

Review Article

A New Collaborative Filtering Recommendation Method Based on Transductive SVM and Active Learning

Xibin Wang ^{1,2}, Zhenyu Dai,³ Hui Li ³ and Jianfeng Yang¹

¹School of Data Science, Guizhou Institute of Technology, Guiyang 550003, Guizhou, China

²Special Key Laboratory of Artificial Intelligence and Intelligent Control of Guizhou Province, Guiyang 550003, Guizhou, China

³College of Computer Science & Technology, Guizhou University, Guiyang 550025, Guizhou, China

Correspondence should be addressed to Hui Li; cse.huili@gzu.edu.cn

Received 4 August 2020; Revised 21 September 2020; Accepted 15 October 2020; Published 2 November 2020

Academic Editor: Qingyi Zhu

Copyright © 2020 Xibin Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the collaborative filtering (CF) recommendation applications, the sparsity of user rating data, the effectiveness of cold start, the strategy of item information neglection, and user profiles construction are critical to both the efficiency and effectiveness of the recommendation algorithm. In order to solve the above problems, a personalized recommendation approach combining semisupervised support vector machine and active learning (AL) is proposed in this paper, which combines the benefits of both TSVM (Transductive Support Vector Machine) and AL. Firstly, a “maximum-minimum segmentation” of version space-based AL strategy is developed to choose the most informative unlabeled samples for human annotation; it aims to choose the least data which is enough to train a high-quality model. And then, an AL-based semisupervised TSVM algorithm is proposed to make full use of the distribution characteristics of unlabeled samples by adding a manifold regularization into objective function, which is helpful to make the proposed algorithm to overcome the traditional drawbacks of TSVM. Furthermore, during the procedure of recommendation model construction, not only user behavior information and item information, but also demographic information is utilized. Due to the benefits of the above design, the quality of unlabeled sample annotation can be improved; meanwhile, both the data sparsity and cold start problems are alleviated. Finally, the effectiveness of the proposed algorithm is verified based on UCI datasets, and then it is applied to personalized recommendation. The experimental results show the superiority of the proposed method in both effectiveness and efficiency.

1. Introduction

With the rapid development of the Internet applications and e-commerce, how to quickly and accurately recommend items (including goods, news, services) to different users that they are interested in has become a critical focus and many researchers have been devoted to this area. The personalized recommendation system is an effective method to solve this problem. It actively mines the users' preferences and pushes personalized items to target users.

Currently, the widely used recommendation methods [1] include collaborative filtering (CF), content-based, knowledge-based, and association rules-based recommendation. Among them, the most successful approach is the recommendation based on CF techniques, which can be divided

into two categories: memory-based and model-based methods. The memory-based CF is to filter and recommend items that users are interested in by calculating similarity measures of user preferences. It mainly uses user behavior information to make recommendations. The model-based CF is based on user preference information samples, training a recommendation model, and then calculating and generating recommendation results based on real-time user preferences.

However, the CF algorithm has the following problems:

- (1) *Data Sparsity and Cold Start Problems* [2]. For the memory-based CF algorithm, it is mainly based on the “user-item” rating matrix. When the matrix is very sparse, the performance of finding the nearest

neighbor will decrease significantly. Additionally, it is hard to draw any inferences for users or items, about which it has not yet gathered sufficient information; the so-called cold start problem will occur. In this case, it is more difficult to find data related to the new elements. For the model-based CF, in most implementations, users' historical preferences are stored in sparse matrices, and there are some obvious problems in the calculation of the sparse matrix, including the possibility that a few users' wrong preferences may have a great impact on the accuracy of the recommendation.

- (2) *Item and User Information May Be Discarded in Modeling.* The memory-based CF algorithm mainly makes use of the user's behavior information for recommendation, ignoring the information of items and users, which is of great help to improve the accuracy of recommendation.
- (3) *Data Quality Problem.* In model-based CF, the accuracy of recommendation may heavily rely on the number and quality of users' historical preference data.
- (4) *Scalability Issues.* With the advent of the era of big data, the rapid growth of users and items poses severe challenges to the scalability of the traditional CF algorithm.

In order to solve the data sparsity problem, many researchers try to use a small number of labeled data and machine learning methods such as classification, clustering, and dimension reduction to enhance the dataset. However, these methods have a common problem: when there are few labeled samples used in model construction, the prediction accuracy often is not high enough. In fact, in the real-world application scenarios, it can be found that most of the samples have no label information, and only very few samples are with labels, often resulting in high sparseness of the dataset, and even a "cold start" problem, which is quite unfavorable for discovering users' potential preferences. Additionally, the manual annotation method has the problems of high cost and is time-consuming [3]. Thus, how to combine the limited labeled samples and a large number of unlabeled samples to build a "user-item" association relationship model to predict users' interest preference for personalized recommendation has become an urgent issue.

Semisupervised learning (SSL) and active learning (AL) can effectively solve the problem of building a high-performance model with only a small amount of labeled data. SSL refers to labeling and utilizing unlabeled data according to some information that can be learned by oneself. On the contrary, AL is interactively exploring the unknown information of unlabeled data according to certain strategies and labeling them with domain knowledge. In this procedure, a small number of labeled "user-item" association data and a large number of unlabeled data are used to construct a model-based personalized recommendation; meanwhile, the ability to discover users' potential preferences is improved.

Based on the aforementioned idea, a semisupervised SVM recommendation algorithm based on AL is proposed in this paper, which uses the minimum principle of feasible domain segmentation in AL strategies to query the most informative samples for labeling. Furthermore, to make better use of the distribution characteristics of unlabeled data during the training process, a graph-based manifold regularization term was introduced into the objective function. Simultaneously, to further utilize the users' label data, valuable review information is mined and added to the feature vector to extract the users' potential preferences.

The main contributions of this paper include the following:

- (1) A method combining SSL and AL is proposed to solve the problem of recommendation quality degradation caused by the scarcity of labeled data in real scenarios. This method is very suitable for application scenarios where the data in the recommendation system is extremely sparse and the data filling quality cannot be guaranteed. Because AL strategy attaches great importance to the quality of data, only a small number of high-quality items are chosen.
- (2) An AL strategy is proposed in this paper to identify those unlabeled samples that cause the largest reduction of version space for labeling. Meanwhile, the batch sampling mode can be also used in the sample labeling process to obtain better training efficiency.
- (3) This paper combines item and user information as the features of the prediction model. And in order to achieve a better performance of the annotation, the users' preference/behavior information (label information) is used as an important index for querying and labeling unlabeled samples to guide the labeling process.
- (4) AL selectively interacts with users and asks for information such as item ratings, which can supplement those aspects of interest with sparse data, helping the interest model to be comprehensive, and achieve a better performance of the recommendation system.

2. Related Work

Traditional collaborative filtering algorithms are mainly divided into three categories: user-based collaborative filtering (UserCF), item-based collaborative filtering (ItemCF) [4], and hybrid collaborative filtering algorithms based on both [5, 6]. In recent years, many researchers have focused on solving data sparsity, cold start problem, neglecting item and user information (demographic information), and scalability problems of collaborative filtering. For the first four problems, the following strategies are frequently utilized:

- (1) Employ machine learning algorithm to predict unrated data, such as naive Bayesian [7], support vector machine (SVM) [8], neural networks [9], topic model [10], and deep learning collaborative filtering

algorithms [11]. These strategies usually employ classification algorithm to fill data; the performance of them largely depends on the quality and quantity of the training samples. However, in most real application scenarios, there are only a few labeled samples; thus, the prediction accuracy often tends to be not satisfactory. Therefore, some researchers proposed personalized recommendation method based on active learning to achieve high score prediction. In [12], a user-centered method is developed, and a model based on the interaction between conversation and collaborative filtering is proposed. By using this model, users can provide relevant information such as ratings under specific motivation; thus, the users can clearly know the benefits of what they are doing; meanwhile, their probability of active ratings is increased. In [13], matrix decomposition was incorporated into the tree structure, employed to accelerate the tree construction and predict the ratings of tree nodes. Since the item with the highest predicted rating may be the user's favorite item, the idea in [14] is to predict the rating of unlabeled items with the highest predicted ratings. Reference [15] proposes a binary prediction method, which changes the nonnull value in the "user-item" rating matrix to 1 and the null value to 0, then predicts the position marked as 0 in the converted matrix, and judges the probability that the user knows the item according to the predicted value. It maximized the possibility of users' actively rating. Reference [16] proposes a novel multi-label active learning approach for web service tag recommendation, which can identify a small number of most informative web services to be tagged by domain experts. Furthermore, it minimized the domain expert efforts by learning and leveraging the correlations among tags to improve the active learning process. Reference [17] proposes leveraging the idea of pool-based active learning to realize a scalable service classification approach. Instead of manually labeling a large number of services to construct a complete training set, the approach starts with a base classifier with a small set of training set and iteratively asks for the labels of the most informative services outside of the initial training set.

Inspired by the above results, this paper introduces semisupervised learning and active learning strategy into collaborative filtering, improving the existing semisupervised learning and active learning algorithms in view of the practical problems existing in collaborative filtering algorithm.

- (2) Use the similarity of user preferences or the similarity of item categories to fill data. For instance, in [18], the similarity is calculated after filling the ratings of the target item and the neighbor item that are not rated together. In [19], the trust model is employed to fill data, and in [20], the similarity is obtained after clustering of users or items. However,

these methods have the same problem: they require a large number of high-quality labeled data, and they do not take into account the users' behavior information, the impact of item, and demographic information on the recommendation effect.

In addition, there are also some methods that are aimed at reducing the computational complexity of recommendation model by using dimension reduction techniques, such as graph-based dimension reduction [21], implicit topic analysis (probabilistic latent semantic analysis, PLSA) [22], potential Dirichlet analysis (LDA) [23], matrix factorization [24], singular value decomposition (SVD) [25], kernel matrix factorization [26], and graph-based methods [27]. To some extent, these methods reduce the time complexity of the model but also lead to the loss of some useful information, meanwhile, ignoring the item information and demographic information.

Based on the above analysis, this paper proposes a recommendation algorithm that combines active learning and semisupervised support vector machines techniques, which considers not only user behavior information but also item information and demographic information. And they are employed to enhance the query and label unlabeled samples and finally result in a better implemented item recommendation.

It should be noted that, during the procedure of analyzing the demographic information and item information, it was found that if users have similar demographic information and favorite item set, they will tend to have similar preferences for many items. It just meets the needs of analysis and clearly means that entities with the similar properties will have the same class label. And the proposed high-quality recommendation system significantly benefits from this data characteristic.

3. Semisupervised Support Vector Machines and Active Learning

3.1. TSVM (Transductive Support Vector Machine). TSVM is a maximum interval classification method based on the hypothesis of low-density segmentation [28]. Similar to the traditional support vector machine, it finds the classification hyperplane with the largest interval as the optimal classification hyperplane; meanwhile, it combines the unlabeled data and labeled data together to train the classification model.

Assume a set of labeled samples with independent and identical distribution:

$$\{(x_1, D_1), \dots, (x_l, D_l)\} \in R^n \times R, i = 1, \dots, l, y_i = \{-1, +1\}. \quad (1)$$

And the unlabeled samples are denoted as follows:

$$\{x_{l+1}, \dots, x_{l+u}\}. \quad (2)$$

Generally, the learning process of TSVM can be considered as the process of solving the following optimization problem:

$$\begin{aligned}
& (y_1, \dots, y_n, \omega, b, \xi_1, \dots, \xi_l, \xi_{l+1}, \dots, \xi_{l+u}) \\
\min & \frac{1}{2} \|\omega\|^2 + C_1 \sum_{i=1}^l \xi_i + C_2 \sum_{i=l+1}^{l+u} \xi_j \\
& \forall_{i=1}^l: y_i [\omega \cdot x_i + b] \geq 1 - \xi_i; \xi_i \geq 0 \\
\text{s.t.} & \forall_{i=l+1}^{l+u}: y_j [\omega \cdot x_j + b] \geq 1 - \xi_j; \xi_j \geq 0,
\end{aligned} \tag{3}$$

where C_1 and C_2 are set by the user to control the degree of punishment for wrongly classified samples. C_2 is the “impact factor” of unlabeled data during training; $C_2 \xi_j$ is called the “impact term” of the j -th unlabeled sample in the objective function.

The training process of TSVM is as follows:

Step 1. Train the initial classifier by the labeled samples using inductive learning with the specified parameters C_1 and C_2 . During the procedure, the estimated number N of positive samples in unlabeled samples also should be specified.

Step 2. Set C_{temp} as a temporary impact factor and compute the decision function values of all unlabeled samples using the initial classifier. The samples having the N largest decision function values will be labeled as positive, and the remaining unlabeled samples are labeled as negative.

Step 3. Retrain the SVM model based on all labeled samples. And then use the newly trained classifier to switch the labels of one pair of different labeled unlabeled samples, using a certain rule to make the value of the objective function in formula (3) minimized as much as possible. This step is repeated until no pair of examples meet the switching condition.

Step 4. Increase the value of C_{temp} uniformly, and return to Step 3. When $C_{\text{temp}} \geq C_2$, the algorithm is terminated and the labels of all unlabeled samples are output.

3.2. AL (Active Learning). The traditional machine learning method is training and learning over a given set of labeled samples to induce a learning model, which is called “inductive learning.” However, in real applications scenarios, the labeled samples are very limited, and labeling a large number of unlabeled samples is time-consuming, labor-intensive, and tedious. In order to reduce the labeling cost as much as possible and reduce the number of needed training sample sets, active learning method is proposed to solve the problem of the lack of labeled samples and optimize the classification model. During the training of AL learner, it actively identifies the most informative unlabeled samples and submits them to users or domain experts for labeling and then adds the labeled samples into the training set to participate in the next round of training. Therefore, even if the initial training set is small, it still can obtain a relative higher classification accuracy. In this way, the cost of labeling samples and training high-performance classifiers can be reduced [29].

In AL strategies, the main task is to determine which unlabeled sample has the most information or the most uncertain and is inquired, and this inquiry strategy is the focus of research. According to different problem scenarios and sample selection strategies, AL is divided into the following three types: membership query synthesis, stream-based selective sampling, and pool-based sampling.

In item-based recommendations, there is less information about the “user-item” association information (the user label items). TSVM is an effective method to solve the lack of labels problem; it can make better use of unlabeled data to improve the prediction accuracy of the classifier. However, due to its inherent characteristics, its effectiveness in practical applications is not very satisfactory. Inspired by the literature [30–32], this paper proposes a new semisupervised support vector machine method based on AL techniques, which can combine the advantages of these two algorithms to overcome the defects of TSVM and identify the samples that have the greatest impacts on classifier performance, meanwhile, significantly reducing the burden of users’ annotation task.

4. A New TSVM Algorithm Based on Active Learning (AL)

In this section, a new TSVM algorithm based on active learning (AL) named TSVM-(AL + Graph) is proposed, which combines the advantages of semisupervised learning and AL techniques. In the approach, in order to take advantage of the data manifold structure, a regularization term is added to penalize any “abrupt changes” of the evaluated function value. And then, an unlabeled data selection strategy named “maximum-minimum segmentation” method is designed for AL. The detail of TSVM-(AL + Graph) algorithm is described as follows.

4.1. Integrate Manifold Regularization Term into Objective Function. In order to build the TSVM-(AL + Graph) model, it should add a regularization term defined on unlabeled samples into the traditional SVM optimization function, and the TSVM optimization problem is as follows:

$$\min \frac{1}{2} \|\omega\|^2 + C_1 \sum_{i=1}^l H_1(y_i f(x_i)) + C_2 \sum_{i=l+1}^{l+u} H_1(|f(x_i)|), \tag{4}$$

where $H_1(\cdot) = \max(0, 1 - \cdot)$ is the classic Hinge loss function used to penalize labeled data and $H_1(|\cdot|) = \max(0, 1 - |\cdot|)$ is a symmetric Hinge loss function used to penalize unlabeled data. In formula (4), when $C_2 = 0$, the problem turned to be the traditional SVM optimization problem; when $C_2 > 0$, the unlabeled data will be penalized that is inside the margin. However, this optimization problem’s solution space is a nonconvex hat shape, which makes it hard to obtain a satisfactory solution. In order to efficiently solve this problem, the method in [33] is employed: the loss function for unlabeled data is replaced by the Ramp loss function, and it is decomposed into a sum of a Hinge loss

function and a concave loss function. The expression of the Ramp loss function is

$$R_s(z) = H_1(z) - H_s(z) = \min(1 - s, \max(0, 1 - z)), \quad (5)$$

where $H_1(z)$ is the Hinge loss function, $H_s(z)$ is a concave loss function, and its corresponding expression is $H_s(z) = \max(0, s - z)$, s is a preset parameter, and $-1 < s \leq 0$. In this paper, set $s = -0.3$.

In this case, equation (4) can be rewritten as

$$\min \frac{1}{2} \|\omega\|^2 + C_1 \sum_{i=1}^l H_1(y_i f(x_i)) + C_2 \sum_{i=l+1}^{l+2u} R_s(y_i f(x_i)). \quad (6)$$

According to [33], the objective function corresponding to TSVM can be solved by CCCP method as follows:

$$\min \frac{1}{2} \|\omega\|^2 + C_1 \sum_{i=1}^l \xi_i + C_2 \sum_{i=l+1}^{l+2u} \xi_i + \sum_{i=l+1}^{l+2u} \beta_i f(x_i)$$

$$\frac{1}{u} \sum_{i=l+1}^{l+2u} f(x_i) = \frac{1}{l} \sum_{i=1}^l y_i \quad (7)$$

s.t.

$$y_i f(x_i) \geq 1 - \xi_i, \forall 1 \leq i \leq l + 2u$$

$$\xi_i \geq 0, \forall 1 \leq i \leq l + 2u,$$

where β_i is related to the derivative of the loss function and can be expressed as

$$\beta_i = \begin{cases} C_2 R'_s[y_i f(x_i)], & \text{if } i \geq l + 1, \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

$$= \begin{cases} C_2, & \text{if } y_i f(x_i) < s \text{ and } i \geq l + 1, \\ 0, & \text{otherwise.} \end{cases}$$

In order to obtain the geometrical structure of data, a common solution is to define L' as a function of Laplacian graph. In this way, the structure of the data manifold can be explored by adding a regularization term that penalizes any “abrupt changes” of the function values, which is evaluated on neighbor samples in the Laplacian graph. Then, the optimization problem corresponding to TSVM can be expressed as

$$\min \frac{1}{2} \|\omega\|^2 + C_1 \sum_{i=1}^l \xi_i + C_2 \sum_{i=l+1}^{l+2u} \xi_i + \sum_{i=l+1}^{l+2u} \beta_i f(x_i) + C_3 f^T L' f$$

$$\frac{1}{u} \sum_{i=l+1}^{l+2u} f(x_i) = \frac{1}{l} \sum_{i=1}^l y_i$$

s.t.

$$y_i f(x_i) \geq 1 - \xi_i, \forall 1 \leq i \leq l + 2u$$

$$\xi_i \geq 0, \forall 1 \leq i \leq l + 2u, \quad (9)$$

where C_2 controls the influence of unlabeled samples over the objective function and C_3 controls the influence of graph-based regularization term. If $C_3 = 0$, TSVM will ignore the manifold information of the training data.

If the solution to the above optimization problem is ω , then the optimization problem (9) can be rewritten as

$$\min \frac{1}{2} \alpha^T K \alpha + C_1 \sum_{i=1}^l \xi_i + C_2 \sum_{i=l+1}^{l+2u} \xi_i + \sum_{i=l+1}^{l+2u} \beta_i y_i \left(\sum_{j=1}^{l+2u} \alpha_j K(x_i, x_j) + b \right) + C_3 \alpha^T K^T L K \alpha$$

$$\frac{1}{2u} \sum_{i=l+1}^{l+2u} \left(\sum_{j=1}^{l+2u} \alpha_j K(x_i, x_j) + b \right) = \frac{1}{l} \sum_{i=1}^l y_i \quad (10)$$

s.t.

$$y_i \left(\sum_{j=1}^{l+2u} \alpha_j K(x_i, x_j) + b \right) \geq 1 - \xi_i, \xi_i \geq 0.$$

By introducing the Lagrange multiplier and solving its dual problem, we can get the corresponding decision function:

$$f(x) = \sum_{i=1}^{l+2u} (y_i \bar{\rho}_i + \gamma_i) K(x_i, x) + b, \quad (11)$$

where $\bar{\rho} = \rho - \beta$, ρ , and γ_i are Lagrange multipliers.

4.2. *Principle of “Maximum-Minimum Segmentation”.* In equation (4), assuming $R(f, L)$ is the objective function, we can get

$$R(f, L) = \sum_{i=1}^l \max(0, 1 - y_i f(x_i)) + \sum_{i=l+1}^{l+u} \min(1 - s, \max(1 - y_i f(x_i))) + \frac{\lambda}{2} \|f\|_H^2. \quad (12)$$

In order to identify the most informative samples, we select the unlabeled example x^* that leads to a small value for the objective function regardless of its assigned class label y^* (positive or negative labels). Based on this idea, “maximum-minimum segmentation” strategy can be described as follows:

$$\min_{x^* \in u} \max_{y^* \in \{-1, 1\}} R(f, L \cup (x^*, y^*)). \quad (13)$$

Furthermore, it can be expressed as

$$\min_{x_j^* \in u} \max_{y_j^* \in \{-1, 1\}} \min_{f \in H} \left(\sum_{i=1}^l \max(0, 1 - y_i f(x_i)) + \sum_{i \in u \cup \{j\}} \min(1 - s, \max(1 - y_i f(x_i))) + \frac{\lambda}{2} \|f\|_H^2 \right). \quad (14)$$

Assuming the optimal decision function f^* can be obtained from formula (4), then formula (14) can be simplified as

$$\begin{aligned} & \min_{x_j^* \in u} \max_{y_j^* \in \{-1, 1\}} R(f, L \cup (x^*, y^*)) \\ & \approx \min_{x_j^* \in u} \max_{y_j^* \in \{-1, 1\}} \left(\sum_{i=1}^l \max(0, 1 - y_j^* f^*(x_j^*)) + \sum_{i=l+1}^{l+u} \min(1 - s, \max(1 - y_j^* f^*(x_j^*))) \right) \\ & = \min_{x_j^* \in u} (\max(0, 1 - f^*(x_j^*)), 1 + f^*(x_j^*)) + \min(1 - s, 0, 1 - f^*(x_j^*), 1 + f^*(x_j^*)) \\ & = \min_{x_j^* \in u} (1 + |f^*(x_j^*)|) \\ & = \min_{x_j^* \in u} |f^*(x_j^*)|. \end{aligned} \quad (15)$$

Through the above analysis, it can be found that the “maximum-minimum segmentation” method is to select the unlabeled samples closest to the optimal hyperplane f^* , which is trained on the current labeled sample set. In the next section, the principle of “maximum-minimum segmentation” will be applied into active learning to construct the proposed TSVM-(AL + Graph) algorithm.

4.3. *TSVM-(AL + Graph) Algorithm.* Given the training sample set and kernel function *Kernel*, the version space is defined as the set of classification hyperplanes; it means the set of all samples that training samples can be divided in the feature space H_{kernel} . Generally, the version space can be defined as

$$\text{Version} = \{f \in H_{\text{Kernel}} | \forall i \in \{1, 2, \dots, l + u\}, y_i f(x_i) > 0\}. \quad (16)$$

The principle of using active learning to choose unlabeled samples for annotation is to identify the samples that lead to the largest reduction of version space. Since formula (14) is equivalent to $\min_{x_j^* \in u} y_j^* f^*(x_j^*)$, thus there exists Proposition 1.

Proposition 1. *Set the version space determined by $l + u$ samples:*

$$\text{Version} = \{f \in H_{\text{Kernel}} | \forall i \in \{1, 2, \dots, l + u\}, y_i f(x_i) > 0\}. \quad (17)$$

Randomly labeled samples (x_{l+2}, y_{l+2}) and (x_{l+3}, y_{l+3}) and then two new version spaces $\text{Version}_{l+2}^{\text{new}}$ and $\text{Version}_{l+3}^{\text{new}}$ can be obtained. If $y_{l+2}f(x_{l+2}) > y_{l+3}f(x_{l+3})$, then $\text{Area}(\text{Version}_{l+2}^{\text{new}}) > \text{Area}(\text{Version}_{l+3}^{\text{new}})$, where $\text{Area}(\text{Version})$ denotes the size of the version space.

Proof. After labeling the samples (x_{l+2}, y_{l+2}) and (x_{l+3}, y_{l+3}) , new version spaces $\text{Version}_{l+2}^{\text{new}}$ and $\text{Version}_{l+3}^{\text{new}}$ are obtained.

Version space $\text{Version}_{l+2}^{\text{new}}$:

$$\text{Version}_{l+2}^{\text{new}} = \left\{ \omega \left| \begin{array}{l} \|\omega\| = 1, \\ y_i(\omega \cdot \phi(x_i) + b) > 0, \quad i = 1, 2, \dots, l+u, \\ y_{l+2}(\omega \cdot \phi(x_{l+2}) + b) > 0. \end{array} \right. \right\} \quad (18)$$

Version space $\text{Version}_{l+3}^{\text{new}}$:

$$\text{Version}_{l+3}^{\text{new}} = \left\{ \omega \left| \begin{array}{l} \|\omega\| = 1, \\ y_i(\omega \cdot \phi(x_i) + b) > 0, \quad i = 1, 2, \dots, l+u, \\ y_{l+3}(\omega \cdot \phi(x_{l+3}) + b) > 0. \end{array} \right. \right\} \quad (19)$$

If $y_{l+2}f(x_{l+2}) > y_{l+3}f(x_{l+3})$, then $y_{l+3}(\omega \cdot \phi(x_{l+3}) + b) > 0$. Further, $y_{l+2}(\omega \cdot \phi(x_{l+2}) + b) > 0$.

In summary, $\text{Version}_{l+3}^{\text{new}} \subset \text{Version}_{l+2}^{\text{new}}$.

Therefore, $\text{Area}(\text{Version}_{l+2}^{\text{new}}) > \text{Area}(\text{Version}_{l+3}^{\text{new}})$.

From Proposition 1, it can be found that, given the sample (x_i, y_i) , if the value of $y_i(x_i)$ is smaller, the labeled sample corresponding to the hyperplane dividing the current version space will reserve the smaller portion. Thus, the sample is more valuable for training the classification model.

The description of the new TSVM algorithm based on AL is shown in Algorithm 1.

In the TSVM-(AL + Graph) algorithm, $y_{\text{previous}}f(x)$ is used instead of $yf(x)$ to measure the sample information, where y_{previous} is not the true label of the sample but is the class label of the previous labeled adjacent sample. The advantage of this approximation strategy is that it does not depend directly on the current classification model; it is based on the relative relationship that similar classification results often have similar class labels. During the procedure, training is restarted only after a certain number of unlabeled samples are annotated. Compared with the traditional method of retraining once for each sample, this method can significantly improve the calculation efficiency.

Furthermore, Algorithm 2 can be obtained by applying the TSVM-(AL + Graph) algorithm to the rating/label prediction, which is shown in Algorithm 2.

5. Experimental Results and Analysis

In this section, firstly, verify the effectiveness of the proposed method based on the UCI dataset; then, apply the verified algorithm to personalized recommendation.

5.1. Experimental Datasets

5.1.1. UCI Dataset. Three datasets (that is, Breast_cancer, WPBC, and Bupa liver) from the UCI machine learning dataset are used to test our proposed algorithm. These datasets have been used in many studies, and they are a binary classification problem. For each dataset, randomly select a number of the data as labeled samples and put them into the labeled sample set L ; remove the labels of all the remaining samples and put them into the unlabeled sample set U . In this way, different sample selection strategies can be used to label unlabeled samples.

5.1.2. MovieLens Dataset. The MovieLens 1M dataset was collected by the University of Minnesota GroupLens research group through MovieLens and contains the anonymous ratings of 3900 movies by 6040 users. In order to facilitate modeling, the movie recommendation problem is converted into two classification problems, that is, “like” and “dislike”; the corresponding class labels are +1 and -1. At the same time, the rating values of 4 and 5 are labeled as +1, and the rating values of 1-3 are labeled as -1. In the experiment, we select 2000 users’ rating data as the experimental data and randomly select 200 data as the test samples; the rest of the data are used as the training set to obtain the classification model. Meanwhile, 5-fold cross-validation was used, and the average of the 5 groups’ data was taken as the experimental result.

5.1.3. Book Dataset. For the personalized book recommendation evaluation, we develop the crawler program and obtain the needed data of book purchase records from jd.com, a well-known e-commerce website in China. The crawled data include user name, user ID, book name, price, purchase time, and user reviews. In this experiment, we also mine review information, selecting and processing real and valuable review information as users’ purchased features are added to training samples. Finally, we train the recommendation model over the adjusted training set for evaluation.

When processing review information, we remove redundant punctuation and pause words and delete those with less than 2 characters, and we manually annotate 5 correct and valuable reviews and 5 spam reviews. For a review to be considered valuable or nonspam, it must meet the following conditions: the review contains a statement; the review expresses some opinions about the book or the characteristics of the book. Six characteristic features and corresponding descriptions of the review information are shown in Table 1.

5.2. Experimental Results on UCI Datasets

5.2.1. Introduction of Comparison Methods. We compare the proposed algorithm TSVM-(AL + Graph) against TSVM-(Random), TSVM, SVM-(AL) [34, 35], and SVM [36]. TSVM-(AL + Graph) algorithm not only exploits the manifold structure of the data to improve the performance

Input:

LDA, UDA /* Labeled sample set, unlabeled sample set
 k /* The number of samples in each round of interaction required labeled

Output:

$f(x)$ /* Classification function

Procedure:

Step 1: set parameters C_1 and C_2 . Select some samples from UDA , annotate them (both positive and negative samples are more than 1), and add them to LDA . Utilizing all labeled samples to establish an initial classification model with inductive learning.

Step 2: calculate values of the decision function for all unlabeled samples. On the basis of the increasing order of $f(x_i)$ values, an unlabeled sequence SDA is formed.

Step 3: select a sample x_i with the minimum objective function value for annotation, that is, $fvalue = \min_{x_i \in UDA} |f(x_i)|$, and record the corresponding label: $y_{previous} = y_i$.

Remove the x_i from UDA and SDA . At the same time, add x_i to LDA :

$UDA \leftarrow UDA - \{x_i\}$, $SDA \leftarrow SDA - \{x_i\}$, $LDA \leftarrow LDA + \{x_i\}$.

Step 4: while $i = 1, 2, \dots, k$

do

if $y_{previous} = 1$, then select the adjacent sample x_{i+p} in the opposite direction of SDA , and label it $y_{previous} = y_{i+p}$, where p can be either a positive or negative value.

if $y_{previous} = -1$, then select the adjacent sample x_{i+p} in the increasing direction of SDA , and label it $y_{previous} = y_{i+p}$, where p can be either a positive or negative value.

Delete the x_{i+p} from UDA and SDA . Simultaneously, add the x_{i+p} to LDA :

$UDA \leftarrow UDA - \{x_{i+p}\}$, $SDA \leftarrow SDA - \{x_{i+p}\}$, $LDA \leftarrow LDA + \{x_{i+p}\}$.

Step 5: retrain the TSVM over the L , and return $f(x)$. If there are still unlabeled examples, return to **Step 2**.

ALGORITHM 1: TSVM based on AL, that is, TSVM-(AL + Graph). The description of TSVM-(AL + Graph) algorithm.

Input:

User-item rating matrix $Recode$, item set $Item = \{i_1, i_2, \dots, i_M\}$, user set $User = \{u_1, u_2, \dots, u_N\}$, rating label $Label = \{c_1, c_2, \dots, c_K\}$.

Output:

The filled user-item rating matrix $Recode$.

Procedure:

Data preprocessing: extract unrated items from the training sample set, and randomly divide these unlabeled data into P datasets. For (Each unlabeled dataset in $DataUnlabel^{User}$)

Step 1: constructing “user-item” features: for unlabeled dataset d_1^{User} and labeled dataset $Data^A$, select m and n attributes from user attributes and item attributes, respectively, to form TSVM-(AL + Graph) features.

Step 2: constructing “user-item” behavior features: combining user preference vector $Perfecet_u = \{c_1, c_2, \dots, c_M\}$ and item attention $Attention_i = \{c_1, c_2, \dots, c_N\}$ vector to construct “user-item” behavior features.

Step 3: rating/label prediction: the features of the unlabeled dataset d_1^{User} and labeled dataset $Data^A$, the “user-item” behavior characteristics, and the labels are all used as the training set of the TSVM-(AL + Graph) algorithm. Then, the algorithm outputs the ratings/labels of the unlabeled data.

Step 4: extending the labeled dataset: after obtaining the ratings/labels of the unlabeled data in **Step 3**, add them to the labeled dataset.

ALGORITHM 2: Rating/label prediction algorithm based on TSVM-(AL + Graph). Rating/label prediction algorithm based on TSVM-(AL + Graph).

of the classifier but also exploits informative examples for human annotator. TSVM-(AL) is the TSVM-(AL + Graph) algorithm without the manifold regularization term. TSVM algorithm initially trains a classifier on both labeled examples and unlabeled examples, which exploits the cluster structure of examples and treats it as prior knowledge about the learning task. SVM algorithm only uses the labeled examples and performs well in the case of a sufficient number of labeled examples, but the performance will be degraded when the labeled examples are scarce.

5.2.2. *Classification Results and Analysis.* This experiment verifies the effectiveness of the proposed algorithm based on the UCI dataset and compares and analyzes the algorithm from different perspectives.

(1). *Set the size of the initial labeled sample set $L = 30$, and the batch sampling size $k = 1$.* The main purpose of this experiment is to comprehensively test the performance of the proposed algorithm, including active learning sampling strategy and random sampling strategy and the utilization of manifold structure before and after the introduction of the

TABLE 1: Extracted 6 features and their sample dictionary terms.

| Features | Sample dictionary terms | Dictionary size |
|-------------------|---|---|
| Opinion phrases | 指定用书(textbook), 经典(classic), 高性价比(cost-effective), 很/特别/非常喜欢(like very much), 精装版(hardcover), 内容详实(content full and accurate), 最新版(the latest version), 经典(classical), 学习必备(learning essential), 质量好(good quality), 垃圾(rubbish), ... | 400 |
| Question patterns | 出版社(where), 怎么样(how about), 作者/译者(who), 出版时间(when), 主要内容(what), ?, ... | 25 |
| Language | 汉语(Chinese), 英语(English), 日语(Japanese), 俄语(Russian), 法语(French), 德语(German), ... | 10 |
| Category of book | 电子书(e-book), 教材教辅(course books), 综合性图书(comprehensive books), 励志(inspirational), 人文社科(humanities and social sciences), 生活(life), 经济管理(economics and management), 医药/卫生(medicine and health), 工业技术(industrial technology), 少儿(children's book), ... | 15 basic categories and several subcategories |
| Length of review | The total number of characters in a review, excluding punctuation | - |
| Product rating | 五角星(five-pointed star) | 5 |

manifold regular term. The comparison results are shown in Figures 1 and 2.

- (1) The method of active learning is better than the method of nonactive learning, and the classification accuracy is higher. Traditional SVM has better performance than other classification models in the case of small samples, but it cannot make good use of the information implicit in a large number of unlabeled samples to improve the performance of the classifier. Figure 1 also illustrates this aspect. From Figures 1 and 2, it can be found that as the number of labeled samples increases, the classification performance of TSVM-(AL+Graph) is gradually improving. When a certain percentage is reached, the classification performance exceeds the traditional SVM. This also shows that active learning is effective and reasonable in sample selection strategy, which is of great help to improve the performance of the classifier.
- (2) It can be found from Figures 1 and 2 that, compared with the random sampling strategy, the samples selected by the active learning strategy are more likely to be “support vectors” and can ensure that the selected samples can further improve the performance of the classifier. At the same time, random sampling has a certain degree of randomness, which cannot guarantee that the larger the number of samples, the more obvious the improvement of classifier performance.
- (3) After introducing the manifold regularization term, the proposed method can make better use of the manifold structure of unlabeled samples. Comparing different datasets in Figures 1 and 2, it can be found that, for Brest_cancer dataset, the performance difference between TSVM-(AL+graph) and TSVM-(AL) is slightly smaller. For WPBC dataset, the performance of TSVM-(AL+graph) is slightly higher than that of TSVM-(AL). Therefore, the introduction of manifold regularization term is helpful to improve the classification performance of TSVM.

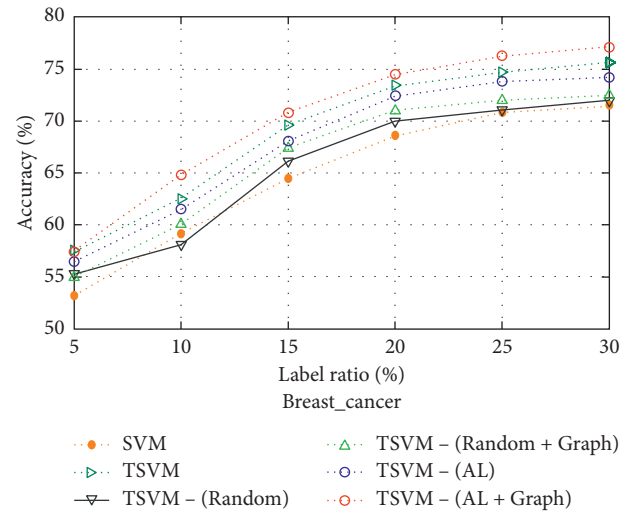


FIGURE 1: The classification accuracy on Breast_cancer dataset as the number of labeled training samples increases.

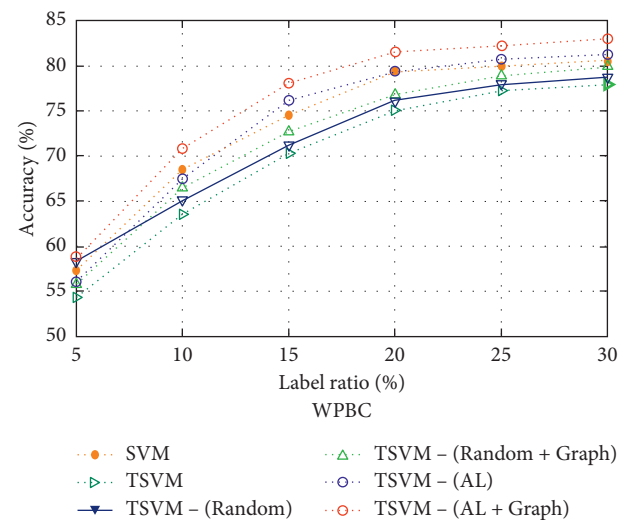


FIGURE 2: The classification accuracy on WPBC dataset as the number of labeled training samples increases.

(2). *The effect of different class label prediction methods on classifier performance.* This experiment compares TSVM-(OAL + Graph) and TSVM-(AL + Graph) and analyzes them on the Bupa liver dataset and WPBC dataset (set the initial labeled sample set $L = 10$), where the TSVM-(OAL + Graph) method adopts the predicted class label using the current classifier as the measure of active learning; the TSVM-(AL + Graph) method adopts the class label of previously labeled adjacent sample, which makes full use of cluster assumption among the data.

From the experimental results in Figures 3 and 4, it can be found that the predictive class labels of adjacent samples labeled previously have more advantages than those predicted by the current classifier in general.

For Bupa liver and WPBC datasets, there is no significant difference in classification performance between the two algorithms, which may be related to the distribution characteristics of the datasets. For Bupa liver dataset, when the proportion of labeled samples is 10%, and when the proportion of labeled samples in the WPBC dataset is 20%, the classification performance of TSVM-(OAL + Graph) exceeds TSVM-(AL + Graph). However, with the increase of labeled sample proportion, the classification performance of TSVM-(AL + Graph) is gradually improved and exceeds TSVM-(OAL + Graph).

5.3. Personalized Recommendation Experiment Results and Analysis. In this section, we implement two personalized recommendations based on real datasets. The first one is a personalized movie recommendation based on MovieLens dataset. In this experiment, the demographic information, user behavior (rating about the movie), and the content of the movie are processed to form a “user-movie” association matrix, and then the model is trained. The result of movies classification is output as the recommendation. According to the classification results, a recommendation list is provided for users instead of the similarity calculation in the traditional collaborative filtering recommendation method. The second evaluation is a personalized book recommendation. This experiment is mainly to analyze the book purchase records, extract the corresponding features, and obtain the personalized recommendation model. In this evaluation, to improve the quality of the book recommendation, the review information that is valuable for improving the performance of the recommendation model is mined and processed to form the users’ purchasing characteristics, and it is then added to the training sample set to train the personalized book recommendation model.

5.3.1. Results and Analysis of Personalized Movie Recommendation. Figure 5 shows the offline experimental results of the recommendation method based on TSVM-(AL + Graph) and the methods in [37], including CF-IWA, PSO-SVM, ItemCF, UserCF, PSO-SVM, GA-SVM, GS-SVM, TSVM, and BP neural network on the MovieLens dataset.

From Figure 5, it can be seen that the classification accuracy of various methods grows with the number of training samples increasing, because with the number of

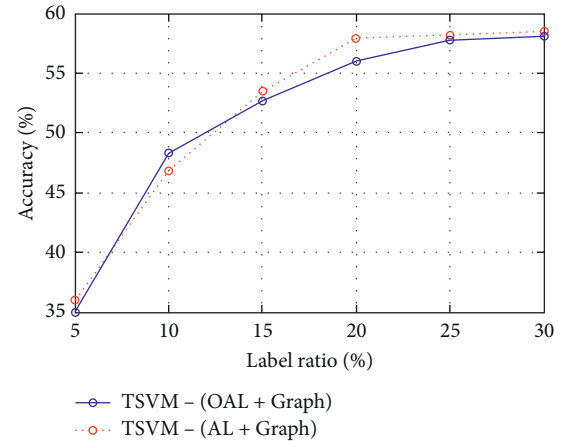


FIGURE 3: The classification accuracy on Bupa liver dataset.

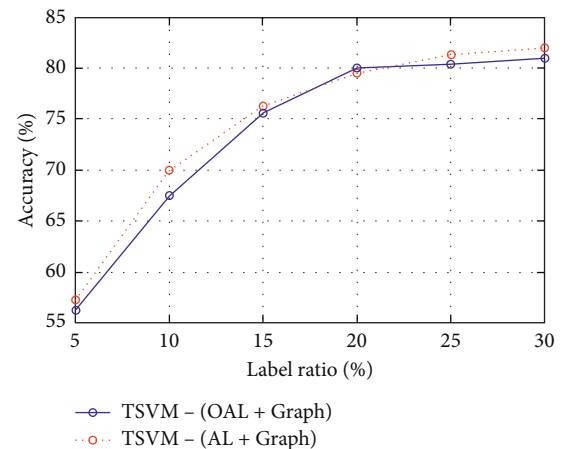


FIGURE 4: The classification accuracy on WPBC dataset.

labeled samples increasing more information can be used for movie recommendation, which provides rich and reliable information for establishing recommendation model. When the labeled training samples account for 20% of the entire training sample set, the performance of the recommendation model based on TSVM-(AL), TSVM-(AL + Graph), and TSVM is not as good as the recommendation model based on traditional SVM, which may be related to the quality and quantity of labeled samples. However, with the number of labeled samples increasing, the advantages of these three methods become more obvious, and the proposed method in this paper is proved to be better than the other six methods. Meanwhile, it can also be found that the model-based method is better than the ItemCF and UserCF methods, which may be caused by the “data sparsity” and “cold start” problems; and the model-based method can make good use of the users’ demographic information, behavior information, and label information, etc., which is useful to alleviate the “data sparsity” and “cold start” problems.

A good recommendation model not only has high accuracy but also has the ability to identify as many items as possible that are interesting to the users (“recall”). An

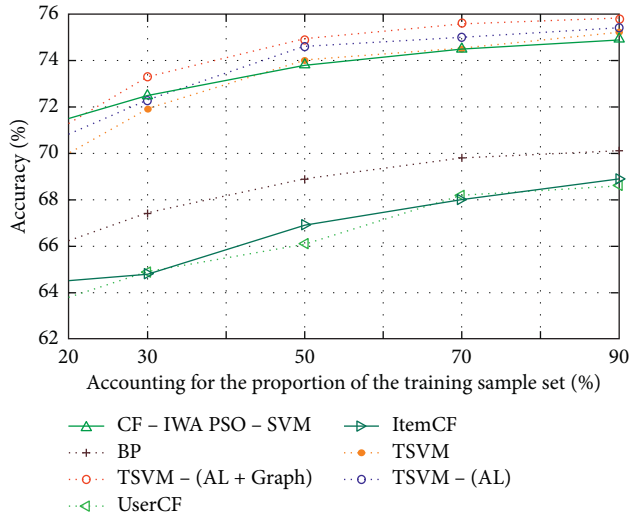


FIGURE 5: Classification accuracy of the seven methods.

important measure of this ability is the F -Score. As shown in Figure 6, the TSVM-(AL + Graph) method has a better F -Score than the other six methods. In particular, when the proportion of training samples accounts for 90% of the entire training set, the F -Score value reaches the highest.

5.3.2. Results and Analysis of Personalized Book Recommendation

(1) *Personalized Recommendation Model Based on TSVM-(AL + Graph) algorithm.* The procedure of the personalized recommendation based on TSVM-(AL + Graph) algorithm is shown in Figure 7.

(2) *User Review Information Preprocessing.* In order to fully explore the advantages of the proposed method, a series of preprocessing of the dataset is conducted. At first, “lexical perspective” preprocessing is conducted. The Chinese word segmentation tool named ICTCLAS is used to segment and annotate the reviews, and then, each review is transformed into a term frequency inverse document frequency (TF-IDF). After that, the “statistical perspective” analysis such as calculating the proportion of keywords and key patterns contained in each review is conducted to form a quantitative matrix.

During the processing, there are six features that can be quantified as follows:

- (1) The proportion of opinion phrases in a review sentence.
- (2) The proportion of question patterns in a review sentence.
- (3) The proportion of language in a review sentence.
- (4) The proportion of book categories mentioned in a review sentence.
- (5) The length of a review sentence.
- (6) The number of “five-pointed stars” given by the user for the overall review of the book (at most 5).

(3) Personalized Book Recommendation Results and Analysis

(1) Mining of book evaluation information.

In order to evaluate the impact of unlabeled samples in different algorithms, the classification performances of TSVM-(AL + Graph), SVM, TSVM, and TSVM-(AL) were compared and analyzed by changing the proportion of labeled samples in the experiment. In this experiment, the range of the proportion of labeled samples is within [2%, 20%], and the total number of unlabeled samples is 1000. Figures 8 and 9 show the classification accuracy caused by the range of the proportion of labeled samples.

As can be seen from Figures 8 and 9, the classification accuracy of TSVM-(AL) and TSVM-(AL + Graph) grows with the proportion of labeled samples increasing. More detailed observation shows that when the number of labeled samples is very few, the classification performance of TSVM-(AL + Graph) is significantly better than SVM and TSVM. For example, in Figure 8, when the proportion of labeled samples is 8.0%, the classification accuracy of TSVM-(AL + Graph) is improved, about 10% better than SVM and 5.0% higher than TSVM. In Figure 9, the classification accuracy of TSVM-(AL + Graph) is about 9.0% higher than SVM and 3.0% higher than TSVM.

Figures 8 and 9 mine the book review information from different perspectives, and the classification accuracy of the “lexical perspective” is slightly higher than “statistical perspective,” which may be related to the inherent statistical characteristics of the model. It is obvious that the classification performance of the method proposed in this paper is better than both SVM and TSVM in these two different perspectives, and the superiority of the proposed method has been sufficiently verified.

(2) Personalized book recommendation method based on user review mining.

In this evaluation, the mined users’ valuable review information is added to the original dataset as the users’ interest feature to form a new dataset, and 70% of data is used for training while the other 30% of data is used for testing; the newly formed “user-book” association data is randomly divided into two parts.

This experiment mainly includes two evaluations. The first one is to compare and analyze the impact of the review information for book recommendation. The other one is to compare the proposed method with other representative methods.

① The effect of the review information.

In order to evaluate the effect of review information during book recommendation, there are four algorithms used for estimation; they are

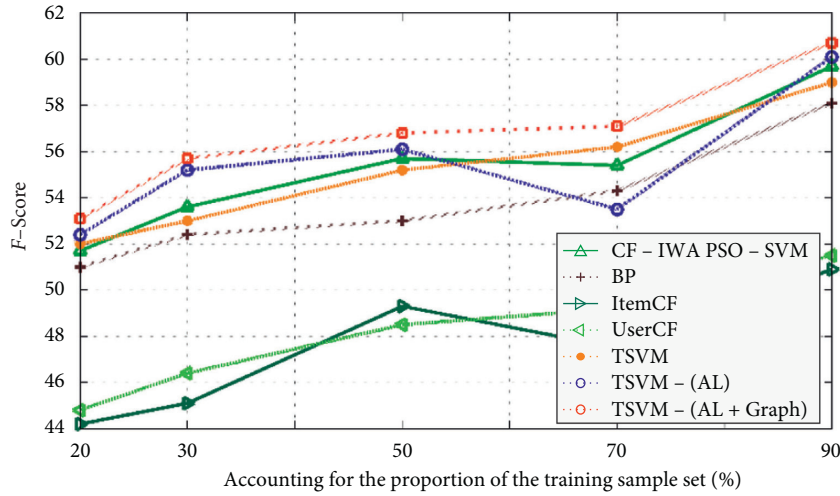


FIGURE 6: F-Score values of the seven methods.

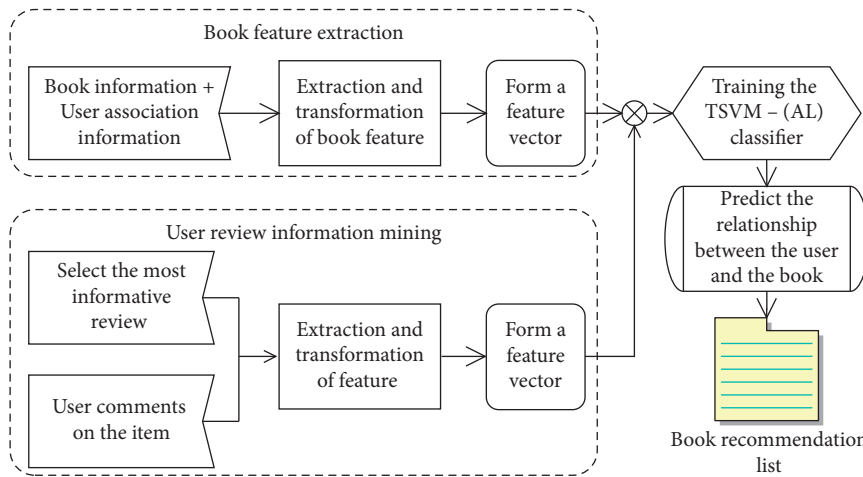


FIGURE 7: The personalized recommendation model based on TSVM-(AL+Graph) algorithm.

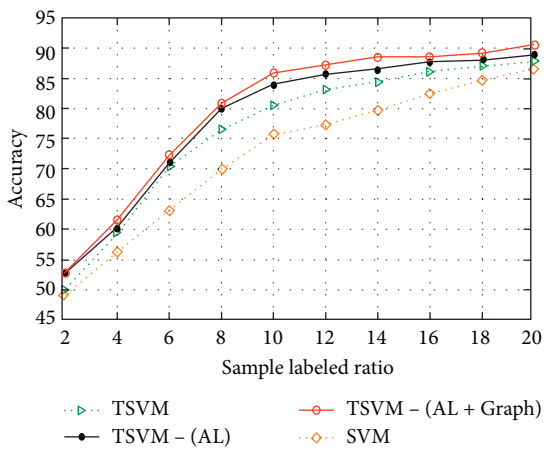


FIGURE 8: Classification accuracy corresponding to “lexical perspective.”

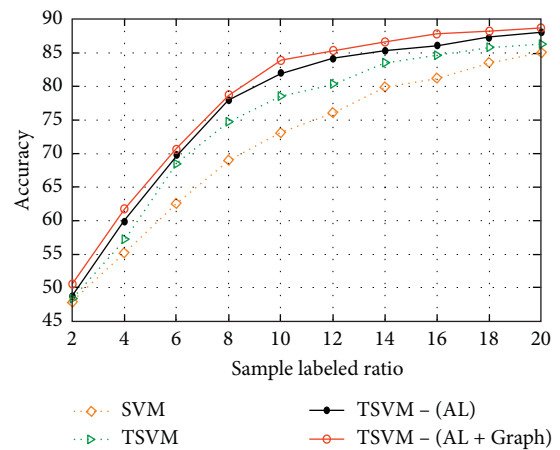


FIGURE 9: Classification accuracy corresponding to “statistical perspective.”

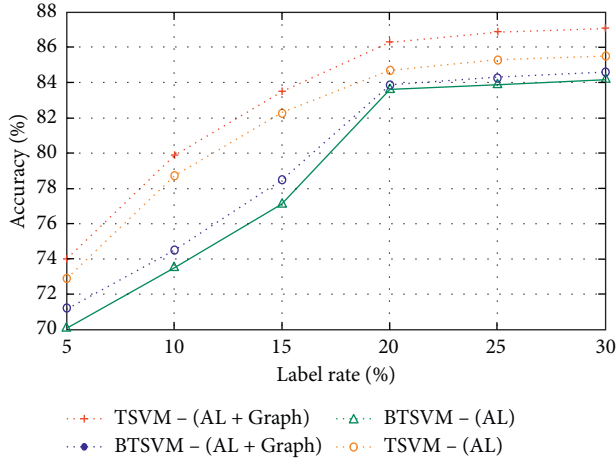


FIGURE 10: The effect of the reviews.

TSVM-(AL + Graph), TSVM-(AL), BTSVM-(AL + Graph), and SVM-(AL). The last two algorithms represent the recommendation methods without utilizing the review information, while TSVM-(AL + Graph) and TSVM-(AL) represent the recommendation methods employing the review information. Figure 10 presents the effectiveness of the four algorithms.

From Figure 10, it can be found that the reviews have a positive impact to improve the recommendation accuracy. It shows that the method of mining the book review proposed in this paper is helpful to improve the recommendation accuracy and is useful to discover users' interests and preferences. Therefore, it is reasonable to integrate users' valuable review information into the training model.

② Performance analysis of different book recommendation methods.

Figure 11 shows the recommendation accuracy of six methods under different labeled sample proportion. The results show that TSVM-(AL + Graph) and TSVM-(AL) methods have relatively high accuracy. This superiority mainly benefits from the following: firstly, the method based on active learning can effectively utilize the small number of labeled samples to explore the unlabeled data. Since the active learning approach will iteratively label those samples that have significant impact on the improvement of the classifier in each iteration and extend them into the training set, the data sparsity and cold start problem will be heavily reduced. Secondly, the manifold regularization term is also employed to discover and utilize the hidden geometric information of the data, which is helpful to reduce the solution space of the nonconvex hat shape; thus, the performance of the classifier is further improved.

The above experimental results show that the proposed method is superior to other approaches in mining users' interests and preferences. The effectiveness of the proposed method in personalized book recommendation scenarios is verified. The experimental data proved that the proposed method can supplement those aspects of interest with sparse

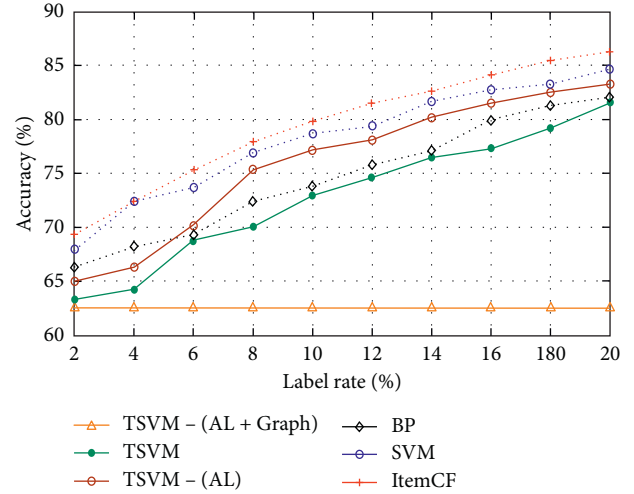


FIGURE 11: The effect of different labeled sample ratios on the recommendation after utilizing the reviews.

data, helping the interest model to be comprehensive, and achieve a better performance of the recommendation system.

6. Conclusion

This paper explores the “data sparsity” (data with labels scarcity), “cold start,” and neglecting demographic characteristics of the traditional collaborative filtering recommendation algorithm in real applications and proposes a new recommendation method based on TSVM and active learning. Firstly, to solve the challenge of using unlabeled data, an active learning strategy is proposed to query and label unlabeled samples. Meanwhile, the manifold regularization term is integrated into the objective function to utilize the distribution characteristics of samples. Secondly, during the construction of the recommendation model, user behavior characteristics, item information, and demographic information are employed to help label and fill the samples to solve the data sparsity and cold start problem. Finally, the proposed algorithm is applied to UCI dataset, personalized movie recommendation, and book recommendation to verify its effectiveness. The experimental results show that the proposed method can effectively mine users' potential preferences, significantly reduce the heavy cost of sample labeling, and lead to a performance improvement of the recommendation system.

Data Availability

The data used to support the findings of the personalized movie recommendation are available from <https://grouplens.org/datasets/movielens/>, and the original data of the precise book review records cannot be released in order to preserve the privacy of individuals.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was partially supported by the Technology Foundation of Guizhou Province (grant no. QianKeHeJiChu [2020]1Y269), New Academic Seedling Cultivation and Exploration Innovation Project (grant no. QianKeHe Platform Talents[2017]5789–21), Program for Innovative Talent of Guizhou Province (grant no. QianCaiJiao [2018]190), National Natural Science Foundation of China (grant nos. 71901078 and 71964009), High-Level Talent Project of Guizhou Institute of Technology (grant no. XJGC20190929), and Special Key Laboratory of Artificial Intelligence and Intelligent Control of Guizhou Province (grant no. KY [2020]001).

References

- [1] Z. Zheng, H. Ma, M. R. Lyu et al., “QoS-aware web service recommendation by collaborative filtering,” *IEEE Transactions on Services Computing*, vol. 4, no. 2, pp. 140–152, 2011.
- [2] H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu, “A new user similarity model to improve the accuracy of collaborative filtering,” *Knowledge-Based Systems*, vol. 56, pp. 156–166, 2014.
- [3] B. Settles, “Active learning literature survey,” Technical Report 1648, University of Wisconsin-Madison, Madison, WS, USA, 2009.
- [4] J. S. Breese, D. Heckerman, and C. Kadie, “Empirical analysis of predictive algorithms for collaborative filtering,” *Uncertainty in Artificial Intelligence*, vol. 98, no. 7, pp. 43–52, 2013.
- [5] M. Deshpande and G. Karypis, “Item-based top- N recommendation algorithms,” *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 143–177, 2004.
- [6] J.-H. Su, H.-H. Yeh, P. S. Yu, and V. S. Tseng, “Music recommendation using content and context information mining,” *IEEE Intelligent Systems*, vol. 25, no. 1, pp. 16–26, 2010.
- [7] K. Wang and Y. Tan, “A new collaborative filtering recommendation approach based on naive bayesian method,” in *Proceedings of the Advances in Swarm Intelligence-Second International Conference*, Chongqing, China, June 2011.
- [8] L. Ren and W. Wang, “An SVM-based collaborative filtering approach for Top-N web services recommendation,” *Future Generation Computer Systems*, vol. 78, no. 2, pp. 531–543, 2018.
- [9] J. He, F. Zhuang, Y. Liu, Q. He, and F. Lin, “Bayesian dual neural networks for recommendation,” *Frontiers of Computer Science*, vol. 13, no. 6, pp. 1255–1265, 2019.
- [10] L. Na, L. Ying, T. Xiao-Jun et al., “Improved collaborative filtering algorithm using topic model,” in *Proceedings of the International Conference on Parallel & Distributed Computing*, Taipei, Taiwan, 2017.
- [11] R. Devooght and H. Bersini, *Collaborative filtering with recurrent neural networks*, <https://arxiv.org/abs/1608.07400>, 2016.
- [12] D. G. Bridge and F. Ricci, “Supporting product selection with query editing recommendations,” in *Proceedings of the ACM Conference on Recommender Systems*, Minneapolis, MN, USA, 2007.
- [13] R. Karimi, W. Martin, A. Nanopoulos et al., “Factorized decision trees for active learning in recommender systems,” in *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence*, Herndon, VA, USA, 2013.
- [14] M. Elahi, F. Ricci, and N. Rubens, “Active learning strategies for rating elicitation in collaborative filtering,” *ACM Transactions on Intelligent Systems and Technology*, vol. 5, no. 1, pp. 1–33, 2013.
- [15] M. Elahi, M. Braunhofer, F. Ricci et al., “Personality-based active learning for collaborative filtering recommender systems,” in *Proceedings of the Congress of the Italian Association for Artificial Intelligence*, Turin, Italy, 2013.
- [16] W. Shi, X. Liu, and Q. Yu, “Correlation-aware multi-label active learning for web service tag recommendation,” in *Proceedings of the 2017 IEEE International Conference on Web Services (ICWS)*, Honolulu, Hawaii, 2017.
- [17] X. Liu, S. Agarwal, C. Ding et al., “An LDA-SVM active learning framework for web service classification,” in *Proceedings of the International Conference on Web Services*, pp. 49–56, San Francisco, CA, USA, 2016.
- [18] G. Li, Z. Zhang, L. Wang, Q. Chen, and J. Pan, “One-class collaborative filtering based on rating prediction and ranking prediction,” *Knowledge-Based Systems*, vol. 124, pp. 46–54, 2017.
- [19] L. Wang, H. Zhang, and Y. Chen, “Collaborative filtering recommendation algorithm based on Dice-Euclidean similarity function,” *Application Research of Computers*, no. 10, pp. 2891–2895, 2015.
- [20] U. Liji, Y. Chai, and J. Chen, “Improved personalized recommendation based on user attributes clustering and score matrix filling,” *Computer Standards & Interfaces*, vol. 57, pp. 59–67, 2018.
- [21] M. K. Najafabadi, A. Mohamed, and C. W. Onn, “An impact of time and item influencer in collaborative filtering recommendations using graph-based model,” *Information Processing & Management*, vol. 56, no. 3, pp. 526–540, 2019.
- [22] L. Huang, W. Tan, and Y. Sun, “Collaborative recommendation algorithm based on probabilistic matrix factorization in probabilistic latent semantic analysis,” *Multimedia Tools and Applications*, vol. 78, no. 7, pp. 8711–8722, 2019.
- [23] Z. Gao, Y. Fan, C. Wu et al., “SeCo-LDA: mining service Co-occurrence topics for composition recommendation,” *IEEE Transactions on Services Computing*, vol. 12, no. 3, pp. 446–459, 2019.
- [24] H. Shin, S. Kim, J. Shin, and X. Xiao, “Privacy enhanced matrix factorization for recommendation with local differential privacy,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 9, pp. 1770–1782, 2018.
- [25] I. Nouioua, N. Amardjia, and S. Belilita, “A novel blind and robust video watermarking technique in fast motion frames based on SVD and MR-SVD,” *Security and Communication Networks*, vol. 2018, no. 10, pp. 1–17, 2018.
- [26] X. Liu, C. Aggarwal, Y. F. Li et al., “Kernelized matrix factorization for collaborative filtering,” in *Proceedings of the SIAM International Conference on Data Mining*, pp. 378–386, Alberta, Canada, 2016.
- [27] B. Shams and S. Haratizadeh, “Graph-based collaborative ranking,” *Expert Systems with Applications*, vol. 67, pp. 59–70, 2017.
- [28] Y. Chen, G. Wang, and S. Dong, “Learning with progressive transductive support vector machine,” in *Proceedings of the IEEE International Conference on Data Mining*, pp. 67–74, Maebashi, Japan, 2002.
- [29] M. Elahi, F. Ricci, N. Rubens et al., “A survey of active learning in collaborative filtering recommender systems,” *Computer Science Review*, vol. 20, pp. 29–50, 2016.
- [30] Z. Wang, S. Yan, and C. Zhang, “Active learning with adaptive regularization,” *Pattern Recognition*, vol. 44, no. 10-11, pp. 2375–2383, 2011.

- [31] C.-L. Li, C.-S. Ferng, and H.-T. Lin, "Active learning using hint information," *Neural Computation*, vol. 27, no. 8, pp. 1738–1765, 2015.
- [32] K. Moshkbar-Bakhshayesh and M. B. Ghofrani, "Combining supervised and semi-supervised learning in the design of a new identifier for NPPs transients," *IEEE Transactions on Nuclear Science*, vol. 63, no. 3, pp. 1882–1888, 2016.
- [33] R. Collobert, F. Sinz, J. Weston et al., "Large scale transductive SVMs," *Journal of Machine Learning Research*, vol. 7, pp. 1687–1712, 2006.
- [34] C. Campbell, N. Cristianini, and A. J. Smola, "Query learning with large margin classifiers," in *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 111–118, Kunnamangalam, India, 2000.
- [35] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," in *Proceedings of the Ninth ACM International Conference on Multimedia*, pp. 107–118, Kyongju, Korea, 2001.
- [36] C. Chang and C. Lin, "LIBSVM: a library for support vector machines, science," 2001, <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [37] X. Wang, F. Luo, C. Sang, J. Zeng, and S. Hirokawa, "Personalized movie recommendation system based on support vector machine and improved particle swarm optimization," *IEICE Transactions on Information and Systems*, vol. E100-D, no. 2, pp. 285–293, 2017.