applied sciences

MDPI

Article

Diversity Balancing for Two-Stage Collaborative Filtering in Recommender Systems

Liang Zhang [1,*], Quanshen Wei [1], Lei Zhang [1], Baojiao Wang [1] and Wen-Hsien Ho [2,3,*]

[1] Business School, Qingdao University, Qingdao 266071, Shandong, China; weiquanshen2008@126.com (Q.W.); qduzhanglei@qdu.edu.cn (L.Z.); alinazl9507@163.com (B.W.)

[2] Department of Healthcare Administration and Medical Informatics, Kaohsiung Medical University, Kaohsiung 807, Taiwan

[3] Department of Medical Research, Kaohsiung Medical University Hospital, Kaohsiung 807, Taiwan

* Correspondence: zhangliang@qdu.edu.cn (L.Z.); whho@kmu.edu.tw (W.-H.H.)

Received: 17 December 2019; Accepted: 24 January 2020; Published: 13 February 2020

check for updates

Abstract: Conventional recommender systems are designed to achieve high prediction accuracy by recommending items expected to be the most relevant and interesting to users. Therefore, they tend to recommend only the most popular items. Studies agree that diversity of recommendations is as important as accuracy because it improves the customer experience by reducing monotony. However, increasing diversity reduces accuracy. Thus, a recommendation algorithm is needed to recommend less popular items while maintaining acceptable accuracy. This work proposes a two-stage collaborative filtering optimization mechanism that obtains a complete and diversified item list. The first stage of the model incorporates multiple interests to optimize neighbor selection. In addition to using conventional collaborative filtering to predict ratings by exploiting available ratings, the proposed model further considers the social relationships of the user. A novel ranking strategy is then used to rearrange the list of top-N items while maintaining accuracy by (1) rearranging the area controlled by the threshold and by (2) maximizing popularity while maintaining an acceptable reduction in accuracy. An extensive experimental evaluation performed in a real-world dataset confirmed that, for a given loss of accuracy, the proposed model achieves higher diversity compared to conventional approaches.

Keywords: recommender systems; collaborative filtering; diversity; multi-interest; ranking function

1. Introduction

The advent of the Internet has exacerbated the information overload problem, which increasingly complicates the retrieval of useful information from the Internet. A recommender system (RS) [1–3] is a powerful tool for solving the information overload problem in many online applications, and the use of RS has steadily increased in recent years. Typically, an RS is designed to recommend proper items to proper users based on their ratings or on their online behavior such as purchasing behavior [2]. Current RS methods can be classified as content-based filtering methods [4,5] and collaborative filtering methods [6–8]. Content-based RSs generate recommendations after analyzing item descriptions and user interest profiles, whereas collaborative filtering RSs recommend items that are of particular interest to the active user by identifying similarities between users or between items. Collaborative filtering is among the most effective and widely used RS methods in many domains due to its easy implementation, e.g., it does not require specific domain knowledge.

The major task of collaborative filtering is to compute the distances between users or between items based on how users have interacted with the system in the past [2,3]. Since users or new users rarely interact extensively in the system, calculating the precise distances and similarities between users or items is difficult however because user ratings and user feedback are limited. This problem is known as the "cold start" problem, or the "data sparsity" problem [8–11]. Most collaborative filtering RSs substantially reduce performance in a cold start scenario.

A common solution for the cold start problem is to consider additional information sources. For example, recommendation performance can be improved by considering information about the social relations of users, such as their social network friendships [12], or by considering the context of information [13]. For example, in an E-commerce scenario, the purchasing decisions made by users are usually affected by the opinions of their friends. Proposed methods of using information about social relationships to improve the prediction accuracy of an RS and to alleviate the cold start problem include SoRec [14], RSTE [15], SoReg [16], and TrustSVD [17]. The objective of these methods is to identify users who are representative of the actual user because of their implicit social relationships with the actual user.

Previous studies of RS design [18–23] have focused on using rating information to improve prediction accuracy. However, this approach can limit recommendation results and introduce the long tail recommendation problem [24,25]. For example, if a movie RS is initially used to get recommendations for comedy movies, the RS may tend to continue recommending the same genre of movies in the future. Thus, the user may soon begin to disregard the recommendations. A recommendation list that has high diversity, markedly increases the interest of users [26]. Therefore, an effective RS requires not only high accuracy, but also high diversity. To enhance the diversity of a recommendation list, researchers have attempted to develop methods of maximizing the range of interests considered by the RS [27]. Since increasing diversity reduces prediction accuracy, the optimal trade-off [28,29] between diversity and accuracy must be determined.

To address the above problems, this study proposes a novel diversity optimization method: Diversity Balancing for Two-Stage collaborative filtering (DBTS). In the first stage of the filtering process, DBTS uses both the social relationships of the user and user-based recommendations to predict missing ratings. In the second stage, the rating recommendation list is generated by leveraging user interests and social influences. The main contributions of this paper are the two novel characteristics of the proposed DBTS: (1) a two-stage optimization strategy for balancing accuracy and diversity and (2) the use of social relationship filtering combined with collaborative filtering to enhance the robustness of the RS. Extensive experiments performed in a real-world dataset confirmed the good performance of the algorithm in terms of diversity and accuracy.

The rest of this paper is organized as follows. The next section provides an overview of existing collaborative filtering methods and diversity enhancing methods. Section 3 presents the details of the proposed approach and the design of the algorithm. Section 4 describes the set-up used in the experiments, the evaluation metrics, and the experimental results. The final section concludes the whole paper.

2. Related Works

2.1. Social-Based Recommender Systems

Although large numbers of users and items generally participate in an RS, user feedback on items is usually limited. A simple and effective way to improve RS accuracy is to consider information sources such as demographic characteristics [29] or information about friends of users in social networks [30–33]. Before making decisions, real-world users tend to consider the opinions and suggestions of friends they trust. Hence, social relationships are a possible source of information that can be used to enhance RS performance. [14–16,18,19].

A social media based RS solves the cold-start and sparsity problems by rating the strengths of social relationships and then considering recommendations from friends who have the strongest relationships with the user. Methods of evaluating social information can be categorized as implicit trust-based methods and explicit trust-based methods [6]. Explicit trust, such as a list of trusted users on the Epinions website or the "friends list" on Facebook, are demonstrated by the user explicitly. Explicit trust values are values that are explicitly stated by the user. Implicit values are those that are not explicitly stated but are inferred on other evidence, such as user profile.

Various methods of exploiting implicit trust information in Collaborative Filtering (CF) systems have already proven effective. For example, the approach described in [30] was based on both user-based and item-based trust information. Implicit trust and global reputation were computed and then used to alleviate the data sparsity problem. Another approach is to use the framework described in [20] to identify implicit trust statements based on the Pareto dominance relations of users. In this recommendations approach, non-prominent users, i.e., those whose opinions are not effective, are excluded from consideration. Based on the small-world property of user communities, ref. [31] presented an algorithm that can derive implicit trust values from user ratings. Implicit trust-based models infer global trust or reputation across the entire user community. Therefore, implicit trust-based models often give inaccurate recommendations when malicious or fraudulent user behaviors are encountered. Thus, explicit trust-based models have attracted interest in recent research.

The probabilistic matrix factorization method is proposed in [14], which combines social network with a user-item rating matrix. This method shares the latent feature space of the user to make social recommendations. A user decision is considered a balance between the personal preference of the user and the preferences of trusted friends. Ref. [15] proposed a social trust ensemble method (RSTE) to linearly combine the preferences of the user and the preferences of trusted friends. Another approach used in [32] was to explicitly merge the performance of trusted neighbors and the active user by averaging their ratings, which eliminated the data sparsity problem. In [33], an ant colony optimization algorithm was used in a trust graph to identify a set of users that are like the active user. In [19], a combination of similarity values and trust statements was used to construct a trust network for the active user. A novel reliability measure was used to evaluate the prediction accuracy of the system. Ref. [34] presented a hybrid approach in which both user preferences and social trust are used to make appropriate recommendations. Rating prediction accuracy was improved by considering untrusted relationships as well as propagation effects among users. The authors of [35] proposed an unsupervised learning approach to applying explicit trust in a global reputation model. Their experiments showed that global reputation becomes significant only when the scale of the ego-network of users is small. The level of trust between an active user and the trustees may change over time. To address the time-variant characteristics of trust, [36] developed a dynamic analysis in which a pheromone updating strategy based on ant colony optimization is used to analyze trust. In [37], the authors compared the advantages and disadvantages of implicit trust and explicit trust and concluded that explicit trust-based methods usually outperform implicit trust-based methods.

2.2. Improving RS Diversity

To improve the user experience in RSs, researchers have attempted to improve the novelty and diversity of suggested items and to improve the visibility of long tail items [24,26,27]. Attempts to improve RS accuracy have focused on finding items highly similar to the past preferences of active users while attempts to improve diversity have focused on evaluating "dissimilarity" among recommended items. Methods of improving accuracy are often incompatible with those for improving diversity. Therefore, the trade-off between accuracy and diversity must be optimized. A major challenge of designing an effective RS is to determine the maximum diversity that can be achieved for an acceptable loss in prediction accuracy.

Proposed models for improving RS diversity can be categorized as two-stage models and unified models. Unified models [24–28] have a single stage optimization framework for solving the joint formulation of a weighted combination of diversity and accuracy. For example, the DiABlO single stage optimization solution proposed in [26] was based on latent factor models. A diversity enhancement regularization term was then used to improve recommendation diversity by constraining the matrix factor decomposition model. In [27], the authors presented a unified framework and a unified convex formulation based on available ratings and item metadata. The split Bregman technique used to solve this formulation achieved an acceptable accuracy–diversity trade-off. In [28], a heat conduction algorithm was proposed for solving the triple dilemma of simultaneously optimizing stability, accuracy, and diversity by ranking the stability of similarities and then considering only the most stable ones.

The recommendation performance of unified models mainly depends on the performance of the optimization algorithms. In many cases, falling into local optimal solutions is more likely than falling into global optimal solutions. However, the first stage of a two-stage recommendation strategy [38–40] focuses on predicting the missing ratings and on using the collaborative filtering method to generate a candidate recommendation list. The second stage applies a certain ranking strategy to maximize diversity of the list. The ranking strategies often require a pre-defined heuristic threshold. Items rated above the threshold are added to the recommendation list. Unlike the item list in a conventional RS, recommended items appear in a descending order of predicted ratings. To improve diversity, various strategies have been proposed for ranking the predicted ratings in the second stage. The RS design described in [41] considered the popularity of items in addition to their relevancy. A multi-objective evolutionary algorithm for simultaneously optimizing accuracy and diversity was then proposed. In [42], clustering technique was used to segregate items. Items were then selected from each cluster to ensure a high diversity of the recommendation list. Ranking strategies developed in other works, e.g., [38,39], have used greedy optimization, item popularity to promote diversity.

The works discussed above either use trust relationship to alleviate the problem of cold start and sparsity so as to improve recommendation accuracy or propose new recommendation list ranking strategies to improve recommendation diversity. However, they neglect the impact of interest overlapping between the target users and trust users on both accuracy and diversity of recommendations. Trust relationships and common interests can connect two users who do not know each other, and they are equally important. Users with the same trust level may have different interests. We consider both multiple interests and trust relationships of users, so diversity and accuracy are reflected in the final recommendation list. The proposed method increases both personalized diversity and aggregate diversity with minimal loss of accuracy.

3. Diversity Balancing for Two-Stage Collaborative Filtering

This section describes the details of the main stages of the proposed DBTS. The aim of the method is to recommend items of interest to users by hybridizing available rating values and social trust information to increase recommendations of long-tail items. The DBTS is a two-stage collaborative filtering algorithm. Users' social information has been used to alleviate the user cold start problem. At the first stage of DBTS, there is a trust propagation strategy to generate trusted users for new users based on users' trust data. The trust-based similarities between users are calculated by Equation (1). Users are then divided into high-trust ones and low-trust ones according to a predefined trust threshold. The similarities of trusted users and the interest distributions of the active user are then used to build a multi-interest collaborative filtering model, which predicts ratings and generates a top-N candidate item list for the active user. The second stage uses a re-rank strategy to adjust the recommendation list based on item popularity and diversity. Figure 1 is a flowchart of the DBTS.

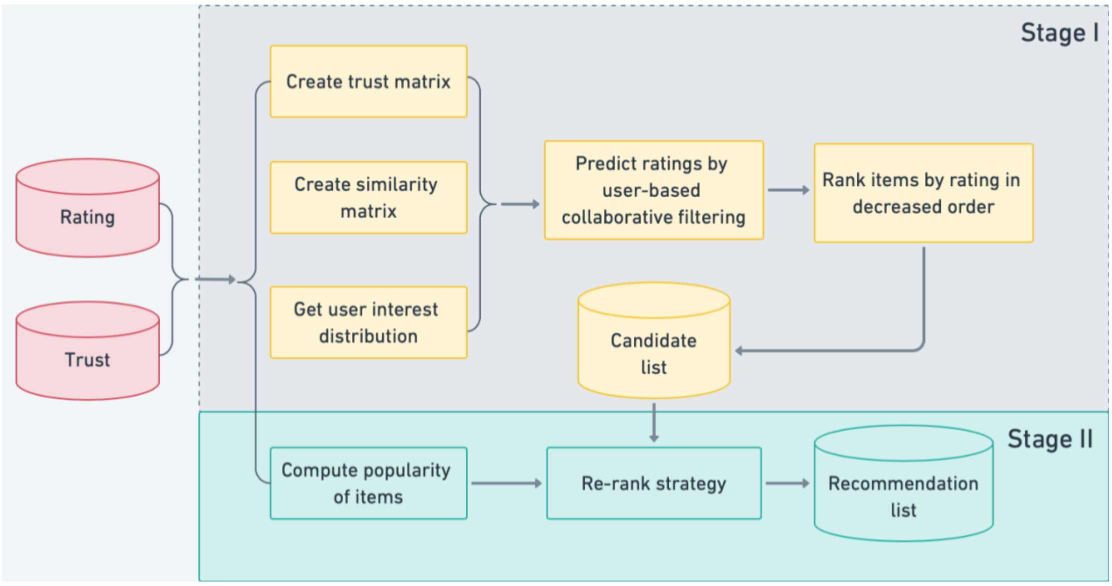
Stage I
Rating
Trust
Create trust matrix
Create similarity matrix
Get user interest distribution
Predict ratings by user-based collaborative filtering
Rank items by rating in decreased order
Candidate list
Stage II
Compute popularity of items
Re-rank strategy
Recommendation list

Figure 1. Overview of the main stages of the proposed method.

3.1. Trust Ranking

In the first stage of the proposed DBTS method, the user trust network $G = <U,T_D>$ is constructed according to user trust relationships. Each node $u \in U$ of network G represents a neighbor of the active user v. The T_D denotes the direct trust set of v. Each edge $T_{u,v} \in T_D$ represents the direct trust between users u and v.

However, methods that only consider direct trust often have limited effectiveness when the sparse data problem occurs. Implicit trust relationships that enrich the trust relationships of users can be obtained by further analyzing rating values. That is, although user u and v do not have a direct trust relationship, an implicit trust value is calculated by applying the Pearson correlation coefficient to the two users based on their available ratings. However, numerous implicit trust relationships are not required. According to the Three Degree of Influence Rule [43], a strong connection with user u, i.e., a short connection distance to user u (less than 3 degrees) might lead to a similar behavior, while a weak connection with user u, i.e., a long connection distance to user u (more than 3 degrees) only transmits information. Therefore, more users with no direct relation would be connected by expanding the implicit trust relations to their third-degree neighbors. As described in [37], the trust-based similarity values $TS_{u,v}$ between user u and v can be calculated as:

$$TS_{u,v} = \begin{cases} \frac{2\times sim(u,v)\times T_{u,v}}{sim(u,v)+T_{u,v}} & sim(u,v) + T_{u,v} \neq 0 \text{ and } sim(u,v) \times T_{u,v} \neq 0 \\ T_{u,v} & sim(u,v) = 0 \text{ and } T_{u,v} \neq 0 \\ sim(u,v) & sim(u,v) \neq 0 \text{ and } T_{u,v} = 0 \end{cases} \quad (1)$$

where $T_{u,v}$ denotes the trust relation between user u and v, which is calculated as:

$$T_{u,v} = \frac{H - N_{u,v} + 1}{H} \quad (2)$$

where H denotes the maximum trust propagation distance among all users. The $N_{u,v}$ denotes the trust propagation distance between users u and v. Trust can propagate from one network node to another via multiple paths. Here, the indirect trust value is set to the trust value for the shortest path. The $sim(u, v)$ is the similarity value for users u and v, which is computed as:

$$sim(u, v) = \frac{\sum_{i \in I_{u,v}} (R_{u,i} - \overline{R_u})(R_{v,i} - \overline{R_v})}{\sqrt{\sum_{i \in I_{u,v}} (R_{u,i} - \overline{R_u})^2} \sqrt{\sum_{i \in I_{u,v}} (R_{v,i} - \overline{R_v})^2}} \times \omega_u \tag{3}$$

where $R_{u,i}$ and $R_{v,i}$ are the ratings of user u and user v, respectively, for item i and where R_u and $\overline{R}_v$ are the average ratings for user u and user v, respectively. The $I_{u,v}$ is the item set rated by both user u and user v. Sometimes, users have few mutually rated items but high rating similarities. Therefore, mutual rating factor $\omega_u = |I_{u,v}|/|I_u|$ is introduced to moderate the influences of mutual ratings in the similarity calculation. The $|I_{u,v}|$ is the number of items rated by both user u and user v. The $|I_u|$ represents the number of items rated by user u.

Finally, according to the trust-based similarity values $TS_{u,v}$, users can be divided into a high trust value set ($TS_{u,v} \geq \theta$) and a low trust value set ($TS_{u,v} < \theta$), where θ is the predefined trust threshold.

3.2. User Interest Distribution

3.2.1. Acquisition of Item Category

To construct a multi-interest model that can be used to obtain the interest distribution of a user, all item categories must be identified, and the user rating information for all item categories must be collected. The K-means clustering algorithm is generally used to cluster items [41]. The item sets are defined as $I = \{i_1, i_2, \ldots, i_m\}$, the user sets are defined as $U = \{u_1, u_2, \ldots, u_n\}$, and the item category sets after clustering are defined as $C = \{c_1, c_2, \ldots, c_k\}$.

3.2.2. User Interest Distributions

The user rating values and number of rated items are used to calculate user interests. A user who has given high rating values for items in a certain item set is assumed to be very interested in these items. The interest distribution and average rating given by user u for item set c are computed as:

$$ID_{u,c} = \frac{M_{u,c}}{M_u} \text{ and } \overline{R\prime}_{u,c} = \frac{R_{u,c}}{R_{max}} \tag{4}$$

where $ID_{u,c}$ represents the interest distribution of user u for c. The $M_{u,c}$ represents the quantities of item c that have been rated by user u. The M_u represents the quantities of all items that have been rated by user u. The $R'_{u,c}$ represents the rating preference for category c. The $R_{u,c}$ represents the average rating that user u has given for c. The R_{max} represents the highest rating given by user u for each item. For user u, the preference for c is calculated as:

$$pref_{u,c} = \frac{2 \times ID_{u,c} \times \overline{R}'_{u,c}}{ID_{u,c} \times \overline{R}\prime_{u,c}} \tag{5}$$

where $pref_{u,c} = \{pref_{u,c1}, pref_{u,c2}, \ldots, pref_{u,ck}\}$ represents the preferences of user u for all item categories. User u participates in the calculation of interest weight only when the preference of user u is greater than that of user v. The mutual preference of users u and v is calculated as:

$$pref_{u,v} = \frac{\sum_{c \in C}^{C} \left| pref_{v,c} >= pref_{u,c} \right|}{\left| pref_u \right|} \tag{6}$$

3.2.3. Trust and Interest Weight Aggregation

In the trust propagation process, the calculation of a similarity weight for two users who are low-trust neighbors must consider not only their trust relations, but also their mutual interests. Therefore, two thresholds are defined in the calculation of aggregation weights:

$$W_{u,v} = \begin{cases} TS_{u,v} & TS_{u,v} \geq \theta \\ \gamma TS_{u,v} + (1-\gamma)pref_{u,v} & 0 < TS_{u,v} < \theta \\ 0 & \text{else} \end{cases} \tag{7}$$

Definition 1. *Trust threshold θ. If the trust between two users has a value greater than θ, the users are high trust neighbors. One user might accept the recommendation of the other user even if they do not have similar preferences. If the trust between the two users is less than θ, they are low trust neighbors. Their trust relations and multi-interest preferences should be considered simultaneously in the recommending process.*

Definition 2. *Interest threshold γ. If two users are low trust neighbors, they have small overlapped interest fields. It will increase the possibility of finding potential interests for the target user, while the recommendation accuracy is decreased. Therefore, the interest threshold γ is set that two users participate in the weight calculation process only when their interest weight exceeds γ.*

3.3. Rating Prediction and Top-N Re-Ranking

3.3.1. Rating Prediction

After ranking the aggregated weights, the top n users with the highest weights are selected as neighbors who participate in the rating prediction. Further, according to the prediction ratings in multiple item categories, items with the highest rating values are selected for use in constructing the top-N recommendation candidate list. Ratings are predicted as:

$$P_{u,i} = \overline{R_u} + \frac{\sum_{v \in U_u} W(u,v) \times (R_{v,i} - \overline{R_v})}{\sum_{v \in U_u} W(u,v)} \tag{8}$$

3.3.2. Top-N Re-Ranking

The conventional recommendation list uses rating values to rank items. Given the difficulty of identifying the potential interests of users, its objective is simply to increase recommendation accuracy. Thus, a diversity factor is introduced to adjust the proportion of items with high diversity. The predicted ratings of the top M items in the candidate list are assumed to exceed the average ratings of users. Equation (9) compares the similarities between the [(1-β)N,M] items and the top βN items. The item with the lowest similarity is then selected for use as a replacement item.

$$min \sum_{i \in [0,\beta N], j \in [(1-\beta)N,M]} \left(\frac{R_i - R_j}{P_i - P_j} \right) \tag{9}$$

where R_i and R_j are the rating values that user u gives for items i and j, respectively, and where P_i and P_j are the popularity of items i and j, respectively.

For a clear understanding of the proposed DBTS algorithm, the detailed steps of the method are given in Algorithm 1.

Algorithm 1. Diversity Balancing for Two-Stage collaborative filtering (DBTS)

```
    Input: rating matrix $R$, social trust matrix $T$, active user $u$, trust threshold $\theta$, interest threshold $\gamma$, ranking
    strategy threshold $\beta$.
    Output: the top-N recommendation list.
Begin
        //Stage I: calculate user similarities.
Step 1   Calculate similarities between users using (1).
Step 2   Divided users into low similarities set and high similarities set using $\theta$.
        //Stage II:calculate the integrated weights.
Step 3   Use K-means to cluster items.
Step 4   Calculate interest weights between active user and other users using (6).
Step 5   Integrate interest weights and similarities using (7).
        //Rating prediction and ranking.
Step 6   Sort users based on their weights in descending order.
Step 7   Choose the top $n$ users as active user neighbors.
Step 8   Predict unknown ratings for the active user.
Step 9   Select $n$ items with highest predicted ratings.
Step 10  Calculate replacement items using (9).
Step 11  Put items with the lowest ratings into recommendation list.
Step 12  Generate the top-N recommendation list.
        End
```

4. Experiments and Analysis of Results

The hardware environment that we use in this study is an MAC system of i7-4870HQ CPU@2.50 GHz and 16GB RAM. The Python3 programming language and third-party libraries such as SciKit-learn and Pandas are used for algorithm implementation and data processing. The dataset is divided into training set (80%) and testing set (20%) for five-fold validations. The training set is randomly divided into five groups. The training set is constructed by selecting four groups randomly, and the fifth group is used as test set. The experiment result is the average value of five experiments.

To evaluate the performance of the DBTS method with well-known recommendation models, we compare the method in this paper with four other algorithms. The focus of the DBTS algorithm is to generate a list of top-N recommendations, so Precision and Recall are used in terms of accuracy. The indicators that include personalized diversity (PD) and aggregate diversity (AD) are used in diversity. The four recommendation algorithms that participate in the comparison are as follows:

User-based Collaborative Filtering, UCF [1]: The UCF method calculates similarity between pairs of users by using the cosine similarity formula. It builds neighborhoods of users who have similar rating patterns. Then the algorithm makes a list of recommendations on unseen items based on opinions of the neighbors.

MoleTrust(MT) [6]: MoleTrust is a local trust propagation algorithm. It calculates the local trust score of two users by walking their social network and by propagating trust along trust edges. Then it predicts the ratings a user gave to an item based on his/her trusted friends.

Latent Dirichlet Allocation (LDA) based Model [44]: It models users as documents, items as words. Then the LDA model is used to generate recommendations.

Bayesian Personalized Ranking (BPR) [45]: BPR is a ranking algorithm based on matrix decomposition. By observing positive observations, real negative feedback (the user is not interested in buying the item) and missing values (the user might want to buy the item in the future), a pairwise relation between users and multiple items is established to conduct recommendation ranking.

4.1. Dataset

The FilmTrust (http://trust.mindswap.org/FilmTrust/) dataset was used to evaluate the proposed method. This dataset was collected from an internet movie website that enables users to give feedback and share opinions with others. Users rate movies on a scale from 0.5 to 4. A high rating value represents a high preference for an item. The dataset contains 35,497 ratings given by 1508 users for 2071 movies. The rating matrix has a density of 1.14%. A user who trusts another user is denoted by a value of 1 in the trust matrix. Otherwise, a user is denoted by a value of 0 in the trust matrix. The trust matrix includes 1853 trust relations among 1642 users. The trust matrix has a density of 0.42%. Fivefold cross validation was used to divide the dataset into a training set (80%) and a testing set (20%).

4.2. Evaluation Metrics

4.2.1. Accuracy Metrics

We use the well-known metrics of Precision and Recall to evaluate the accuracy of the algorithm. Precision, a concept borrowed from information retrieval [26], is defined as

$$\text{Precision} = \frac{t_p}{t_p + f_p} \tag{10}$$

where t_p means true positive, denoting the correctly classified positive examples;f_p means false positive, denoting incorrectly classified positive examples. Precision usually decreases as the length of the recommendation list increases. A high precision value indicates that many users are interested in the recommended items.

Recall measures the chances that a relevant item is selected for recommendation.

$$\text{Recall} = \frac{t_p}{t_p + f_n} \tag{11}$$

where f_n is the false negative, denoting the incorrectly classified negative examples. The Recall value usually increases as the length of recommendation list increases.

4.2.2. Diversity Metric

Previous works have generally used two criteria to evaluate recommendation diversity [26,27]:

First, personalized diversity (PD) is used to describe the dissimilarity between any two items in the recommendation list. As the diversity of the recommended items increases, the probability of users finding items of interest increases. The PD is defined as

$$\text{PD} = \frac{1}{|U|}\sum_{u\in U}\left(1 - \frac{\sum_{i,j\in L(u), i\neq j}(1 - sim(i,j))}{\frac{1}{2}\left|L(u)\right|\left(\left|L(u)\right| - 1\right)}\right) \tag{12}$$

where $|U|$ is the number of users and where $sim(i, j)$ is the similarity between item i and item j; $L(u)$ is the recommendation list of the users and $|L(u)|$ is the number of items in the $L(u)$.

Secondly, aggregate diversity (AD) is measured by the total number of unique items recommended for all users. A high AD value ensures that more items can be seen by users of an RS. The AD is defined as

$$\text{AD} = \frac{\sum_{u\in U}\left|L(u)\right|}{|I|} \tag{13}$$

where $|I|$ is the number of recommended items for all users.

4.3. Sensitivity Analysis

The proposed method has three pre-defined parameters: trust threshold θ, interest threshold γ, and ranking strategy threshold β. The optimal values of θ and γ are determined by the highest accuracy the algorithm achieves. θ is used to distinguish high trust users from low trust users. Users with similarities below θ will participate in the stage 2 of the DBTS algorithm. In the following experiments, the number of neighbors is set to $K = 20$. The top n items in the candidate set are selected from the recommendation list set on $n = 20$.

4.3.1. Effect of Trust Threshold θ

An experiment was performed to determine the appropriate trust threshold θ; θ ranges from 0 to 1 in increments of 0.1. Figure 2 shows the effect of θ in the DBTS method.

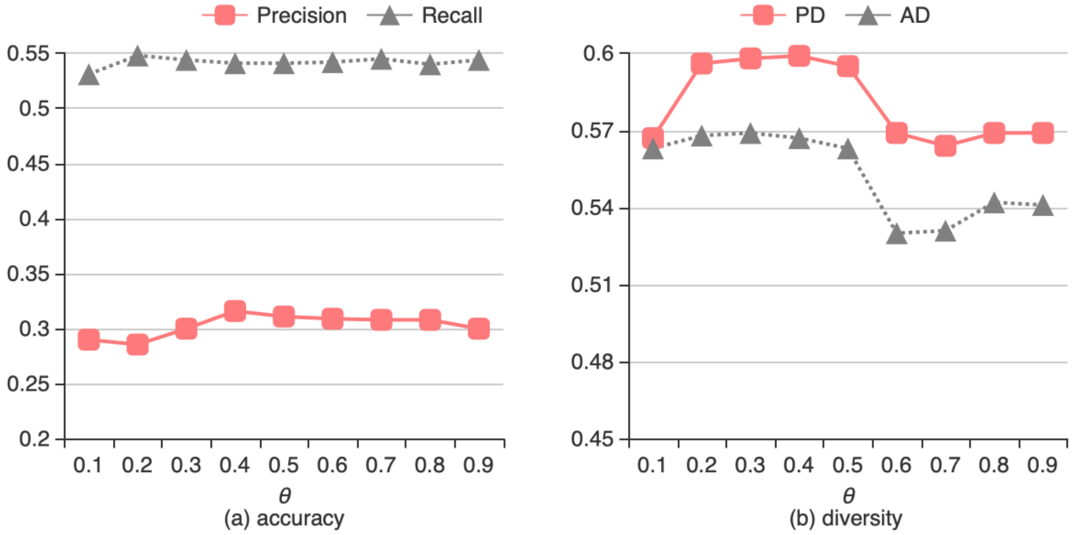

Figure 2. Effect of trust threshold θ on (**a**) accuracy and (**b**) diversity of Diversity Balancing for Two-Stage collaborative filtering (DBTS).

We first examine the effect of trust threshold θ, ranging from 0 to 1 with increment of 0.1. Figure 2a shows the accuracy and Figure 2b the diversity of the DBTS algorithm for different values of θ. The optimal value of θ leads to the best performance in terms of accuracy and diversity. From Figure 2a we can see that θ has very little influence on either precision or recall. The highest accuracy happens when $\theta = 0.4$. From Figure 2b we can see that the algorithm also has the highest diversity when $\theta = 0.4$. Therefore, the optimal value of θ is set to 0.4.

4.3.2. Effect of Interest Threshold γ

To determine the optimal interest threshold γ, we performed an experiment in which γ ranges from 0 and 1 with increments of 0.1. Figure 3 shows the accuracy and diversity of the DBTS method for varying γ.

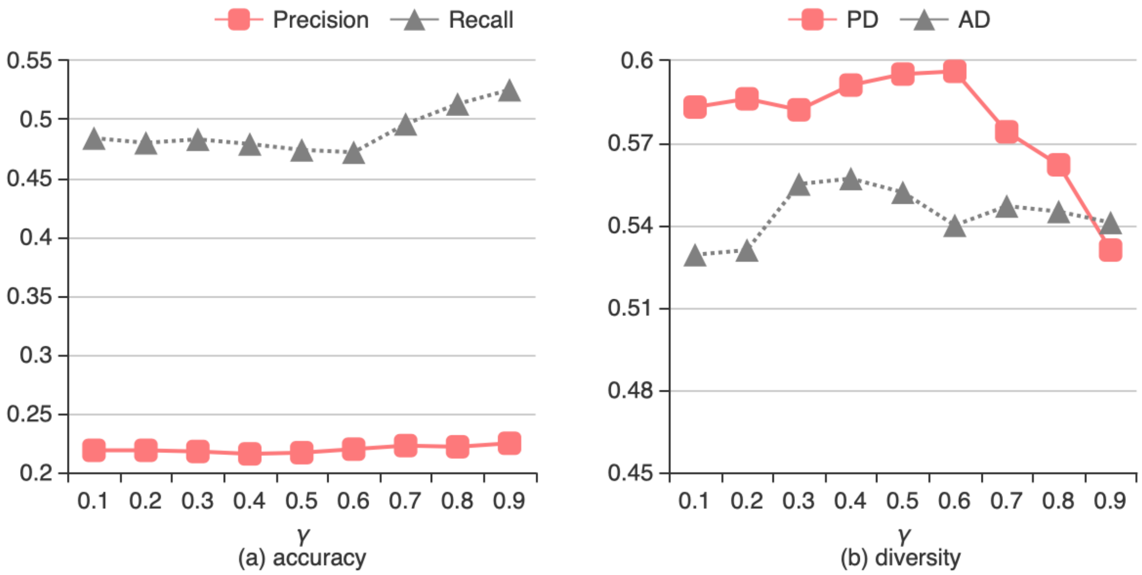

Figure 3. Effect of interest threshold γ on (**a**) accuracy and (**b**) diversity of DBTS.

Figure 3 shows the accuracy and diversity of DBTS for different values of interest threshold γ. From Figure 3a we can see that γ has very little influences on either precision or recall. Figure 3b shows that a high value of γ leads to low diversity performance of the DBTS algorithm in metrics of both PD and AD. Considering both accuracy and diversity, we set γ to the optimal value 0.4.

4.3.3. Effect of Weight Parameter β

The ranking strategy threshold determines whether the recommendation results emphasize accuracy or diversity. Figure 4 shows the accuracy and diversity of the DBTS algorithm for different values of β.

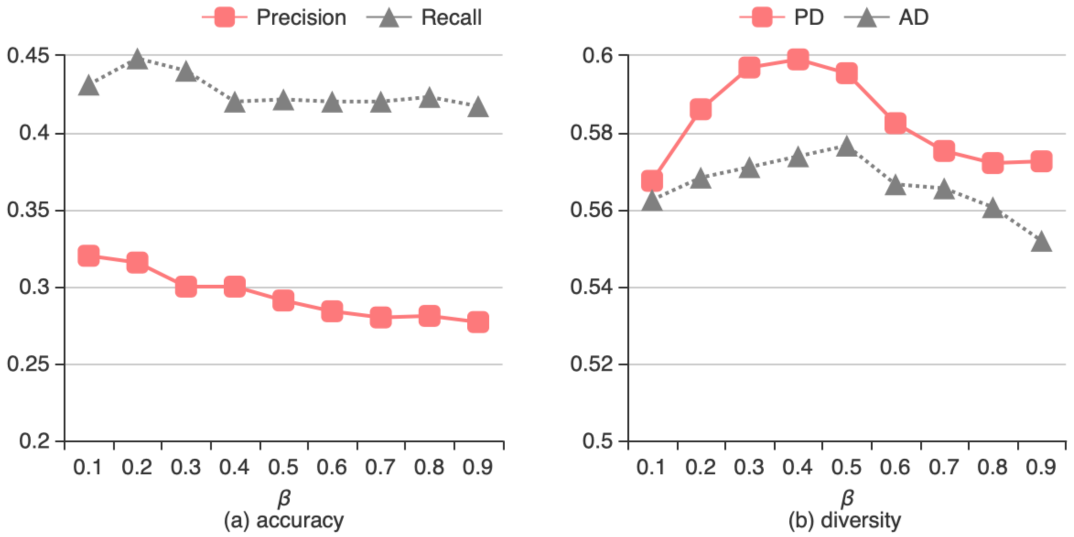

Figure 4. Effect of ranking strategy threshold β on (**a**) accuracy and (**b**) diversity of DBTS.

Figure 4a shows that, when β increases, there is a gradual decent on the curves of both precision and recall, which indicates the more items reordered in the recommendation list, the less accuracy the algorithm performs. In terms of accuracy metrics values, the method performs best when β ranges from 0.1 to 0.4. Figure 4b shows that both PD and AD curves first rise and then descend. When β is set from 0.3–0.5, the DBTS algorithm achieves the best diversity performance. Therefore, we set $\beta = 0.4$, which has the smallest decrease in accuracy and the highest increase in diversity.

4.3.4. The Execution Time of DBTS

Several experiments were conducted in order to compute the execution time of DBTS. The results are reported in Figure 5.

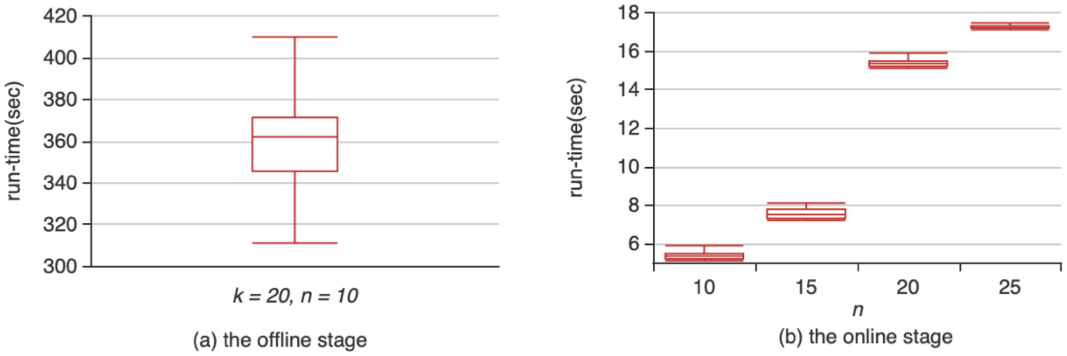

Figure 5. Execution time of (**a**) offline stage and (**b**) online stage.

Like other two-stage recommendation methods, the DBTS algorithm has an offline computing stage and an online recommendation stage. First, we experimented the offline stage of the DBTS algorithm. Ten independent experiments were conducted, which are shown in Figure 5a. It can be seen that the offline execution time has a mean value of 360 s and a standard deviation of 37.7 s. This stage has a long computation time because it needs to calculate trust and interest similarities according to the trust propagation algorithm. Then we experimented the online stage of the DTBS algorithm. The number of recommendation list was set as 10, 15, 20, and 25. Four groups of experiments were carried out with 10 independent ones for each group. The experiment results are shown in Figure 5b. When the number of recommendations is set as $n = 10$, the online stage's execution time is around 3 s. As the length of the recommendation list increases up to 20 and more, the algorithm's execution time increases to above 15 s.

4.3.5. Discussion of Experimental Results

To validate the effectiveness of the proposed method, the DBTS algorithm was compared with several conventional recommendation algorithms, in metrics of accuracy and diversity. Figure 6 shows the results for the four algorithms.

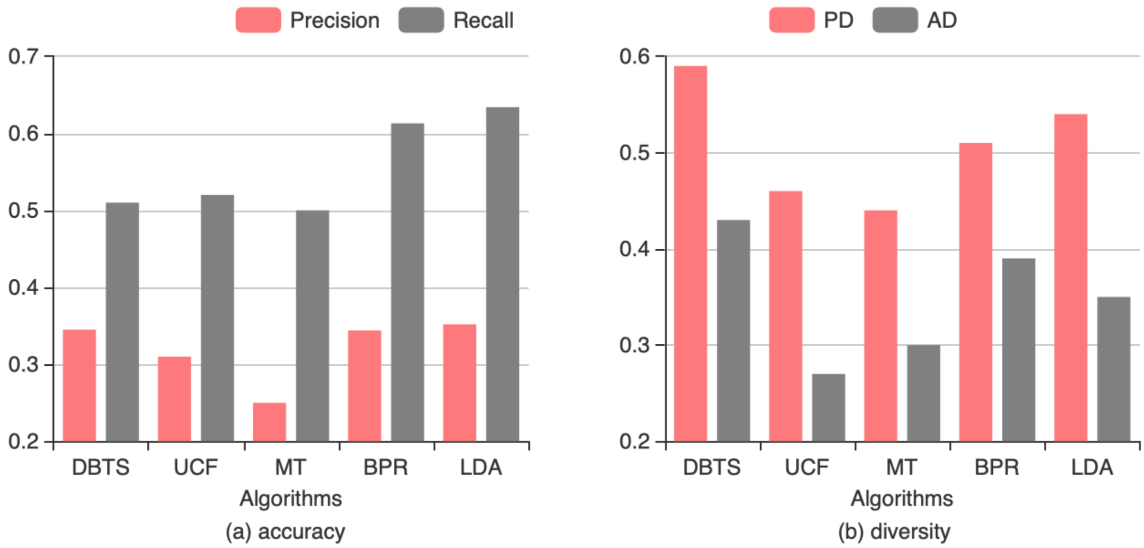

Figure 6. Comparison of (**a**) accuracy and (**b**) diversity between DBTS and other algorithms.

Figure 6a shows that the DBTS has higher accuracy compared to UCF (13%) and MoleTrust (40%), but has poorer accuracy than BPR (2%) and LDA (7%). The UCF has lower accuracy because only user similarities determined by ratings of corresponding items are used to predict user preferences. When UCF is applied in social networking environments, its performance is the worst among the three methods because it does not consider the social relations of users or the popularities of items.

Figure 6b indicates that the DBTS algorithm achieves the highest diversity compared to other algorithms. The recommendations of BPR and LDA are highly biased towards popular items, which hugely limited their diversity performance. The UCF has lower diversity because, as the number of recommended items increases, UCF only considers user preferences regarding popular items. Thus, personalized diversity and aggregate diversity are decreased. MoleTrust treats the trust of users equally and ignores differences in user selections under different trust values, which decreases accuracy. The DBTS uses a diversity factor to adjust the proportion of re-ranked items, which simultaneously improves aggregate diversity and personalized diversity.

5. Conclusions

The novel DBTS recommender method proposed in this study alleviates monotonous recommendations in RSs based on the social relationships and multiple interests of users. The proposed method is performed in two stages. The rating prediction stage sets a trust threshold and an interest threshold to integrate the trust similarities of users with multiple interest distributions. The problem of sparse data is also alleviated. In the recommendation list generating stage, a ranking strategy is applied to increase diversity with an acceptable loss in accuracy. The recommendation list is then generated and adjusted according to the popularity and diversity of items. The proposed DBTS method was validated through a series of experiments based on a real-world dataset. The validation results indicate that DBTS effectively improves both the aggregate diversity and the personality diversity of recommendations while maintaining high accuracy.

Several directions for future research can further improve the performance of the proposed method. First, since DBTS requires arbitrarily pre-defined ranking threshold values, our future studies will investigate the use of an optimization algorithm for automatically choosing the proper parameters. Second, we will investigate how time and context information can cause dynamic changes in user interests.

Author Contributions: The algorithm architecture was proposed by L.Z. (Liang Zhang) and Q.W. Numerical data analyses were performed by Q.W. The manuscript was prepared by L.Z. (Liang Zhang), W.-H.H., L.Z. (Lei Zhang) and B.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Shandong Province Natural Science Foundation of China, grant number ZR2018MG005 and grant number ZR2016GM12.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Konstan, J.A.; Miller, B.N.; Maltz, D.; Herlocker, J.L.; Gordon, L.R.; Riedl, J. GroupLens: Applying collaborative filtering to Usenet news. *Commun. ACM* **2000**, *40*, 77–87. [CrossRef]
2. Lu, J.; Wu, D.; Mao, M.; Wang, W.; Zhang, G. Recommender system application developments: A survey. *Decis. Support Syst.* **2015**, *74*, 12–32. [CrossRef]
3. Cremonesi, P.; Koren, Y.; Turrin, R. Performance of recommender algorithms on top-N recommendation tasks. In Proceedings of the 2010 ACM Conference on Recommender Systems, Barcelona, Spain, 26–30 September 2010.
4. Balabanovic, M.; Shoham, Y. Content-based, collaborative recommendation. *Commun. ACM* **1997**, *40*, 66–72. [CrossRef]
5. Wang, Z.; Sun, L.; Zhu, W.; Yang, S.; Li, H.; Wu, D. Joint social and content recommendation for user-generated videos in online social network. *IEEE Trans. Multimed* **2013**, *15*, 698–709. [CrossRef]
6. Massa, P.; Avesani, P.; Tiella, R. A Trust-enhanced recommender system application: Moleskiing. In Proceedings of the 2005 ACM Symposium on Applied Computing, Santa Fe, New Mexico, 13–17 March 2005.
7. Jamali, M.; Ester, M. A matrix factorization technique with trust propagation for recommendation in social networks. In Proceedings of the Fourth ACM Conference on Recommender Systems, Barcelona, Spain, 26–30 September 2010; pp. 135–142.
8. Moradi, P.; Ahmadian, S.; Akhlaghian, F. An effective trust-based recommendation method using a novel graph clustering algorithm. *Phys. A Stat. Mech. Its Appl.* **2015**, *436*, 462–481. [CrossRef]
9. Hernando, A.; Ortega, F. A probabilistic model for recommending to new cold-start non-registered users. *Inf. Sci.* **2017**, *376*, 216–232. [CrossRef]
10. Lika, B.; Kolomvatsos, K.; Hadjiefthymiades, S. Facing the cold start problem in recommender systems. *Expert Syst. Appl.* **2014**, *41*, 2065–2073. [CrossRef]
11. Afra, S.; Aksaç, A.; Õzyer, T.; Alhajj, R. Link prediction by network analysis. In *Prediction and Inference from Social Networks and Social Media*; Springer: Cham, Germany, 2017; pp. 97–114.
12. Yang, B.; Lei, Y.; Liu, J.; Li, W. Social collaborative filtering by trust. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1633–1647. [CrossRef]
13. Lu, Y.; Tsaparas, P.; Ntoulas, A.; Polanyi, L. Exploiting social context for review quality prediction. In Proceedings of the 19th International Conference on World Wide Web, Raleigh, NC, USA, 26–30 April 2010; pp. 691–700.
14. Ma, H.; Yang, H.; Lyu, M.R.; King, I. Sorec: Social recommendation using probabilistic matrix factorization. In Proceedings of the 17th ACM Conference on Information and Knowledge Management, Napa Valley, CA, USA, 26–30 October 2008; pp. 931–940.
15. Ma, H.; King, I.; Lyu, M.R. Learning to recommend with social trust ensemble. In Proceedings of the 32nd International ACM SIGIR Conference on Research, Boston, MA, USA, 19–23 July 2009; pp. 203–210.
16. Ma, H.; Zhou, D.; Liu, C.; Lyu, M.R.; King, I. Recommender systems with social regularization. In Proceedings of the fourth ACM International Conference on Web Search and Data Mining, Hong Kong, Chnia, 9–12 February 2011; pp. 287–296.
17. Guo, G.; Zhang, J.; Smith, N.Y. TrustSVD: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In Proceedings of the AAAI, Hyatt Regency, Austin, TX, USA, 25–30 January 2015; pp. 123–129.
18. Wang, J.C.; Chiu, C.C. Recommending trusted online auction sellers using social network analysis. *Expert Syst. Appl.* **2008**, *34*, 1666–1679. [CrossRef]
19. Moradi, P.; Ahmadian, S. A reliability-based recommendation method to improve trust-aware recommender systems. *Expert Syst. Appl.* **2015**, *42*, 7386–7398. [CrossRef]

20. Azadjalal, M.M.; Moradi, P.; Abdollahpouri, A.; Jalili, M. A trust-aware recommendation method based on Pareto dominance and confidence concepts. *Knowl. Based Syst.* **2016**, *116*, 130–143. [CrossRef]
21. Koohi, H.; Kiani, K. A new method to find neighbor users that improves the performance of Collaborative Filtering. *Expert Syst. Appl.* **2017**, *83*, 30–39. [CrossRef]
22. Kong, X.; Jiang, H.; Wang, W.; Bekele, T.M.; Xu, Z.; Wang, M. Exploring dynamic research interest and academic influence for scientific collaborator recommendation. *Scientometrics* **2017**, *113*, 369–385. [CrossRef]
23. Davoodi, E.; Kianmehr, K.; Afsharchi, M. A semantic social network-based expert recommender system. *Applied Intell.* **2013**, *39*, 1–13. [CrossRef]
24. Hamedani, E.M.; Kaedi, M. Recommending the long tail items through personalized diversification. *Knowl. Based Syst.* **2019**, *125*, 348–357. [CrossRef]
25. Wang, S.F.; Gong, M.G.; Li, H.L.; Yang, J.W. Multi-objective optimization for long tail recommendation. *Knowl. Based Syst.* **2016**, *104*, 145–155. [CrossRef]
26. Gogna, A.; Majumdar, A. DiABlO: Optimization based design for improving diversity in recommender system. *Inf. Sci.* **2017**, *378*, 59–74. [CrossRef]
27. Gogna, A.; Majumdar, A. Balancing accuracy and diversity in recommendations using matrix completion framework. *Knowl. Based Syst.* **2017**, *125*, 83–95. [CrossRef]
28. Hou, L.; Liu, K.; Liu, J.; Zhang, R.T. Solving the stability-accuracy-diversity dilemma of recommender systems. *Phys. A Stat. Mech. Its Appl.* **2016**, *468*, 415–424. [CrossRef]
29. Hawalah, A.; Fasli, M. Utilizing contextual ontological user profiles for personalized recommendations. *Expert Syst. Appl.* **2014**, *41*, 4777–4797. [CrossRef]
30. Shambour, Q.; Lu, J. An effective recommender system by unifying user and item trust information for B2B applications. *J. Comput. Syst. Sci.* **2015**, *81*, 1110–1126. [CrossRef]
31. Yuan, W.W.; Guan, D.H.; Lee, Y.K.; Lee, S.U.; Hur, S.J. Improved trust-aware recommender system using small-worldness of trust networks. *Knowl. Based Syst.* **2010**, *23*, 232–238. [CrossRef]
32. Guo, G.; Zhang, J.; Thalmann, D. Merging trust in collaborative filtering to alleviate data sparsity and cold start. *Knowl. Based Syst.* **2014**, *57*, 57–68. [CrossRef]
33. Parvin, H.; Moradi, P.; Esmaeili, S. TCFACO: Trust-aware collaborative filtering method based on ant colony optimization. *Expert Syst. Appl.* **2019**, *118*, 152–168. [CrossRef]
34. Lee, W.P.; Ma, C.Y. Enhancing collaborative recommendation performance by combining user preference and trust-distrust propagation in social networks. *Knowl. Based Syst.* **2016**, *106*, 125–134. [CrossRef]
35. Pasquale, D.M.; Lidia, F.; Messina, F. Providing recommendations in social networks by integrating local and global reputation. *Inf. Sci.* **2018**, *78*, 58–67.
36. Bedi, P.; Sharma, P. Trust based recommender system using ant colony for trust computation. *Expert Syst. Appl.* **2012**, *39*, 1183–1190. [CrossRef]
37. Massa, P.; Avesani, P. Trust metrics on controversial users: Balancing between tyranny of the majority. *Int. J. Semant. Web Inf. Syst.* **2007**, *3*, 39–64. [CrossRef]
38. Tommaso, D.N.; Rosati, J.; Tomeo, P.; Eugenio, D.S. Adaptive multi-attribute diversity for recommender systems. *Inf. Sci.* **2014**, *328*, 234–253.
39. Vargas, S.; Castells, P. Improving sales diversity by recommending users to items. In Proceedings of the ACM Conference on Recommender Systems, ACM 2016, Boston, MA, USA, 15–19 September 2016; pp. 145–152.
40. Pathak, A.; Patra, B.K. A knowledge reuse framework for improving novelty and diversity in recommendations. In Proceedings of the ACM Press the Second ACM IKDD Conference 2015, Bangalore, India, 18–21 March 2015; pp. 11–19.
41. Adomavicius, G.; Kwon, Y.O. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Trans. Knowl. Data Eng.* **2012**, *24*, 896–911. [CrossRef]
42. Wang, J.; Liu, Y.; Sun, J.; Jiang, Y.; Sun, C. Diversified recommendation incorporating item content information based on MOEA/D. In Proceedings of the 2016 49th Hawaii International Conference on System Sciences (HICSS), Koloa, HI, USA, 5–8 January 2016; IEEE: Piscataway, NJ, USA, 2016.
43. Mingyang, W.; Chongchong, J.; Donghui, Y. Social friend recommendation mechanism based on three-degree influence. *J. Comput. Appl.* **2015**, *35*, 1984–1987.

44. Steyvers, M.; Griffiths, T. Probabilistic topic models. *Handb. Latent Semant. Anal.* **2007**, *427*, 424–440.
45. Rendle, S.; Freudenthaler, C.; Gantner, Z.; Schmidt-Thieme, L. BPR: Bayesian personalized ranking from implicit feedback. In Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Vancouver, BC, Canada, 19–22 July 2009; AUAI Press: Barcelona, Spain, 2009; pp. 452–461.