly alleviate data sparsity. However, under the present experimental environment, the performance of HFT algorithm is not as good as that of the algorithm with co-preference, such as TrustSVD, TrustMF, and MCCA.

In terms of MAE, TrustSVD performs better than TrustMF in datasets with higher proportion of training set, while TrustMF performs better than TrustSVD in datasets with lower proportion of training set. It indicates that in the present experimental environment, TrustSVD is suitable for datasets with dense data, while TrustMF is suitable for datasets with sparse data. However, the results of MAE and RMSE are not always consistent. In terms of the ratio of each training set and test set, the RMSE of TrustSVD is lower than that of TrustMF. Therefore, in terms of RMSE, the performance of TrustSVD is better than that of TrustMF. Both these algorithms consider the co-preference relationship among users, but since the dataset does not obtain the explicit co-preference relationship, it is replaced by the co-preference user set with the similarity threshold of 0.9. The results of the experiment show that the effect is favorable.

In conclusion, compared with the baseline algorithms, the MCCA shows certain improvement in terms of MAE and RMSE. Compared with TrustSVD and TrustMF, the experimental effect improved by nearly 7.2%. Compared with HFT, considering both comment and co-preference, the experimental effect improved by nearly 12.5%. MCCA uses the multidimensional comment information of products to fairly alleviate data sparsity. Moreover, compared with TrustSVD, MCCA directly replaces the average rating and item deviation in TrustSVD with the comment score that reflects the characteristics of products, which reduce the errors in the iterative calculation process, improve the accuracy of recommendation, and fairly alleviate the data sparsity.

5 Conclusion

The MCCA algorithm considered comments and co-preference relationship is proposed. Rather than using ratings alone and solving the "good reputation" problem, our proposed method uses comments from multiple dimensions, meanwhile considering users' co-preference relationship by Pearson similarity relations, and finally using the MCCA algorithm to predict user ratings. To some extent, the proposed algorithm can solve the problem of lack of objectivity resulting from overall high ratings, and it can fairly alleviate the data sparsity. Experiments conducted on two real data sets show that the proposed algorithm can make a more accurate rating prediction.

However, natural language processing on word level has limitations. For example, the dependency relations (price, low) indicates that the product is cheap and fine, but "low" may be considered a negative word. Therefore, future research might analyze comments at the sentence level, which may use machine learning for specific context training.

References

1. Ahmed, A.: Modeling trust-aware recommendations with temporal dynamics in social networks. IEEE Access **8**, 149676–149705 (2020)
2. Li, Y.K.: A novel implicit trust recommendation approach for rating prediction. IEEE Access **8**, 98305–98315 (2020)
3. Zhang, X.: CommTrust: computing multi-dimensional trust by mining E-commerce feedback comments. IEEE Trans. Knowl. Data Eng. **26**, 1631–1643 (2014)
4. Turney, P.: Semantic orientation applied to unsupervised classification of reviews. In: Meeting on Association for Computational Linguistics, Philadelphia, pp. 417–424 (2002)
5. Zhuang, L.: Movie review mining and summarization. In: ACM International Conference on Information and Knowledge Management, Arlington, Virginia, pp. 43–50 (2006)
6. Luo, J.: Domain word clustering based on word2vec and semantic similarity. In: 33th Chinese Control Conference, Nanjing (2014)
7. Li, C.: Mining dynamics of research topics based on the combined LDA and WordNet. IEEE Access **7**, 6386–6399 (2019)
8. Vijjini, A.R.: A sentiwordnet strategy for curriculum learning in sentiment analysis. In: 25th International Conference on Applications of Natural Language to Information Systems, Germany, pp. 170–178 (2020)
9. Zhu, S., Li, Y., Shao, Y., Wang, L.: Building semantic dependency knowledge graph based on HowNet. In: Hong, J.-F., Zhang, Y., Liu, P. (eds.) CLSW 2019. LNCS (LNAI), vol. 11831, pp. 525–534. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-38189-9_54
10. Wei, S.: A hybrid approach for movie recommendation via tags and ratings. Electron. Commer. Res. Appl. **18**, 83–94 (2016)
11. Kharrat, F.: Recommendation system based contextual analysis of Facebook comment. In: Computer Systems and Applications, pp. 1–6. IEEE, Agadir (2017)
12. Ma, W.: Your tweets reveal what you like: introducing cross-media content information into multi-domain recommendation. In: 27th International Joint Conference on Artificial Intelligence, Palo Alto, pp. 3484–3490 (2018)
13. Mcauley, J., Leskovec, J.: Hidden factors and hidden topics: Understanding rating dimensions with review text. In: 7th ACM Conference on Recommender Systems, Hong Kong, pp. 165–172 (2013)
14. Zhang, Y.: Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In: International ACM SIGIR Conference on Research & Development in Information Retrieval, Gold Coast, pp. 83–92 (2014)
15. Zhang, Y.: Incorporating phrase-level sentiment analysis on textual reviews for personalized recommendation. In: Eighth ACM International Conference on Web Search and Data Mining, Shanghai, pp. 435–440 (2015)
16. Bao, Y.: TopicMF: simultaneously exploiting ratings and reviews for recommendation. In: Twenty-Eighth AAAI Conference on Artificial Intelligence, Québec (2014)
17. Ling, G.: Ratings meet reviews, a combined approach to recommend. In: 8th ACM Conference on Recommender Systems, CA, pp. 105–112 (2014)
18. Ifrim, G.: Enhancing collaborative filtering by user interest expansion via personalized ranking. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) **42**(1), 218–233 (2011)

19. Chen, T.: Recommendation algorithm based on trust in social network environment. J. Softw. **3**, 771–781 (2017)
20. Georgoulas, K.: User-centric similarity search. IEEE Trans. Knowl. Data Eng. **29**(1), 200–213 (2016)
21. Guo, D.: User relationship strength modeling for friend recommendation on Instagram. Neurocomputing **239**, 9–18 (2017)
22. Yang, B.: Social collaborative filtering by trust. IEEE Trans. Pattern Anal. Mach. Intell. **39**(8), 1633–1647 (2017)
23. Deng, S.: On deep learning for trust-aware recommendations in social networks. IEEE Trans. Neural Netw. Learn. Syst. **28**(5), 1164–1177 (2017)
24. Jamali, M.: A matrix factorization technique with trust propagation for recommendation in social networks. In: ACM Conference on Recommender Systems, pp. 135–142 (2010)
25. Wei, J.: Collaborative filtering and deep learning based recommendation system for cold start items. Expert Syst. Appl. **69**, 29–39 (2017)
26. Wang, H.: Collaborative deep learning for recommender systems. In: 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, pp. 1235–1244 (2015)
27. Koren, Y.: Collaborative filtering with temporal dynamics. Commun. ACM. **53**(4), 89–97 (2010)
28. Liu, Z.H.: Recommendation Algorithm fusing implicit similarity of users and trust. In: 21st IEEE International Conference on High Performance Computing and Communications, Zhangjiajie, pp. 2084–2092 (2019)
29. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, pp. 426–434 (2008)
30. Guo, G.: TrustSVD: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In: AAAI Conference on Artificial Intelligence, Austin, pp. 123–129 (2015)