future internet

MDPI

Article

Graph Representation-Based Deep Multi-View Semantic Similarity Learning Model for Recommendation

Jiagang Song [1], Jiayu Song [2,*], Xinpan Yuan [3], Xiao He [4] and Xinghui Zhu [4]

[1] School of Computer Science and Engineering, Guangxi Normal University, Guiling 541004, China; songjg@stu.gxnu.edu.cn
[2] School of Computer Science and Engineering, Central South University, Changsha 410083, China
[3] School of Computer Science, Hunan University of Technology, Zhuzhou 412007, China; xpyuan@hut.edu.cn
[4] College of Information and Intelligence, Hunan Agricultural University, Changsha 410128, China; aac@stu.hunau.edu.cn (X.H.); zhuxh@hunau.edu.cn (X.Z.)
* Correspondence: jiayusong@csu.edu.cn; Tel.: +86-185-7491-9704

Abstract: With the rapid development of Internet technology, how to mine and analyze massive amounts of network information to provide users with accurate and fast recommendation information has become a hot and difficult topic of joint research in industry and academia in recent years. One of the most widely used social network recommendation methods is collaborative filtering. However, traditional social network-based collaborative filtering algorithms will encounter problems such as low recommendation performance and cold start due to high data sparsity and uneven distribution. In addition, these collaborative filtering algorithms do not effectively consider the implicit trust relationship between users. To this end, this paper proposes a collaborative filtering recommendation algorithm based on graphsage (GraphSAGE-CF). The algorithm first uses graphsage to learn low-dimensional feature representations of the global and local structures of user nodes in social networks and then calculates the implicit trust relationship between users through the feature representations learned by graphsage. Finally, the comprehensive evaluation shows the scores of users and implicit users on related items and predicts the scores of users on target items. Experimental results on four open standard datasets show that our proposed graphsage-cf algorithm is superior to existing algorithms in RMSE and MAE.

Keywords: GraphSAGE; graph embedding; collaborative filtering

check for updates

Citation: Song, J.; Song, J.; Yuan, X.; He, X.; Zhu, X. Graph Representation-Based Deep Multi-View Semantic Similarity Learning Model for Recommendation. *Future Internet* **2022**, *14*, 32. https://doi.org/10.3390/fi14020032

Academic Editor: Eirini Eleni Tsiropoulou

Received: 30 November 2021
Accepted: 17 January 2022
Published: 19 January 2022

1. Introduction

With the rapid development of Internet technology, mining and analyzing massive network information has become a hot topic and a challenging problem. A recommendation system can help users solve the problem of information overload when there is no clear demand or a huge amount of information, and provide users with accurate and fast business information (such as about goods, projects, services, etc.), which has became a common interest and research hotspot of industry and academia in recent years [1]. However, there are various types of data and wide application scenarios. Recommendation systems will encounter the challenges of cold start and sparse matrix. The collaborative filtering technique is the most widely used method in recommendation systems, which predicts future preferences of users by analyzing their historical behavioral data [2–4]. At present, collaborative filtering has been favored by many researchers in the field of recommendation. The fusion of collaborative filtering technology with the recommendation method can effectively solve the problems of cold start and sparse matrix in traditional recommendation systems, and improve the performance and accuracy of recommendation systems. However, data sparsity and cold start are the main problems that limit the performance of collaborative filtering recommendation algorithms.

The emergence of social networks has led to an increasing number of recommendation algorithms, such as SoRec, RSTE, SocialMF, TrustMF, and so forth. These use the rich information provided by social networks to address the recommendation problem. However, in primitive social networks, trust relationships tend to be binary. Intuitively, this trust relationship representation is coarse-grained due to the differences in the intensity of users' trust in different people. In addition, the sparse trust relationships in social networks make fewer social relationship data available [5]. In fact, many users do not have direct trust relationships. However, they have many common neighbors who are likely to have high levels of trust between them. Considering such indirect, implicit trust relationships when modeling recommendation models can effectively improve the performance of recommendation models. However, traditional social network-based recommendation algorithms tend to ignore these implicit trust relationships. To model the global and local structure of information networks, a series of graph embedding (GE) models were proposed.

Graph embedding models embed large-scale information networks into a low-dimensional space, where a low-dimensional feature vector represents each network node. This low-dimensional feature vector effectively preserves the structural information of the information network and can be applied in various graph-based machine learning tasks, such as node classification, clustering, link prediction and visualization. The GraphSAGE model is an inductive learning framework that can efficiently generate unknown vertices embedding by using the attribute information of vertices [6,7]. It is used to derive the user trust relationships from the original social network that hold both local and global information of the social network [8–10], and a graph-embedded model-based collaborative filtering recommendation algorithm is proposed. Intuitively, the low-dimensional feature representation of user nodes in social networks can be learned through graph embedding and can be integrated into traditional social network-based recommendation algorithms to address the problems of coarse granularity. However, few research efforts have considered integrating graph embedding techniques into traditional social network-based recommendation algorithms to improve the performance of traditional social network-based recommendation.

The Method. This paper proposes a collaborative filtering recommendation algorithm based on a graph embedding model to address the problems of traditional social network-based recommendation algorithms. First, the graph embedding model technique is used to learn the low-dimensional feature representation of users in social networks. This low-dimensional feature representation of users simultaneously preserves the social network's global and local structural information, that is, the first-order and second-order trust between users. Then, a fine-grained trust relationship between users is inferred based on the low-dimensional feature representation of the users. Finally, the rating weights of trusted and similar users on the target item are averaged to predict the users' ratings of the target item. Experimental results on four benchmark datasets show that the graph embedding model-based collaborative filtering algorithm outperforms the traditional collaborative filtering algorithm.

Contributions. The main contributions of this paper are listed as follows.

- To overcome the challenges of cold start, we propose a novel GraphSAGE based collaborative filtering recommendation method, called GraphSAGE-CF. This paper tries to integrate GraphSAGE with a collaborative filtering technique for recommendation systems;
- A GraphSAGE based user embedding method is developed to learn low dimensional feature representation of global and local structures of users in social networks. Then, the implicit trust relationship between users can be measured;
- We conduct comprehensive experiments on four commonly used benchmark datasets. The experimental results demonstrate that our method outperforms the state-of-the-art.

Roadmap. The rest of this paper is organized as follows: in Section 2, the existing research related to our work is reviewed. In Section 3, the problem of recommendation and the related notions are formulated. In Section 4, the collaborative filtering algorithm based on the graph embedding model is proposed and the details of this method are discussed.

In Section 5, the comprehensive experiments are discussed. We conclude this paper in Section 6.

2. Related Work

2.1. Collaborative Filtering

The collaborative filtering recommendation algorithm is the most widely used recommendation method. It was proposed by Goldberg et al. [11] in 1992, was applied to news filtering in 1994, and has become one of the most popular recommendation algorithms in the corporate world [12]. There are three main categories of collaborative filtering-based recommendation methods—user-based collaborative filtering [13], item-based collaborative filtering [14,15], and matrix factorization based collaborative filtering algorithms [16]. The core of the user-based collaborative filtering algorithm is to first identify users with high similarity to the target users by their preferences, predict the target users' ratings of the items of interest based on the ratings of the similar users, and recommend the series of items with the highest ratings to the users. The idea of the item-based collaborative filtering algorithm is similar, and recommendations are made by calculating the similarity between items. The matrix decomposition-based collaborative filtering algorithm represents the user's item rating information in the form of a matrix. It represents the user and the item in the low-dimensional space through the matrix decomposition operation, which portrays the correlation between household items as the inner product of the user's item vectors. However, in many cases, it may encounter problems such as low recommendation performance, cold start, and long tail, which are caused by the highly sparse and uneven distribution of data [17,18].

The most widely used recommendation technique is the collaborative filtering recommendation algorithm. It mines the user's hidden preferences from the user's historical activity records and makes recommendations based on the user's hidden preferences. Collaborative filtering algorithms are mainly divided into three categories—memory-based collaborative filtering recommendation algorithms, model-based collaborative filtering algorithms and hybrid recommendation algorithms. Matrix-based recommendation algorithms have received a lot of attention in the industry because of their scalability and predictive ability in handling large scale data. PMF, SVD++, NMF, MMMF and NPCA are typical matrix-based recommendation algorithms.

SoRec [19] first proposed a social recommendation model based on the shared representation of user feature matrix, which laid a solid foundation for subsequent researchers to carry out research on collaborative filtering algorithms based on matrix decomposition. Based on different ways of constructing social information, TrustMF, a social recommendation model based on the relationship between trust and trusted, was proposed by Yang et al. [20]. TrustMF models trusting and trusted behaviors from the perspective of trust relationship generation, while SoRec models them only from the mathematical perspective, and the matrix decomposition process does not have any practical meaning. Ma et al. [21] also proposed a recommendation model RSTE for weighted social information that takes into account the preferences of trusted friends. A new social recommendation model based on a trust propagation mechanism in social networks, SocialMF [22], was proposed by Jamali et al. Similar to SocialMF, Ma et al. [23] proposed a social regularized recommendation model, SoReg, which argues that trust relationships and friend relationships are two different behaviors; a trust relationship is related to similar interests and hobbies among users, while a friend relationship is based entirely on real life social relationships. This information is added to the original SVD++ [24] to propose the TrustSVD model [25].

In traditional collaborative filtering algorithms, the extreme sparsity of the user-item rating matrix seriously affects the performance of the collaborative filtering algorithm. For example, it is difficult for the collaborative filtering-based recommendation algorithm to find other users with similar interests based on the user's rating information when the user's rating information is scarce or missing. This paper uses the graph embedding model to improve the traditional user-based collaborative filtering algorithm. This is especially to further integrate fine-grained user trust relationships based on the user-based collaborative filtering algorithm and to consider explicit and implicit trust between users to alleviate the data sparsity and cold start problems in the traditional collaborative filtering algorithm.

2.2. Graph Embedding

Graph embedding is a graph representation learning method that maps high-dimensional vectors to a low-dimensional space. The main objective of graph embedding is to achieve the following two aspects: the learned embedding vectors in the low-dimensional space can reconstruct the original graph structure, and the learned embedding vectors can support the inferential analysis of the graph. Depending on the object of embedding, graph embedding can be divided into two categories—embedding of the whole graph and embedding of nodes or edges in large-scale graphs. Graph embedding techniques fall into three main categories—matrix decomposition-based graph embedding methods [26–31], random wander-based graph embedding methods [32–34], and deep learning-based graph embedding methods [35–37].

2.2.1. Matrix Decomposition-Based Graph Embedding Method

The idea of the matrix decomposition based method is to represent the association information between nodes in a graph by means of a matrix, and the association moments of the graph by means of a matrix decomposition method. The array is dimensionally reduced to generate a low-dimensional vector for each node. Depending on the different properties of the matrix, matrix decomposition methods can be divided into two categories: Graph Laplacian Eigenmap Factorization and Node Proximity Matrix Factorization [38]. If similar pairs of nodes are mapped into a more distant vector space during graph embedding, Graph Laplacian Eigenmap Factorization gives the pair a larger penalty value. Typical Graph Laplacian Eigenmap Factorizations include LE [26], CGE [29], and Isomap [39].Through matrix decomposition methods, node Proximity Matrix Factorization incrementally simulates graph similarity in a low-dimensional space.

2.2.2. Graph Embedding Method Based on Random Wandering

The random wander-based graph embedding method uses different random wander strategies to obtain local or global information of the graph, generate a sequence of nodes, and generate a reduced dimensional embedding of the corresponding nodes by the Skip-Gram algorithm. Based on different graph structures, the random wandering methods can be divided into two categories: random wandering in homogeneous networks and random wandering in heterogeneous networks. Random wandering in homogeneous networks contains Node2vec [33] and so on. Random wandering in heterogeneous networks contains Metapath2vec [34] and so on.

2.2.3. Deep Learning Based Graph Embedding Method

The inability of shallow models to capture highly nonlinear structures usually results in the failure to generate optimal solutions. As a result, a large number of neural network computations are applied to graph data analysis to improve the shortcomings of shallow models. Typical deep learning-based graph embedding techniques include SDNE [35], DNGR [36], and GraphSAGE [37]. SDNE is a deep autoencoder that obtains the final embedding vector through a highly nonlinear function and an optimized objective function that effectively maintains the first- and second-order similarity of the network. DNGR is effective in maintaining the first- and second-order similarity of the network by combining random roaming and the depth autoencoder approach solves the problem whereby traditional linear dimensionality reduction methods cannot maintain the non-linear structure of graphs. GraphSAGE is a framework for inductive learning that can efficiently generate embeddings of unknown vertices using the attribute information of the vertices. The core idea is to generate the embedding vector of a target vertex by learning a function that aggregates the representation of neighbor vertices. It consists of three main steps: 1. sampling each vertex of the graph using a similar breadth-first traversal; 2. aggregating the information embedded in the neighbor vertices with an aggregation function; 3. using a nonlinear activation function to obtain the embedding vector of each vertex of the graph.

Specifically, in the sampling phase, Graph Sage first selects a point, then randomly selects the first-order neighbors of this point, and then randomly selects their first-order neighbors with these neighbors as the starting point. For example, in the following figure, we want to predict node 0, so first, first-order neighbors 2, 4, and 5 of node 0 are randomly selected, and then first-order neighbors 8 and 9 of node 2 are randomly selected; first-order neighbors of node 4 are randomly selected and first-order neighbors of nodes 5, 13, 15 are shown in Figure 1.

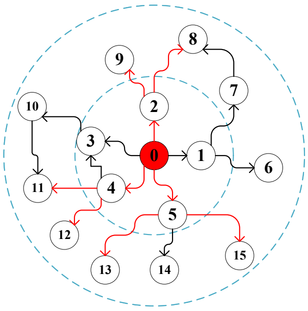

Figure 1. GraphSage detailed process.

In the aggregation stage, Graph Sage directly extracts the subgraph from the whole graph, starting from the most edge node, updating the nodes layer by layer inward to generate the target node embedding. Specifically, first aggregate the 2-hop neighbor features, generate one-hop neighbor embedding and then aggregate one-hop neighbor embedding to generate target node embedding, so as to obtain two-hop neighbor information. Graph Sage also analyzes the properties of aggregation functions and proposes that aggregation functions need to meet the following conditions: (1) Aggregation functions need to adapt to the number of aggregation nodes—no matter the number of aggregated nodes, aggregation is performed. The dimensions of the feature vectors obtained after the operation must be the same; (2) The aggregation function needs to have permutation invariance—the order of the obtained feature vectors should be independent of the input order of the data; (3) Aggregate functions must be differentiable. Therefore, Graph Sage provides several applicable aggregation functions– average aggregation, pooling aggregation, LSTM aggregation.

In this paper, GraphSAGE is used to learn the embedding representation of user nodes in a social network and is combined with a traditional collaborative filtering algorithm.

The graph embedding technique embeds a large-scale information network into a low-dimensional vector space, where each node in the information network is represented by a low-dimensional feature vector. This low-dimensional feature vector representation preserves the structural information of the information network. The network embedding technique has a wide range of applications in machine learning tasks such as node classification, clustering, link prediction, and visualization. Typical graph embedding techniques include DeepWalk, Graph Factorization, and GraphSAGE.

Graph Factorization uses a matrix decomposition model to learn embedded representations of large-scale networks. However, the objective function of the matrix decomposition used in Graph Factorization is not designed for networks and therefore cannot preserve information about the global structure of information networks. Moreover, the Graph Factorization model is only applicable to directionless information networks. DeepWalk learns embedded representations of information networks by using a random wander algorithm. However, the DeepWalk model does not clearly describe which properties of the preserved information network. In addition, the DeepWalk model expects nodes with similar neighbor structures to have similar low-dimensional feature representations and does not take into account the relationships between low-dimensional feature vectors between nodes with direct linking relationships.The DeepWalk model is only applicable to unweighted networks. GraphSAGE learns large-scale information networks through objective functions that preserve the local and global structure of the large-scale information network. In addition, GraphSAGE uses an edge-sampling strategy to address limitations in the classical stochastic gradient descent algorithm. GraphSAGE models are applicable to large-scale homogeneous information networks, including undirected/directional, weighted, and unweighted information networks. In addition, the representation of node features learned from the GraphSAGE model simultaneously preserves first-order and second-order similarity between nodes.

Given the efficiency of the GraphSAGE model, this paper uses the GraphSAGE model to learn the embedded representation of user nodes in social networks and to integrate the embedded representation of user nodes in social networks into a traditional collaborative filtering recommendation algorithm.

3. Problem Definition

3.1. Recommend Formal Description of the Problem

The co-filtering algorithm mainly uses the user-item scoring matrix to predict the user's rating of an item. The user-item scoring matrix consists of a set of M users and a set of N items. Each item rui in R is the rating of item i by user u. The scoring data γ_{ui} can take any real number, but normally the scoring data are an integer and $\gamma_{ui} \in \{0,1,2,3,4,5\}$, where a value of 0 means that the user has not rated this item.

Social network information is usually represented as a directed social relationship graph $G = (U, E)$, where U is the set of users and the set of edges E represents the social trust relationship between users. $t_{uv} \in [0,1]$ represents the trust weight between user u and user v. $T_{uv} = 0$ means that no trust relationship has been established between user u and user v. The social network information is usually represented as a directed social relationship graph $G = (U, E)$, where U is the set of users and the set of edges E represents the social trust relationship between users. The trust relationships between all users form the trust relationship matrix T.

The purpose of the social network-based recommendation system is to use the user-item scoring matrix and social network information to predict user u's rating of item i and thus recommend items that the user may be interested in.

3.2. User-Based Collaborative Filtering Recommendation Algorithm

The user-based collaborative filtering algorithm assumes that similar users have similar preferences, and the procedure is: (1) find users similar to the current active users by some kind of similarity metric; (2) use the weighted average of similar users' ratings of the target item to predict the current active users' ratings of the target item. Adopting appropriate metrics to calculate the similarity between users is key to user-based collaborative filtering algorithms. Typical similarity metrics include cosine similarity, Pearson's correlation coefficient, and adjusted cosine similarity. For example, according to the cosine similarity metric, the similarity between users u, v is defined as:

$$sim(u,v) = cos(u,v) = \frac{\sum_i \gamma_{ui} \times \gamma_{vi}}{\sqrt{\sum_i \gamma_{ui}^2} \times \sqrt{\sum_i \gamma_{vi}^2}}, \tag{1}$$

where i is the item that users u, v have jointly rated.

After the user-based collaborative filtering algorithm calculates the similarity between users according to some similarity metric, it finds the set of users that are most similar to the currently active users. Then, predict user u's rating of item j according to Equation (2).

$$\gamma_{uj}^R = \bar{\gamma_u} + \frac{\sum_{v \in D} sim(u,v) \times (\gamma_{vj} - \bar{\gamma_v})}{|\sum_{v \in D} sim(u,v)|}, \tag{2}$$

where $\bar{\gamma_u}$ denotes the average rating of user u for all rated items,$\bar{\gamma_v}$ denotes the average rating of user v for all rated items, and D denotes the set of users most similar to user u.

Finally, the user-based collaborative filtering algorithm recommends the K items with the highest prediction scores to the current user.

4. Collaborative Filtering Algorithm Based on Graph Embedding Model

Traditional collaborative filtering recommendation algorithms suffer from severe data sparsity and cold start problems. Social network-based collaborative filtering algorithms assume that friends have common interest preferences, and usually improve traditional collaborative filtering algorithms by exploiting the explicit trust relationship between users in social networks. However, traditional collaborative filtering algorithms based on social networks have the following problems: (1) the trust relationships between users in social networks are coarse-grained, that is, only 0 or 1 can be used to indicate the trust level between users; (2) only explicit trust relationships are considered, while implicit trust relationships are ignored. Explicit trust relationships for the observable, trust relationships with direct connections, it captures the local structure of the social network information. In addition, many users do not have a direct trust relationship, but they have many common neighbors and are likely to have a high level of trust between them. Even if two users do not establish a direct trust relationship, if they have a similar neighbor structure, it is likely that there is a strong trust relationship between these two users. In this paper, the trust relationship observed by the social network is referred to as first-order trust and the trust relationship derived from the neighbor structure is referred to as second-order trust. The second-order trust relationship captures information about the global structure of the social network. Considering both explicit first-order trust relationships and implicit second-order trust relationships when modeling a recommendation model can effectively improve the performance of the recommendation model.

Therefore, in this paper, the GraphSAGE model is used to derive the user trust relationships from the original social network that hold both local and global information of the social network, and a graph-embedded model-based collaborative filtering recommendation algorithm is proposed. In the following, we first describe the process by which the GraphSAGE model learns the low-dimensional embeddings of a node that preserves both local and global information, and then explain in detail the model of the collaborative filtering recommendation algorithm based on the graph embedding model.

4.1. GraphSAGE

As a classic algorithm of the graph embedding model, GraphSAGE proposes an algorithm framework that can easily obtain the embedding of new nodes. Its basic idea is to learn how a node's information is aggregated through the characteristics of its neighbor nodes. After learning the aggregation function, use the known characteristics and neighbor relationships of each node to quickly obtain the embedding of the new node.

Select Aggregator

Mean Aggregator. Since the neighbors of the nodes in the graph are irregular and disordered, it is necessary to construct an aggregate function that is symmetric, that is, changes the order of inputs but the output of the function remains the same, while having high expressive power.

$$h_v^k \leftarrow \sigma(W \cdot MEAN(\{h_v^{k-1}\} \cup \{h_u^{k-1}, \forall u \in N(v)\})). \tag{3}$$

Once the vector representation of the node is generated, the mean aggregator splices the *k*th-1st layer vectors of the target node and the neighbor node, and then averages each dimension to obtain the result and conducts a nonlinear transformation to generate the *k*-th layer vector representation of the target node.

Pooling Aggregator. The pooling aggregator performs a nonlinear transformation on the neighbor representation vector of the target vertex, followed by a pooling operation to splice the result with the representation vector of the target vertex, and finally a nonlinear transformation to obtain the kth level representation vector of the target vertex.

$$h^k_{N(v)} = max(\{\sigma(W_{pool}h^k_{u_i} + b), \forall u_i \in N(v)\}), \tag{4}$$

$$h^k_v = \sigma(W^k \cdot CONCAT(h^{(}_v k - 1^{)}, h^{k-1}_{N(u)})). \tag{5}$$

LSTM Aggregator. LSTM is more expressive than simple averaging operations; however, since the LSTM function is not symmetric with respect to the input, it requires a disorder operation on the neighbors of the vertices when used.

4.2. Collaborative Filtering Algorithm Based on Graph Embedding Model

An embedded representation of user nodes learned by the GraphSAGE model captures both the local and global structure of the social network. Based on the embedded representation of the user nodes, the trust between users is defined as:

$$SIM_{uv} = \frac{h^T_u \cdot h_v}{||h_u||_2 ||h_v||_2}. \tag{6}$$

Compared to the original trust relationships in social networks, the user trust level obtained by Equation (6) is not only fine-grained and dense, but also captures both explicit first-order trust relationships and implicit second-order trust relationships between users of social networks. In other words, if there is no explicit trust relationship between the users, but the users have similar neighborhood structure; the user trust Suv calculated according to Equation (6) captures the second-order trust relationship between the users.

Likewise to the memory-based collaborative filtering algorithm, after computing the trust relationship between users according to Equation (6), Equation (7) is used to calculate the prediction score of user u for item *j*.

$$\gamma^T_{uj} = \bar{\gamma_u} + \frac{\sum_{v \in L} SIM_{uv} \times (\gamma_{vj} - \bar{\gamma_v})}{|\sum_{v \in L} SIM_{uv}|}, \tag{7}$$

where $\bar{\gamma_u}$ denotes the average rating of user u on all rated items, $\bar{\gamma_v}$ denotes the average rating of user u on all rated items, $\gamma_v j$ denotes the rating of user v on item j, and H denotes the set of users that user u trusts the most. Note that, as shown in Equation (2), in the classical user-based collaborative filtering algorithm, $sim(u, v)$ denotes the similarity between users computed from user rating data, while in Equation (7), $SIM_u v$ denotes the user trust derived from the social network structure, which captures the explicit first-order trust relationship between social network users and the implicit second-order trust relationship.

The final rating of the item is influenced by both similar users and trusted users. Therefore, the weighting parameter is used to balance the two influences, that is, the final predicted rating of item *j* by user u is defined as:

$$\gamma_{uj} = \alpha\gamma^R_{uj} + (1 - \alpha)\gamma^T_{uj}, \tag{8}$$

where γ^R_{uj} denotes the predicted score of user u for item j based on the influence of similar users with similar scoring data, calculated according to Equation (2); γ^T_{uj} denotes the predicted score of user u for item j based on the influence of trusted users, calculated according to Equation (7). The two parts of the score are integrated by the weighting parameter α, which constitutes the final prediction score of user u on item j.

In summary, the framework of the collaborative filtering algorithm based on graph embedding is shown in Figure 2.

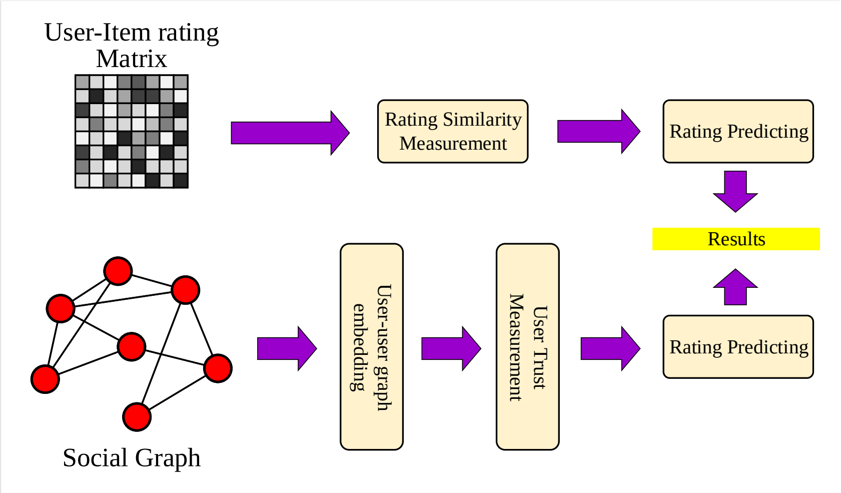

Figure 2. Framework of graph embedding model based collaborative filtering algorithm.

5. Experiment

In order to verify the performance of the GraphSAGE collaborative filtering recommendation algorithm, a comparison analysis with other popular recommendation algorithms is performed on a real dataset. In addition, the proposed GraphSAGE co-filtering algorithm is called "GraphSAGE-CF".

5.1. Dataset

Four real world datasets were selected for experimental comparative analysis. The public benchmark dataset Yelp2 contains users, businesses, and rating information across 11 metropolitan areas in four countries. Two subsets of public benchmark dataset Yelp2 (https://www.yelp.com/dataset (accessed on 15 January 2022)), North Carolina(NC) and Wisconsin (WI), were selected as the experiment datasets. The NC dataset includes 225,580 rating records, 22,737 users and 12,502 items, and 111,394 trust social relations. The WI dataset includes 80,643 rating records, 8386 users and 4593 items, and 34,099 trust social relations. The Epinions dataset includes user trust relationships, ratings and item categories, which is crawled from the popular social network website Epinions (www.epinions.com (accessed on 15 January 2022)). This dataset contains 764,352 rating records, 18,088 users and 261,649 items, 355,813 trust relationships. The LastFM (https://grouplens.org/datasets/hetrec-2011/ (accessed on 15 January 2022)) dataset contains 1874 users and 2828 items, 71,411 rating records and 25,174 relations of trust. Some of their important properties are summarized in Table 1. A brief introduction of them follows.

Table 1. Statistics of Evaluation Datasets.

Property	NC	WI	Epinions	LastFM
# Users	22,737	8386	18,088	1874
# Items	12,502	4593	261,649	2828
# Ratings	225,580	80,643	764,352	71,411
Density of relations	0.0793%	0.2093%	0.0161%	2.0286%
# social relations	111,394	34,099	355,813	25,174

5.2. Evaluation Metric

In this paper, we use the widely used evaluation metrics in recommendation algorithms: root mean squared error ($RMSE$) and mean absolute error (MAE) to evaluate the performance of recommendation algorithms.

$$RMSE = \sqrt{\frac{\sum_{(i,j)\in R_{test}} |r_{ij} - \hat{r_{ij}}|^2}{|R_{test}|}} \tag{9}$$

$$MAE = \frac{\sum_{(i,j)\in R_{test}} |r_{ij} - \hat{r_{ij}}|}{|R_{test}|}, \tag{10}$$

where r_{ij} denotes the actual score value, $\hat{r_{ij}}$ denotes the score value predicted by the system, and $|R_{test}|$ denotes the number of records in the test dataset. the smaller the $RMSE$ and MAE values, the better the recommendation performance of the recommended algorithm.

5.3. Settings

To validate the effectiveness of the recommendation algorithm proposed in this paper, the following comparison method is selected.

- **Baseline1** uses the average of all scored items scored by the current active users as the predicted scores of the target items by the current active users;
- **Baseline2** The user-based collaborative filtering algorithm finds users with similar interests to the current active users based on their historical ratings, and then uses the weight of similar users' ratings on target items to predict the average of the current active users' ratings on target items;
- **PMF** [40] was proposed by Mnih and Salakhutdinov, which can be considered a probabilistic extension of the SVD model. PMF learns the implicit feature vector of users and items from the user-item rating matrix and uses the inner product of the implicit feature vector of users and items to predict the missing items in the user-item rating matrix;
- **TrustSVD** [25] proposes an advanced SVD++ algorithm. Based on the original SVD++, it takes both explicit trust relationship and rating information of users into model construction;
- **DeepCoNN** [41] proposes a parallel framework by jointly using the users' feedbacks, two parallel neural networks are used to deal with users' and items' information synchronously.

The user–item scoring matrix is randomly partitioned five times, with 80% of the data extracted each time as the training set and the remaining 20% as the test set. The final run takes the average of the algorithm's results across the five test sets. For fair comparison, the parameters of different algorithms are set with reference to the corresponding literature or experimental results of the comparison algorithms, and the optimal performance of each comparison algorithm is obtained under these parameter settings. In Baseline2, the cosine is used to calculate the similarity between users; in PMF, the hidden feature vector dimension is 10 and the regularization factor $\lambda_u = \lambda_v = 0.001$; in GraphSAGE-CF, the cosine is used to calculate the similarity between users.

The experiment was run on a quad-core Intel(R) Core(TM) i5-7400H CPU with 3.00 GHz main frequency, 8 GB RAM, Window10 operating system and J2SE1.8.

5.4. Performance Comparison

In order to filter out trust relationships with weak trust, the trust threshold δ is set, and trust relationships between users with trust strength less than δ are filtered out. On the Epinions and LastFM datasets, the confidence thresholds δ = 0.95 and δ = 0.8, respectively, the number of neighbors N = 20, and the integration parameters α = 0.5 and 0.7, respectively, were set. On the NC and WI datasets, the confidence thresholds δ = 0.95 and δ = 0.8 and the integration parameters α = 0.9 and 0.8, respectively, were set. All comparison algorithms were performed on four datasets. The experimental results on each dataset are shown in Table 2, Table 3, Table 4 and Table 5, respectively.

Table 2. Performance comparison on the Epinions Dataset.

Competitor	*RMSE*	*MAE*
Baseline1	1.432	1.142
Baseline2	1.311	1.034
PMF	1.201	0.903
TrustSVD	1.135	0.867
DeepCoNN	1.113	0.859
GraphSAGE-CF	1.091	0.841

Table 3. Performance comparison on the LastFM Dataset.

Competitor	*RMSE*	*MAE*
Baseline1	1.143	0.879
Baseline2	1.037	0.794
PMF	0.923	0.723
TrustSVD	0.887	0.685
DeepCoNN	0.863	0.674
GraphSAGE-CF	0.842	0.668

Table 4. Performance comparison on the NC Dataset.

Competitor	*RMSE*	*MAE*
Baseline1	1.478	1.208
Baseline2	1.403	1.124
PMF	0.923	1.062
TrustSVD	1.174	0.904
DeepCoNN	1.133	0.886
GraphSAGE-CF	1.093	0.867

Table 5. Performance comparison on the WI Dataset.

Competitor	*RMSE*	*MAE*
Baseline1	1.298	1.035
Baseline2	1.207	0.947
PMF	0.923	0.898
TrustSVD	0.957	0.758
DeepCoNN	0.928	0.726
GraphSAGE-CF	0.891	0.714

From Tables 2–5, it can be observed that the performance of Baseline1 algorithms is poor, which is due to the fact that they do not take into account the differences in items and are non-personalized methods. Except for GraphSAGE-CF, PMF outperforms other comparison methods, again validating the effectiveness of the matrix decomposition method. On four datasets, the recommendation accuracy of the proposed collaborative filtering recommendation algorithm based on a graph embedding model is higher than the other comparison algorithms, validating the effectiveness of the proposed algorithm in this paper. On Epinions, LastFM, NC and WI datasets, the improvement of the proposed algorithm in this paper is 2.38%, 2.63%,2.60% 3.14% compared with the optimal results in DeepCoNN, respectively, using RMSE as the reference index. Based on the above rating prediction error, RMSE and MAE, the recommendation methods are compared on the Epinions, LastFM, NC and WI datasets. We have the following main findings:

- PMF consistently outperforms Baseline1 and Baseline2. Because Baseline1 only uses the average of all scored items scored by the active users as the predicted scores of the target items, Baseline2 uses the weight of similar users' ratings on target items to predict the average of the active users' ratings; At the same time, PMF learns the implicit feature vector of users and items from the user–item rating matrix and uses the inner product of the implicit feature vector of users and items;

- TrustSVD obtains a much better performance than PMF. Both methods take the SVD algorithm into model construction. However, TrustSVD uses an advanced SVD++ algorithm and takes direct trust relationship and rating information into the model construction;
- DeepCoNN performances are better than TrustSVD, PMF, Baseline1, and Baseline2. The reason is that DeepCoNN is based on two parallel neural networks, which further indicate the power of neural network models in social recommendations;
- Our proposed technique, GraphSAGE-CF, outperforms all the baseline methods. Compared with the above methods, our model learns low dimensional feature representation of the global and local structures of users in social networks to assist the rating prediction.

5.5. *Influence of Parameter α*

In the GraphSAGE-CF algorithm, the integration parameter α has a significant impact on the recommended performance. A larger α value means that the prediction scores are more heavily weighted by user ratings and rely more on scoring data in GraphSAGE-CF. Conversely, a smaller α value means that the predictive scores in GraphSAGE-CF are more dependent on the social network. In Epinions and NC, α is set to 0.9. In LastFM and WI, α is set to 0.4. in addition, the number of neighbors N is set to 5 in Epinions; the number of neighbors N is set to 5. In this section, a sensitivity analysis of the integration parameter α is performed through a set of experiments. The experimental results are shown in Figures 3–6.

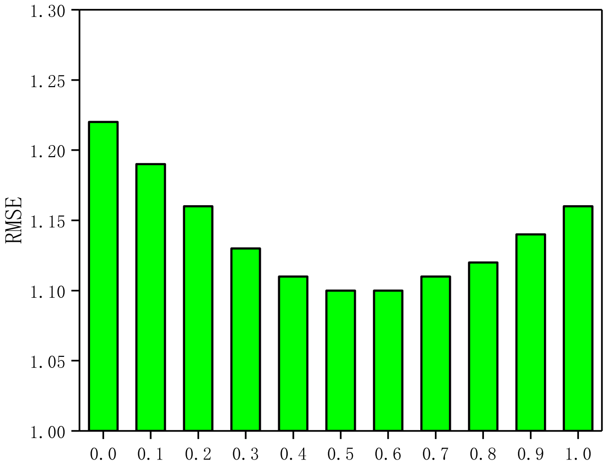

Figure 3. The Impact of parameter α on Epinions.

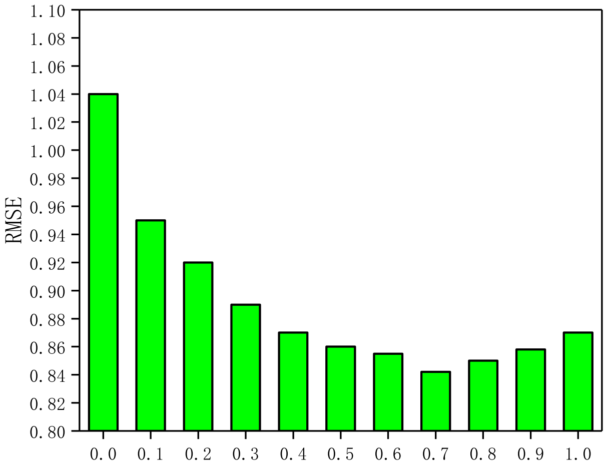

Figure 4. The Impact of parameter α on LastFM.

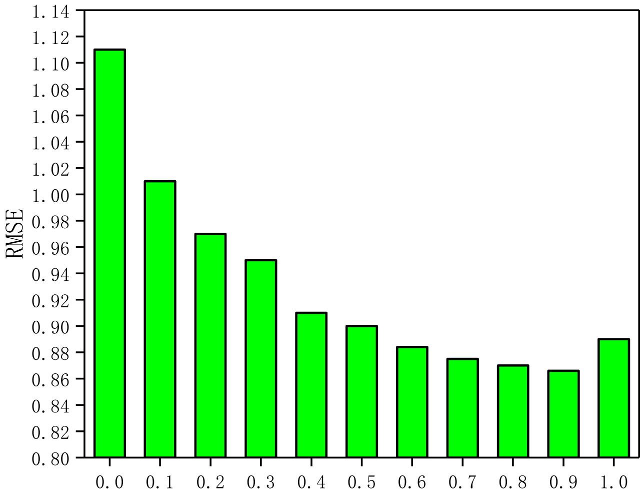

Figure 5. The Impact of parameter α on NC.

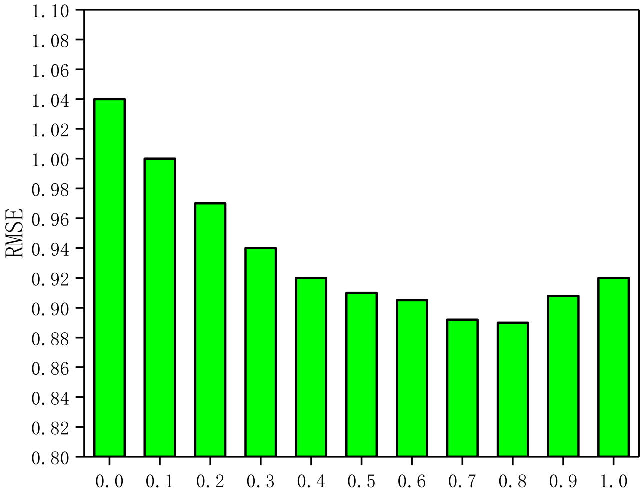

Figure 6. The Impact of parameter α on WI.

It can be seen from Figures 3–6 that the RMSE first decreases with increasing α in four datasets, and then gradually increases after reaching the minimum value. In addition, in Epinions, GraphSAGE-CF performs best when α is near 0.6; in LastFM, it performs best when α is around 0.7; in NC, it performs best when α is 0.8 and in WI it performs best when α is about 0.9. The results indicate that the prediction scores derived from both the scoring data and the social network data contribute to the final prediction scores and are more dependent on the prediction scores derived from the scoring data.

5.6. Influence of Parameter δ

Weakly trusting user trust relationships are filtered out and a small number of strongly trusting user trust relationships are fused in the GraphSAGE-CF. Conversely, a smaller value of δ means that a large number of user trust relationships with relatively low trustworthiness are fused in GraphSAGE-CF. In this section, a sensitivity analysis of the trust threshold δ is performed through a set of experiments. In Epinions and NC, δ is set to 0.8, 0.85, 0.9, and 0.95 and in LastFM and WI, δ is set to 0.4, 0.5, 0.6, 0.7, and 0.8. In addition, the number of neighbors N in both datasets is set to 5. The experimental results are shown in Figures 7–10.

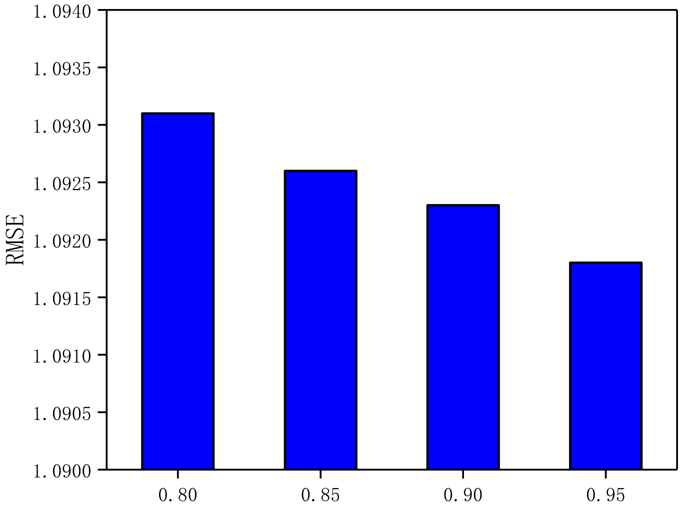

Figure 7. The Impact of parameter δ on Epinions.

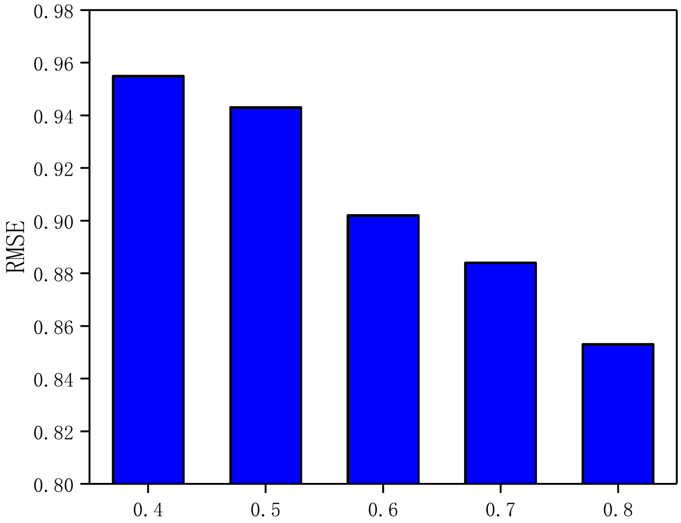

Figure 8. The Impact of parameter δ on LastFM.

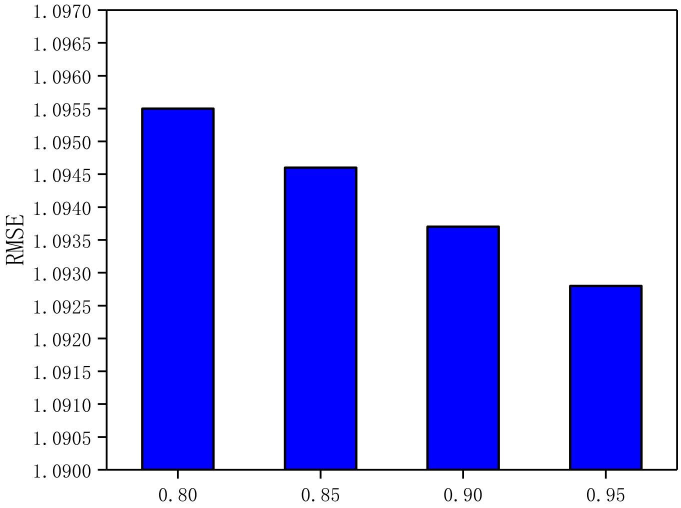

Figure 9. The Impact of parameter δ on NC.

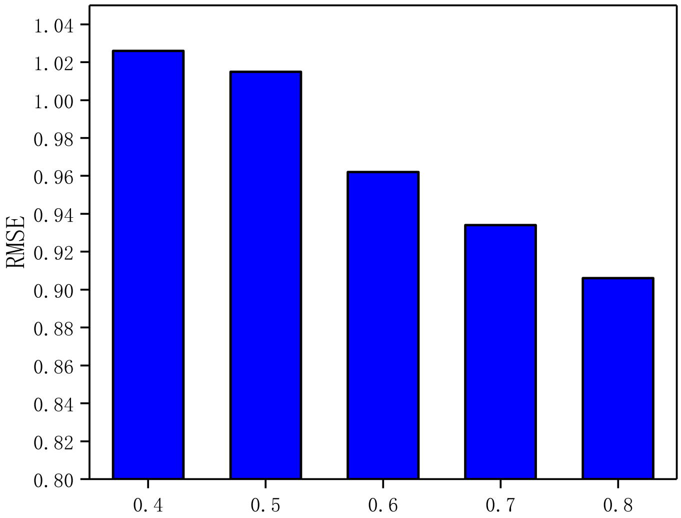

Figure 10. The Impact of parameter δ on WI.

From Figures 7–10, it can be observed that in the four datasets, the RMSE values have a rough trend of decreasing with increasing δ and improving recommendation accuracy. The results show that it is more beneficial for GraphSAGE-CF to integrate fine-grained and strong trust relationships with users to improve recommendation performance. Finally, GraphSAGE-CF did not obtain the lowest RMSE with the same δ value in both datasets, possibly because the network embedding technique learns different scales of user trust strength in different social networks. For example, in Epinions and NC, the maximum user trust strength is 0.9553, while in LastFM and WI, the maximum trust strength is 0.8769.

6. Conclusions

In this paper, a collaborative filtering recommendation algorithm based on a graph embedding model is proposed, which uses the graph embedding technique to learn the low-dimensional embedded representation of users in a social network and to derive fine-grained trust relationships between users based on their embedded picture. This fine-grained trust relationship captures the first-order and second-order trust levels between users. Finally, rating weights of trusted and similar users were used to predict users' ratings of the target item. The experimental results on real datasets demonstrate that the graph embedding model-based collaborative filtering algorithm outperforms the traditional collaborative filtering algorithm. This paper only focuses on the integration of graph embedding techniques in memory-based collaborative filtering algorithms. In matrix decomposition models, improving the performance of traditional collaborative filtering algorithms is considered a future research direction. More specifically, the users with many clicks are embedded by standard collaborative filtering; A weighted combination of embedded representations of the users with fewer clicks is used to compute other users' embedded graphics. In this way, some well-trained user representations can indirectly calculate another few-shot or zero-shot user representations, thereby improving the generalization performance on few-shot users or new users. At the same time, this representation model can flexibly handle new users that appear in the future without retraining.

Author Contributions: Data curation, X.Y.; Investigation, X.H.; Validation, X.Z.; Writing—original draft, J.S. (Jiagang Song); Writing—review & editing, J.S. (Jiayu Song) All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (NSFC 61836016,61672177,62072166).

Data Availability Statement: Not availability.

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

Abbreviations

The following abbreviations are used in this manuscript:

u	user
i	item
γ_{ui}	Scoring data of user u
γ_{uj}^{R}	user u's rating of item j
γ_{uj}^{T}	prediction score of user u for item j
$\bar{r}_u$	the average rating of user u for all rated items
$\bar{r}_v$	the average rating of user v for all rated items
D	the set of users most similar to user
L	the set of users that user u trusts the most
h_v^k	the k-th layer vector representation of the target node
α,δ	Weight parameter
$SIM(u,v)$	the user trust derived from the social network structure
h_u	The embedding of user u
h_v	The embedding of user v

References

1. Zhang, C.; Wang, Y.; Zhu, L.; Song, J.; Yin, H. Multi-graph heterogeneous interaction fusion for social recommendation. *Acm Trans. Inf. Syst. (TOIS)* **2021**, *40*, 1–26. [CrossRef]
2. Wang, Y. Survey on deep multi-modal data analytics: Collaboration, rivalry, and fusion. *Acm Trans. Multimed. Comput. Commun. Appl. (TOMM)* **2021**, *17*, 1–25. [CrossRef]
3. Wang, Y.; Zhang, W.; Wu, L.; Lin, X.; Fang, M.; Pan, S. Iterative views agreement: An iterative low-rank based structured optimization method to multi-view spectral clustering. *arXiv* **2016**, arXiv:1608.05560.
4. Wu, L.; Wang, Y.; Gao, J.; Wang, M.; Zha, Z.-J.; Tao, D. Deep coattention-based comparator for relative representation learning in person re-identification. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 722–735. [CrossRef] [PubMed]
5. Chen, Y.; Wang, Y.; Ren, P.; Wang, M.; de Rijke, M. Bayesian feature interaction selection for factorization machines. *Artif. Intell.* **2022**, *302*, 103589. [CrossRef]
6. Zhang, C.; Song, J.; Zhu, X.; Zhu, L.; Zhang, S. Hcmsl: Hybrid cross-modal similarity learning for cross-modal retrieval. *ACM Trans. Multimed. Comput. Commun. Appl. (TOMM)* **2021**, *17*, 1–22. [CrossRef]
7. Zhu, L.; Song, J.; Zhu, X.; Zhang, C.; Zhang, S.; Yuan, X. Adversarial learning-based semantic correlation representation for cross-modal retrieval. *IEEE Multimed.* **2020**, *27*, 9–90. [CrossRef]
8. Zhu, L.; Zhang, C.; Song, J.; Liu, L.; Zhang, S.; Li, Y. Multi-graph based hierarchical semantic fusion for cross-modal representation. In Proceedings of the 2021 IEEE International Conference on Multimedia and Expo (ICME), Shenzhen, China, 5–9 July 2021; pp. 1–6.
9. Zhu, L.; Song, J.; Yang, Z.; Huang, W.; Zhang, C.; Yu, W. DAP^{2}CMH: Deep adversarial privacy-preserving cross-modal hashing. *Neural Process. Lett.* **2021**, 1–21. [CrossRef]
10. Zhang, C.; Xie, F.; Yu, H.; Zhang, J.; Zhu, L.; Li, Y. PPIS-JOIN: A novel privacy-preserving image similarity join method. *Neural Process. Lett.* **2021**, 1–19. [CrossRef]
11. Goldberg, D.; Nichols, D.; Oki, B.M.; Terry, D. Using collaborative filtering to weave an information tapestry. *Commun. ACM* **1992**, *35*, 61–70. [CrossRef]
12. Qian, B.; Wang, Y.; Hong, R.; Wang, M.; Shao, L. Diversifying inference path selection: Moving-mobile-network for landmark recognition. *IEEE Trans. Image Process.* **2021**, *30*, 4894–4904. [CrossRef] [PubMed]
13. Albadvi, A.; Shahbazi, M. A hybrid recommendation technique based on product category attributes. *Expert Syst. Appl.* **2009**, *36*, 11480–11488. [CrossRef]
14. Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th International Conference on World Wide Web, Hong Kong, China, 1–5 May 2001; pp. 285–295.
15. Ambulgekar, H.; Pathak, M.K.; Kokare, M. A survey on collaborative filtering: Tasks, approaches and applications. In *Proceedings of the International Ethical Hacking Conference 2018, Kolkata, India, 2018*; Springer: Berlin, Germany, 2019; pp. 289–300.
16. Koren, Y.; Bell, R.; Volinsky, C. Matrix factorization techniques for recommender systems. *Computer* **2009**, *42*, 30–37. [CrossRef]
17. Guoxia, W.; Heping, L. Survey of personalized recommendation system. *Comput. Eng. Appl.* **2012**, *48*, 66–76.
18. Shi, Y.; Larson, M.; Hanjalic, A. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Comput. Surv. (CSUR)* **2014**, *47*, 1–45. [CrossRef]
19. 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, CIKM 2008, Napa Valley, CA, USA, 26–30 October 2008.
20. Yang, B.; Lei, Y.; Liu, D.; Liu, J. Social collaborative filtering by trust. In Proceedings of the IJCAI International Joint Conference on Artificial Intelligence, Beijing, China, 3–9 August 2013.

21. Ma, H.; King, I.; Lyu, M.R. Learning to recommend with social trust ensemble. In Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, 19–23 July 2009.
22. Jamali, M.; Ester, M. A matrix factorization technique with trust propagation for recommendation in social networks. In Proceedings of the ACM Conference on Recommender Systems, Barcelona, Spain, 26–30 September 2010.
23. Ma, H.; Zhou, D.; Liu, C.; Lyu, M.R.; King, I. Recommender systems with social regularization. In Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, 9–12 February 2011.
24. Koren, Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008.
25. Guo, G.; Zhang, J.; Yorke-Smith, N. Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.
26. Belkin, M.; Niyogi, P. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Adv. Neural Inf. Process. Syst.* **2001**, *14*, 585–591.
27. Tenenbaum, J.B.; Silva, V.D.; Langford, J.C. A global geometric framework for nonlinear dimensionality reduction. *Science* **2000**, *290*, 2319–2323. [CrossRef] [PubMed]
28. Roweis, S.T.; Saul, L.K. Nonlinear dimensionality reduction by locally linear embedding. *Science* **2000**, *290*, 2323–2326. [CrossRef] [PubMed]
29. Luo, D.; Ding, C.H.; Nie, F.; Huang, H. Cauchy graph embedding. In Proceedings of the ICML, Bellevue, WA, USA, 28 June–2 July 2011.
30. Cao, S.; Lu, W.; Xu, Q. Grarep: Learning graph representations with global structural information. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, Melbourne, Australia, 19–23 October 2015; pp. 891–900.
31. Ou, M.; Cui, P.; Pei, J.; Zhang, Z.; Zhu, W. Asymmetric transitivity preserving graph embedding. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1105–1114.
32. Perozzi, B.; Al-Rfou, R.; Skiena, S. Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 701–710.
33. Grover, A.; Leskovec, J. node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 855–864.
34. Dong, Y.; Chawla, N.V.; Swami, A. metapath2vec: Scalable representation learning for heterogeneous networks. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 135–144.
35. Wang, D.; Cui, P.; Zhu, W. Structural deep network embedding. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1225–1234.
36. Cao, S.; Lu, W.; Xu, Q. Deep neural networks for learning graph representations. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 16, pp. 1145–1152.
37. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive representation learning on large graphs. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 1024–1034.
38. Goyal, P.; Ferrara, E. Graph embedding techniques, applications, and performance: A survey. *Knowl. Based Syst.* **2017**, *151*, 78–94. [CrossRef]
39. Balasubramanian, M. The isomap algorithm and topological stability. *Science* **2002**, *295*, 7. [CrossRef] [PubMed]
40. Salakhutdinov, R.; Mnih, A. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 3–6 December 2007*; Platt, J.C., Koller, D., Singer, Y., Roweis, S.T., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2007; pp. 1257–1264.
41. Zheng, L.; Noroozi, V.; Yu, P.S. Joint deep modeling of users and items using reviews for recommendation. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, UK, 6–10 February 2017.