International Journal of Engineering & Technology, 12 (2) (2023) 38-47

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Do Trust based Social Recommendation Algorithms Work as Intended?

Chaitanya Krishna Kasaraneni[1*] and Mahima Agumbe Suresh[2]

[1]*Egen, Inc.*
[2]*San Jose State University*
[*]*chaitanya.kasaraneni@egen.solutions*

Abstract

Recommender systems are powerful tools that filter and recommend content/information relevant to a given user. Collaborative filtering is the most popular technique used in building recommender systems and it has been successfully incorporated in many applications. These conventional recommendation systems require a minimum number of users, items, and ratings in order to provide effective recommendations. This results in the infamous cold-start problem where the system is not able to produce effective recommendations for new users. Recently, there has been an escalation in the popularity and usage of social networks, which persuades people to share their experiences in the form of reviews and ratings on social media. The components of social media such as the influence of friends, interests, and enjoyment create the opportunities to develop solutions for sparsity and cold start problems of recommendation systems. This paper aims to observe these patterns and analyze three of the existing social recommendation systems, SocialMF, SocialFD, and GraphRec. SocialMF and SocialFD algorithms are based on matrix factorization and distance metric learning respectively whereas GraphRec is an attention based deep learning model. Through extensive experimentation with the datasets that these algorithms were tested on and one new dataset, we compared the results based on evaluation metrics including Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). To investigate how trust impacts the performance of these models, we evaluated them by modifying the trust and social component. Experimental results show that there is no conclusive evidence that trust propagation plays a major part in these models. Moreover, these models show a slightly improved performance in the absence of trust statements.

Keywords: *Distance Metric Learning; Matrix Factorization; Neural Networks; Recommender Systems; Social Networks; Social Recommendations*

1. Introduction

With the proliferation of Internet usage, the amount of information available over World Wide Web (WWW) has increased enormously. It is becoming increasingly necessary to recommend relevant parts of online information to users based on their preferences. Recommendation systems fill this gap by predicting “ratings” or “preferences” that a user would give to an item [1]. These are used in variety of areas, mostly in playlist recommendations on Spotify, video recommendations on Netflix and YouTube, product recommendation on Amazon and e-bay, or content recommendations on social media such as Facebook and Twitter[1].

In a recommendation system/recommender system (RS), there are a set of users and a set of items, where each user gives ratings to a subset of items available. The task of the RS is to predict the rating r that user u would give to a non-rated item i or to recommend user u with some items based on the ratings that are already given by user to other items.

RS are generally classified into two types: memory-based recommender systems and model-based recommender systems. Memory-based algorithms, also known as collaborative filtering RS, explore the user-item rating matrix and recommend based on the ratings of item i by a set of users whose rating profiles are most similar to that of user u[2]. Model-based approaches learn and only store the parameters of a model. As a result, these algorithms have no need to explore the rating matrix. Model-based approaches are very fast after the algorithms learn parameters of the model. The performance bottleneck for model-based approaches is the training phase, whereas memory-based approaches have no training phase. However, the prediction is slower as user-item matrix needs to be accessed several times.

In the present day, with the rapid increase in the popularity and usage of social networks, there is a dramatic growth in number of registered users and various products, which also leads to intractable increase of the cold start problem (new users into the system with less past social behavior) and the sparsity of datasets. Collaborative filtering works effectively when users have expressed a minimum number of ratings to have common ratings with other users in the dataset. For relatively new users, the performance suffers due to the cold start problem. In RS, cold start users are users who are either new to the platform or have given only a few ratings. Using similarity-based approaches, it is infeasible to find corresponding similar users since the cold start users only have a few ratings.

The interpersonal relationships, especially the friends' circles in social networks make it possible to solve the cold start and sparsity problem. The richness of social media give us some valuable insights to drive user recommendations, especially for items such as music, movies, news, brands, and travel. Many social network-based models for recommender systems have been developed to refine the performance but only a handful have considered social circles in their respective approaches. This gap motivates the development of an RS that considers the personal interests of users, interpersonal similarity [3] of interests with their friends, and influence of these interpersonal interests.
A social rating network consists of a social network with ratings expressed by each user to some items apart from creating social relations to other users. A sample social rating network is depicted in Figure 1.

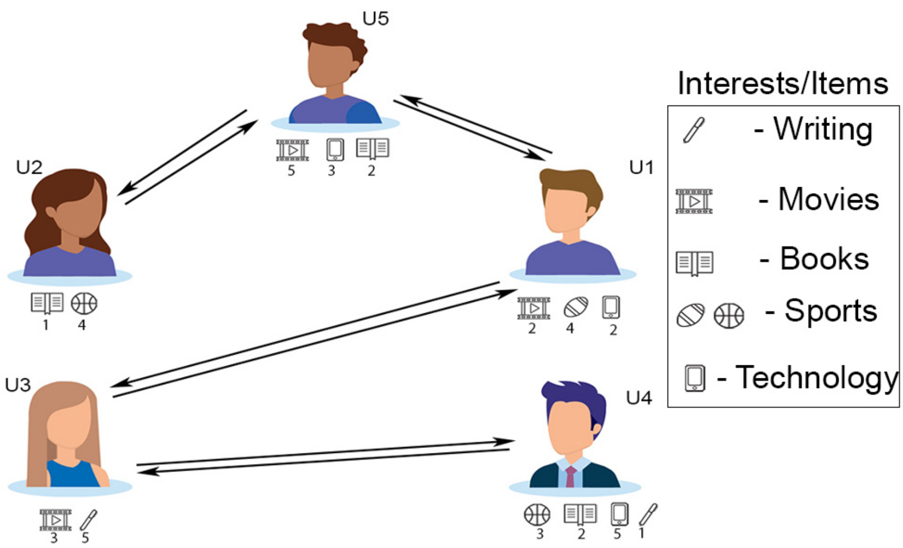

Figure 1: Sample Social Network indicating relations between users and their interests

Table 1 shows the matrix representation of the user-item ratings and Table 2 shows the user-user relationship. Here '1' indicates that user u trusts v. The terms "trust network" and "social network" are used synonymously in this paper.

Table 1: Sample User-Item Rating Matrix

	Sports	Phones	Movies	Writing	Reading
U1	4	2	2		
U2	4				1
U3			3	5	
U4	3	5		1	2
U5		3	5		2

Table 2: Sample Social Trust Matrix

	U1	U2	U3	U4	U5
U1			1		1
U2					1
U3	1			1	
U4			1		
U5	1	1			

Many RS approaches have been proposed using social rating networks [4],[5],[6],[7],[8],[9]. Of them [4],[5],[6],[7] are memory-based methods that explore a social network and find neighborhoods of direct or indirect trusted users and recommend to users by aggregating ratings. These approaches use transitive property to obtain trust to indirect neighbors. These memory-based algorithms are slower compared to model-based approaches in test phase since they have to traverse the entire social network.
Model-based RS using social rating networks have been developed in [8][9]. These techniques utilize the matrix factorization to obtain latent features for each user and item from the ratings observed. Experiments show that these model-based approaches perform better compared to state-of-the-art memory-based algorithms. But the major setback is that these algorithms do not take account of trust propagation. To solve this issue, SocialMF [10], a matrix factorization technique based recommendation model was proposed. This model includes trust propagation to improve the quality of recommendations. Optimizing this, another method called SocialFD[11] that incorporates distance metric learning alongside matrix factorization was proposed.
With the recent increase in use of graph neural networks, attention, and deep learning, an attention based deep learning model known as GraphRec [12] was developed. This model contains two components. The first one is to learn user latent factors that contains two separate aggregations, one for learning interactions between users and items in the user-item graph and the other for social aggregation. The second

component is extracting item latent factors which contains user aggregation. Finally, model parameters are learned via predictions by integrating both the components.
This paper evaluates the performance of SocialMF [10], SocialFD [11] and GraphRec [12], both in presence and absence of social trust. More accurately, the contributions of this paper are:

- Create a new large dataset, extending a prior dataset called TwitterEgo [32], with high quality social circle information extracted from Twitter
- Perform extensive experiments on the SocialMF, SocialFD, and GraphRec algorithms with the datasets that these algorithms were tested on and the new dataset based on TwitterEgo
- Compare the results of these experiments based on the evaluation metrics including Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE)
- Investigate the performance of these models in the following cases:
  - When there is no trust between any users i.e., the number of trust statements is 0,
  - When the users are friends only with themselves i.e., users trust only themselves, and
  - When the users are friends with everyone else excluding themselves

The rest of this paper is arranged as follows: Some related works are discussed in section 2. Section 3 summarizes the SocialMF [10], SocialFD [11] and GraphRec [12] models. Experiment methodology and datasets are explored in section 4. Experimental results and comparisons are analyzed in section 5. Finally, the paper is concluded with some directions for future work in section 6.

2. Related Work

This section reviews some of the works in recommendations using social network. Trust propagation is widely considered in memory-based approaches whereas model-based recommendation approaches broadly use matrix factorization [13][14][15]. But the major setback is that these algorithms do not consider social network of users. Model-based recommendation approaches have been developed which utilize matrix factorization technique for recommendations in social networks [8][9], but these approaches do not examine trust propagation. In this section, some model-based works in social networks are discussed after reviewing memory-based models.
Using a modified breadth first search technique on the trust network, a memory-based algorithm called TidalTrust [7] was proposed to determine a prediction. TidalTrust tries to find raters with the shortest distance from the user and combines their ratings weighted with trust between the user and these raters [7]. TidalTrust combines the trust value between user u's direct neighbors and v weighted by the trust values of u and its direct neighbors to compute the trust value between user u and v who are indirectly connected [7].
Another approach called MoleTrust which is similar to TidalTrust was introduced in [5]. The major difference is that MoleTrust [5] considers all raters till maximum-depth of the input irrespective of specific user or item. Backward exploration is used in MoleTrust, to compute the trust between users u and v. i.e., the calculated trust value is an aggregation of trust between user u and users who directly trust user v weighted by their direct trust values.
In [16], the authors proposed a maximum flow trust metric called Advogato. This approach helps in discovery of trusted users in an online community. Input for Advogato will be the total number of users to be trusted n. The algorithm needs to understand the whole network structure in order to transform the network to be able to edges of network with capacities. Furthermore, Advogato only calculates the nodes to trust but not different degrees of trust. This technique is not suitable for trust-based recommendations as the trusted users are independent of users and items in the network and the distinction between trusted users is negligible.
To consider enough ratings and excluding the noisy data, a random walk approach called TrustWalker was proposed in [4]. This approach combines both item-based and trust-based recommendations. This method not only considers ratings of required item, but also the ratings of similar items. The likelihood of considering these similar items increases with the increase in walk length. Additionally, this framework contains both trust-based and item-based recommendations as special cases. Experiments show that this algorithm outperforms other existing memory-based techniques allowing them to calculate confidence of predictions.
In [8], authors proposed an approach called STE which is a matrix factorization based approach for social network based recommendations. This approach is a sequential combination of basic matrix factorization technique [15] and a social network based technique. Experimental results show that this approach excelled the existing basic matrix factorization based recommendation techniques. However, the feature vectors of direct neighbors of user u affect the ratings of u instead of affecting the feature vector of u in this model [10]. And also, this model doesn't address trust propagation. Although social network is integrated, real world recommendations are not reflected in this model. Furthermore, this model's interoperability is difficult as two sets of dissimilar feature vectors is considered.
In recent years, there have been many developments in deep neural networks for graph data, especially the social network data [22]. These are known as Graph Neural Networks (GNNs). Works like [23][24][25] have been proposed to learn meaningful insights and representations for graph data. The main idea in these works is to use neural networks for aggregating features from local graph neighborhoods iteratively. Some of these models use graph neural networks. DANSER [17] is one of the most recent algorithms that uses dual graph attention networks to learn representations for two-fold social effects, where one is modeled by a user-specific attention weight and the other is modeled by a dynamic and context-aware attention weight[17]. There are some other social recommender models. Of them, SocialMF [10], SocialFD [11], and GraphRec [12] are focused in this paper.

3. Models used in this paper

Most of the conventional recommender system algorithms do not consider the social relations among the users in a network. With the increasing usage of social networking applications, incorporating this information into recommendation systems has also become increasingly important. We compared the performance of some baseline algorithms, GraphRec[12], RSTE[8], SoRec[9], SoReg[33], Singular value decomposition (SVD)[14] based RS, SocialFD[11], and SocialMF[10]. Figure 2 depicts the performance of these algorithms on standard recommendation datasets in terms of RMSE and MAE.

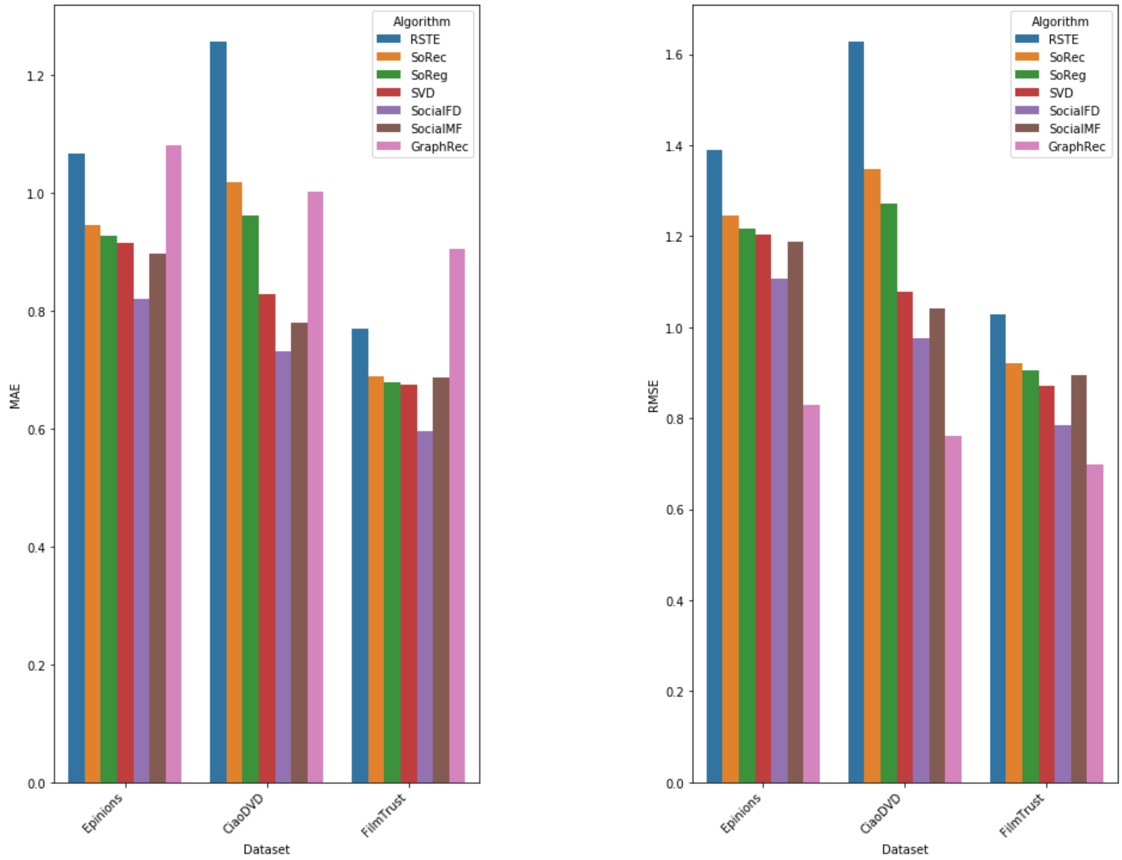

Figure 2: Performance comparison of various recommender algorithms.

These results show average RMSE and MAE from k-Fold cross validation. These algorithms show better performance with learning rates in the range 0.001 to 0.01. The performance of social algorithms is dependent on a social regularization parameter (which is given as a hyperparameter).
Of these seven algorithms, SocialMF [10], SocialFD [11], and GraphRec [12] performed better in terms of RMSE and MAE. These algorithms also include social relations and use trust propagation in the recommendation process, and are usually used as baselines in new trust-based recommender systems.

3.1. SocialMF Model

Jamali and Ester proposed this method in [10]. This model incorporates propagation of trust into matrix factorization for recommending a product/an item in social networks and is closely related to STE model[8]. This model addresses trust transitivity in social networks i.e., this model considers propagation of trust. From the graphical representation of SocialMF model in Figure 3 [10], it is evident that feature vector of a user is dependent on feature vectors of user's direct neighbor. This is a recursive dependence, i.e., feature vector of direct neighbor depends on feature vectors of his/her direct neighbors.
In baseline matrix factorization model [15] and STE model [8], features are learned from only observed ratings. However, in real world social networks, most of the users only participate in social network but do not express ratings to items. This makes it hard to learn feature vectors from observed ratings. SocialMF model handles these users by learning to tune the latent features of these users close to their neighbors. Hence, even if user does not express any ratings, the feature vectors are learned in a way that these are close to feature vectors of their neighbors. As the learned features are typically based on the retained observed ratings, the evaluation of these learned features for users who haven't expressed ratings is difficult.
In a social network, some users actively participate in rating a product or writing a review, but most of the users express very few ratings. These users are called cold-start users. This algorithm has shown improved performance on cold-start users compared to the STE [8] model. However, the SocialMF model has higher cost in calculation of social factor and its gradients against user and item feature vectors.

3.2. SocialFD Model

In this sub-section, Social Recommender that combines Factorization and Distance metric learning, also called SocialFD [11] model is discussed. Yu et. al. proposed this model to make recommendations more reliable. This model is inspired by the concept "distance reflects likability." With the success of distance metric learning in classification tasks [18], [19] Yu et. al. integrated distance metric with matrix factorization in this model [11]. The main idea of distance metric learning is to "learn a desired distance metric that can make data points with the same class label closer and discriminate data points in different sets with larger distance" [11]. SocialFD model, on the contrary, tries to minimize the distance between each user and his/her friends and items that are rated positively. Also, this algorithm maximizes the distance between users and items rated negatively.

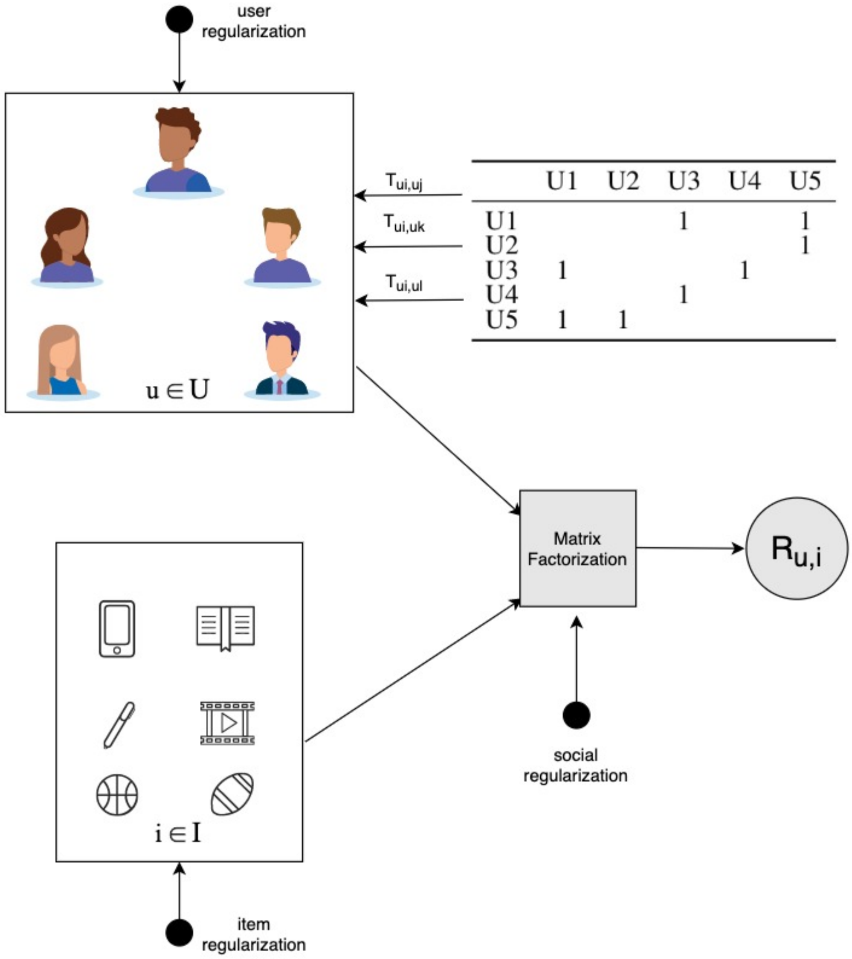

Figure 3: Graphical representation of SocialMF model[10].

The trust propagation in SocialFD model plays an important role and is similar to that of the collaborative filtering model [10]. Given the information of users' likes and friends where user is denoted by *u*, item by *i*, and friends by *k*, SocialFD also pulls user *k* and item *i* relatively closer in addition to pulling user *u* and item *i* closer. The sparsity problem of user *k* can be overcome by recommending user *u*'s preferred items. Likewise, SocialFD model keeps user *u* away from disliked item *j*, pushing item *j* far from user *k*. Additionally, SocialFD algorithm decreases the distance between users with similar interests or indirect connections.
One major drawback with the matrix factorization in general is that it is hard to combine a well-trained vector. On contrast, SocialFD model does flexible inclusion of the ready-made representation of additional knowledge. All the assumptions till now were ratings and social network connections. However, in real-time, user profiles contain huge amounts of texts. These texts can also be accumulated to enhance the quality of recommendations [20][21].
The graphical illustration of SocialFD model can be seen in Figure 4 [11]. In the figure, user and items are represented using circles and crosses respectively. These representations are represented in low-dimensional space in the figure. Closer the two symbols are, higher the probability that user prefers that item or trusts another user. Mahalanobis distance is used in this model and is calculated product of latent features difference and the distance metric. At training stage of the model, constraints are imposed such that users and preferred items/ friends are closer and distant from disliked items/users. The ratings and social connections help model to determine the positions of users and items. i.e., if user has expressed only few ratings, his/her social connections/relations can help recommending items to user. These obtained latent features are interpreted as coordinates and the distance is used to generate meaningful recommendations.

3.3. GraphRec Model

The GraphRec Model consists of three components: user modeling, item modeling and rating prediction [12]. In user modeling, the latent factors users are learned by the model.There are two aggregations in this component, item aggregation and social aggregation. The item aggregation helps in learning item-space user latent factor from user-item ratings data. This is learned by considering the items that user *u* has interacted with and the opinions i.e. ratings that *u* has on these items. In social aggregation, social-space user latent factor is learned from the social data. According to social correlation theories[26][27], users opinions towards an item or preferences are either influenced or similar to their direct friends in social networks. In social aggregation of GraphRec,the authors proposed social-space user latent factors, which is to aggregate the item-space user latent factors of neighboring users from the social graph, to incorporate the social correlation theories. Combining the item-space latent factor and social-space latent factor, the total user latent factor is learned.
The next component is item modeling. In this component, the item latent factor can be learned by user aggregation. User aggregation associates item *i* with users that interacted with *i* and their opinions. These opinions or ratings from different users help in capturing the features of same item in different ways provided by users. This helps in modeling item latent factors.

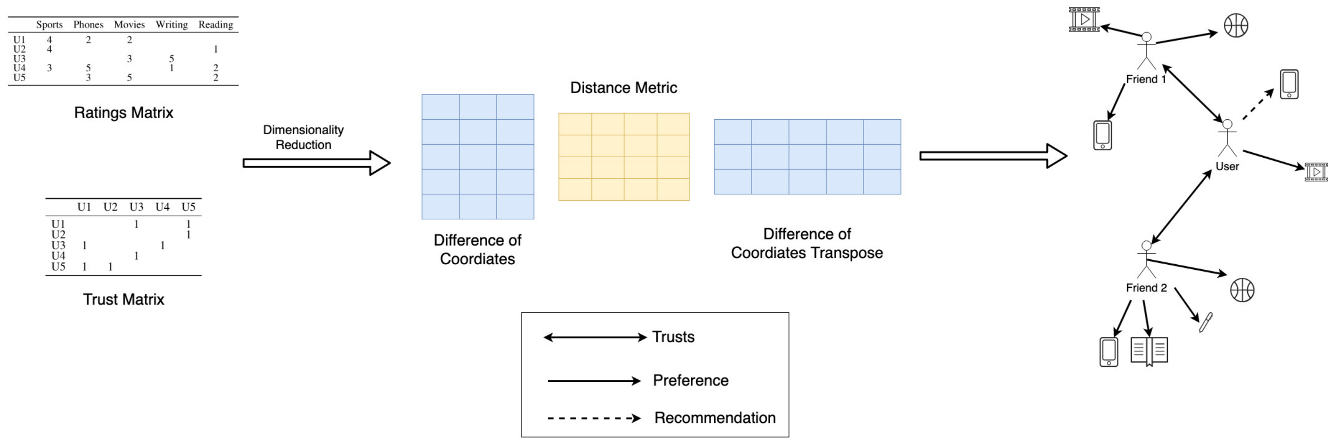

Figure 4: Graphical Representation of SocialFD model [11].

Finally in the third component, the model parameters are learned ratings are predicted using GraphRec model. To learn the model parameters, the authors utilized the most commonly used objective function which is formulated as:

$$Loss = \frac{1}{2|O|}\sum_{i,j\varepsilon O}(r'_{ij} - r_{ij})^2 \quad (1)$$

where O is the number of total observed ratings, r_{ij} is rating given by user i to item j.
For optimization of objective function, the authors used RMSprop defined in [28] rather than the vanilla Stochastic Gradient Descent (SGD). This RMSprop, each time,selects training instance randomly and updates each model parameter towards the direction of its negative gradient [12]. The three embedding item, user and opinions are initialized randomly and learned during the training stage.
The layers in the model are

$$g_1 = [h_i \oplus z_j] \quad (2)$$

$$g_2 = \sigma(W_2 \cdot g_1 + b_2) \quad (3)$$

$$g_l = \sigma(W_l \cdot g_{l-1} + b_l) \quad (4)$$

rating is predicted using:

$$r'_{ij} = w^T \cdot g_{l-1} \quad (5)$$

where l is the index of a hidden layer, and r'_{ij} is the predicted rating from u_i to v_j
To avoid overfitting, a persistent problem in optimization of deep neural networks, dropout [?] - a regularization technique for deep neural networks, is utilized. While testing, the dropout regularization is disabled which allows the usage of whole network.

4. Experiments

This section gives an overview of the experiments and datasets used in these experiments. Also included are the evaluation metrics used to evaluate these experiments done with SocialMF, SocialFD and GraphRec models.

4.1. Approach

In this paper, our goal is to study how the three representative social recommenders, i.e., SocialMF, SocialFD, and GraphRec, leverage social network information. Towards this goal, we make the hypothesis that “if the recommender system truly captures the social network information, making perturbations to the social network should have a significant impact on the performance of these systems.” We will study these algorithms on four datasets and three ways to perturb the social network information, as described in the rest of the section.

4.2. Datasets

The major bottleneck in research of social network based recommender systems is the lack of publicly available social rating network datasets. These models were experimented with four datasets. Epinions.com is one of the popular publicly available social rating network datasets. For experimentation with these models, we used a version of Epinions dataset published by authors of [21]. On average, each user has 8 expressed ratings and have 7 direct neighbors. The next dataset we used is a version of CiaoDVD by authors of [30]. This is a smaller one compared to Epinions dataset. Another relevant dataset we used is FilmTrust [31]. FilmTrust is the smallest dataset used in experimentation crawled from FilmTrust website in 2011.
We have created an additional dataset to experiment with these models. This dataset is an extension of TwitterEgo dataset by authors of [29]. The basic dataset consists of social circles from Twitter data which was crawled from public sources. Using Tweepy API, we extracted

tweets to which each of the users reacted. Using the tweet IDs, the tweets to which users reacted are rated as 1 and the tweets from other users to which the users did not react were rated as 0. Finally, we created a ratings dataset for the users and tweets. We generated the social relations dataset using the social circles from the original dataset. We divided the data into test and train using stratified split technique. Table 3 shows the compositions of each of the datasets used for experimentation.

Table 3: Data Statistics

Dataset	Users	Items	Ratings	Relations	Rating Scale
CiaoDVD	7,375	105,114	284,086	111,781	1.0 – 5.0
Epinions	40,163	139,738	664,823	487,183	1.0 – 5.0
FilmTrust	1,508	2,071	35,497	1,853	1.0 – 5.0
TwitterEgo	10,419	177,558	367,868	566,822	0.0 – 1.0

4.3. Social Network Modifications

Taking the original datasets, we modified the social trust data in each of the four datasets to fit for our experiments. This sub-section discusses the modifications we made to the datasets and an example for each.

4.3.1. There is no trust between any users

For this experiment, we modified the social data by removing all the trust statements and providing number of trust statements as 0. i.e. the social network part of data is not considered. The sample social trust matrix would look as in table 4.

Table 4: Sample Social Trust Matrix for Users have no trust

	U1	U2	U3	U4	U5
U1					
U2					
U3					
U4					
U5					

4.3.2. Users trust only themselves

In this experiment, the social data is modified in such a way that user u only trusts or friends with u and not anyone else. From the example from Figure 1, the modified social trust matrix would look as in Table 5.

Table 5: Sample Social Trust Matrix for Users trust only themselves

	U1	U2	U3	U4	U5
U1	1				
U2		1			
U3			1		
U4				1	
U5					1

4.3.3. Users trusts everyone else except themselves

In this experiment, the social data is modified in such a way that user u_1 only trusts or friends with all other users in the network U. From the example from Figure 1, the modified social trust matrix would look as in Table 6.

Table 6: Sample Social Trust Matrix for Users trusts everyone else except themselves

	U1	U2	U3	U4	U5
U1		1	1	1	1
U2	1		1	1	1
U3	1	1		1	1
U4	1	1	1		1
U5	1	1	1	1	

4.4. Evaluation Metrics

In these experiments, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) are chosen to evaluate the quality of recommendations produced by these models.
RMSE is calculated using:

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{u,i}(r_{ui} - r'_{ui})^2} \tag{6}$$

and MAE is defined by:

$$MAE = \frac{1}{n}\Sigma_{u,i}|r_{ui} - r'_{ui}| \tag{7}$$

where n denotes number of ratings in test set, r_{ui} is actual rating and r'_{ui} is the predicted rating. Lower MAE and lower RMSE indicate that the missing ratings are predicted more accurately.

4.5. Hyperparameter Tuning

We tuned the hyperparameters such as learning rate, regularization factors (for users, items, and social circles), number of factors, batch size, and number of epochs. All three algorithms are highly dependent on learning rate. The performance of SocialMF and SocialFD algorithms is dependent on social regularization parameter. At lower values of social regularization parameter, the performance of SocialMF and SocialFD algorithms is similar to that of the baseline matrix factorization algorithms.

5. Results

In this section, the results of SocialMF, SocialFD, and GraphRec models are reported for each dataset and compared using the RMSE and MAE evaluation metrics. The SocialMF model results are shown in Table 7, SocialFD model results are in Table 8, and GraphRec results in Table 9.
It is important to remark here that the code used for implementation of SocialMF algorithm is a modified version provided by the authors of SocialFD [11]. This version has an implementation difference and is mostly in terms of the training phase where the authors of SocialFD have used SGD to obtain the local minimum. The underlying graphical model is still the same. Results of the SocialMF model can be seen in Table 7

Table 7: SocialMF model results on 4 datasets with modified social trust information

Experiment	Metrics	CiaoDVD	Epinions	FilmTrust	TwitterEgo
User trusts friends	RMSE	1.0269	1.1044	0.8429	0.1015
	MAE	0.7718	0.8683	0.6389	0.0234
User doesn’t trust anyone	RMSE	1.0248	1.1455	0.8419	0.0998
	MAE	0.7685	0.8425	0.6344	0.0160
Users trust only themselves	RMSE	1.0259	1.1037	0.8399	0.1129
	MAE	0.7817	0.8702	0.6389	0.0215
User trusts everyone except themselves	RMSE	1.0269	1.1549	0.8521	0.0973
	MAE	0.7785	0.8524	0.6449	0.0159

NOTE: Because of the usage of random seed and gradient descent in these experiments, up to two percent difference in RMSE and MAE is negligible. Also, RMSE and MAE of TwitterEgo dataset are low compared to other datasets because its rating scale is binary, i.e., 0 or 1.
The difference of RMSE between experiments “users trust friends” and “user doesn’t trust anyone” for Epinions increases to 4%. This increase is slightly higher than our threshold, but the MAE is lower. As graph density changes for each experiment, there might be some effect of this change on the SocialMF model.
Although both SVD-based RS [14] and SocialMF utilize matrix factorization in recommending content or products to users, there are some differences in implementation. In SocialMF, there is an additional step to update the user u’s latent factors based on the user u’s neighbors. This implies that only user matrix is updated in SocialMF and the social trust matrix remains the same. In case of “user doesn’t trust anyone,” the user matrix does not get updated and the original user matrix is retained. However, compared to SVD-based RS, the errors decrease significantly. This might be due to the use of user and item regularization terms. More exploration is needed in understanding how transitivity of trust is affected by changing the social trust information.
Based on how the inference is influenced by the trust matrix, it is hard to draw any formal conclusions about the impact of the trust matrix on the recommendation. This could be because of multiple reasons. First, based on the fact that social network metrics for these graphs are varied, it may not have anything to do with the social network structure itself. It is, however, possible that the dataset itself is not one in which social influence plays a role. We need further experimentation to verify this formally. Another reason could be that the modified trust matrices somehow “balanced” out the user latent vectors or the loss function. Details for each modified social graph are below.
In SocialMF, the user latent feature vector is the weighted average of the latent feature vectors of adjacent users in the social graph. The implication of the changed social networks here is that the “user doesn’t trust anyone” experiment maintains the original latent feature

vectors and recommendations are solely based on users that are similar in terms of their ratings. The behavior of "user doesn't trust anyone" is therefore expected to be similar to SVD. However, the presence of the social factor (which is setup as a hyperparameter) might have scaled the recommendation scores.
Extending the argument to the "user trusts only themselves" social graph, the user latent vectors are weighted by their own ratings further. We expected that it might have reduced the influence of similar users in the traditional sense to have lesser influence on a user's recommendation. For the "user trusts everyone except themselves" social graph, the user latent vectors are influenced by the average of the latent vectors of other users. For "user trusts everyone except themselves" social graph, we expected that the average latent vectors might pollute the user similarity with respect to ratings. All of these social graphs were expected to harm the performance of SocialMF. However, the experimental results did not show a significant change in performance.
A key reason for this could be the social trust parameter that tunes the influence of the social network on the recommendation. The authors of SocialMF [10] use a Gaussian prior to determine this factor, which plays a crucial role in their loss function. For our trivial social networks, the priors do not hold. The results, however, do highlight the need to explore the role of the social network further.
In the future, we propose to run further experimentation with random trust matrices to draw more formal conclusions about the role of the social network in the SocialMF algorithm.

Table 8: SocialFD model results on 4 datasets with modified social trust information

Experiment	Metrics	CiaoDVD	Epinions	FilmTrust	TwitterEgo
User trusts friends	RMSE	0.9645	1.0458	0.7806	0.0221
	MAE	0.7261	0.7932	0.5948	0.0159
User doesn't trust anyone	RMSE	0.9641	1.0456	0.7803	0.0161
	MAE	0.7260	0.7906	0.5947	0.0069
Users trust only themselves	RMSE	0.9594	1.0485	0.7709	0.0198
	MAE	0.7368	0.7899	0.6008	0.0079
User trusts everyone except themselves	RMSE	0.9590	1.0548	0.7702	0.0159
	MAE	0.7160	0.7897	0.5893	0.0068

From the Tables 7 and 8, it can be inferred that the SocialFD algorithm in general performs better than the SocialMF algorithm consistently for these diverse datasets. This indicates that distance metrics may have an important role in social recommendations. However, when we compare these models between "user trusts friends" and "user doesn't trust anyone", there is no significant change in the RMSE and MAE. In some cases, there seems to be a marginal improvement without trust information. Looking back at our hypothesis in Section 4.1, the results indicate that there is a need for deeper exploration on these lines.

Table 9: GraphRec model results on 4 datasets with and without social trust information

Experiment	Metrics	CiaoDVD	Epinions	FilmTrust	TwitterEgo
User trusts friends	RMSE	1.0022	1.0818	0.9057	0.0209
	MAE	0.7611	0.8299	0.6970	0.0149
User doesn't trust anyone	RMSE	0.9989	1.0786	0.9051	0.0194
	MAE	0.7645	0.8255	0.6921	0.0140
Users trust only themselves	RMSE	1.0342	1.1008	0.8975	0.0198
	MAE	0.7903	0.8382	0.7108	0.0151
User trusts everyone except themselves	RMSE	0.9689	1.0983	0.9005	0.0182
	MAE	0.7945	0.8356	0.6891	0.0137

GraphRec showed improved performance while considering social trust information when compared to other models like SocialMF [10], SoRec[9], etc. Although GraphRec is a more complex deep learning algorithm, we did not observe much difference in RMSE and MAE as compared to SocialFD. In some cases, these errors increased significantly. For example, for our TwitterEgo dataset, the MAE increased significantly and for the FilmTrust dataset both metrics increased. We also observed that for the same model, performance improves in some scenarios and deteriorates in the others. Regardless, the difference in performance is very little. This highlights the fact that deep learning techniques cannot guarantee a better performance even with huge datasets such as Epinions and TwitterEgo. It is particularly interesting that in most cases, the trivial social trust datasets perform better than the default "user trusts friends" datasets.
All of the above experiments indicate the following key observations. Comparing social recommendation algorithm with trust information indicates that SocialFD is a superior algorithm for all the datasets and social trust data compared to SocialMF. However, from our experiments, social trust information does not significantly improve the performance of these representative models on diverse datasets. Also, we understand that superior performance of SocialFD algorithm has something to do with the utilization of distance metric. Further study is needed in the direction of incorporating distance metric learning into social recommender systems. The computational overhead and complexity of deep learning models may not make a significant difference to the performance. It also highlights the importance of studying the interpretability of deep learning social recommender systems in general.

6. Conclusion and future work

Recommender systems are powerful tools that filter and recommend content/information relevant to a given user. With the advent of social networks, it has become very important to utilize the data on social ties between in a social network to recommend a product to the users who expressed a few ratings. In this paper, we explored three such social recommender models, SocialMF, SocialFD, and GraphRec. SocialMF is a model that incorporates trust propagation into matrix factorization, SocialFD model uses distance metric learning in addition to matrix factorization, and GraphRec is a model that uses attenttion-based graph neural networks.

Experiments on 4 real life datasets, CiaoDVD, Epinions, FilmTrust, and TwitterEgo, show that these algorithms outperform the conventional collaborative filtering algorithms as well as the previously developed social recommender systems. Of these three, the SocialFD model performs better than the SocialMF model with inclusion of trust. At lower social parameter values, these models' performance is similar to the performance of collaborative filtering algorithms. However, when social trust factor is not given, these models show similar performance compared to the models that contain social trust parameter. From these experiments, it can be seen that there is less conclusive evidence that social recommender systems are influenced by social trust data. The experiments highlight the need to explore further to gain better understanding of the role of social networks in recommender systems.

This work suggests some interesting directions for future research. These models can be extended further to handle zero and negative trust relations. In general, negative trust, also called distrust, gives more information about a user than positive opinions. Also, currently, social regularization parameter is given as an input to these models. Future work can help in the development of a model that incorporates automatic tuning of social trust. In real-time, user profiles and item profiles contain huge amounts of text data and other features. These features can also be accumulated into the recommender system to enhance the quality of recommendations. In the future, we would like to explore a dataset where social network is explicitly synthesized to have an impact on recommendation and repeat these experiments on that synthetic dataset.

Acknowledgement

This research is partially supported by Charles W. Davidson College of Engineering at San Jose State University.

References

[1] Ricci, F., Rokach, L. & Shapira, B. Introduction to Recommender Systems Handbook. *Recommender Systems Handbook*. pp. 1-35 (2011)
[2] Goldberg, D., Nichols, D., Oki, B. & Terry, D. Using Collaborative Filtering to Weave an Information Tapestry. *Commun. ACM*. **35**, 61-70 (1992,12)
[3] Qian, X., Feng, H., Zhao, G. & Mei, T. Personalized Recommendation Combining User Interest and Social Circle. *IEEE Transactions On Knowledge And Data Engineering*. **26**, 1763-1777 (2014)
[4] Jamali, M. & Ester, M. TrustWalker: A Random Walk Model for Combining Trust-Based and Item-Based Recommendation. *Proceedings Of The 15th ACM SIGKDD International Conference On Knowledge Discovery And Data Mining*. pp. 397-406 (2009)
[5] Massa, P. & Avesani, P. Trust-Aware Recommender Systems. (Association for Computing Machinery,2007)
[6] Ziegler, C. Towards decentralized recommender systems. (University of Freiburg,2005)
[7] Golbeck, J. Computing and Applying Trust in Web-based Social Network. (University of Maryland,2005)
[8] Ma, H., King, I. & Lyu, M. Learning to Recommend with Social Trust Ensemble. (Association for Computing Machinery,2009)
[9] Ma, H., Yang, H., Lyu, M. & King, I. SoRec: Social Recommendation Using Probabilistic Matrix Factorization. *Proceedings Of The 17th ACM Conference On Information And Knowledge Management*. pp. 931-940 (2008)
[10] Jamali, M. & Ester, M. A Matrix Factorization Technique with Trust Propagation for Recommendation in Social Networks. *Proceedings Of The Fourth ACM Conference On Recommender Systems*. pp. 135-142 (2010)
[11] Yu, J., Gao, M., Rong, W., Song, Y. & Xiong, Q. A Social Recommender Based on Factorization and Distance Metric Learning. *IEEE Access*. **5** pp. 21557-21566 (2017)
[12] Fan, W., Ma, Y., Li, Q., He, Y., Zhao, E., Tang, J. & Yin, D. Graph Neural Networks for Social Recommendation. (Association for Computing Machinery,2019)
[13] Koren, Y. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. (Association for Computing Machinery,2008)
[14] Koren, Y., Bell, R. & Volinsky, C. Matrix Factorization Techniques for Recommender Systems. *Computer*. **42**, 30-37 (2009)
[15] Richardson, M. & Domingos, P. Mining Knowledge-Sharing Sites for Viral Marketing. (Association for Computing Machinery,2002)
[16] Levien, R. Attack-Resistant Trust Metrics. *Computing With Social Trust*. pp. 121-132 (2009)
[17] Wu, Q., Zhang, H., Gao, X., He, P., Weng, P., Gao, H. & Chen, G. Dual Graph Attention Networks for Deep Latent Representation of Multifaceted Social Effects in Recommender Systems. (Association for Computing Machinery,2019)
[18] Goldberger, J., Hinton, G., Roweis, S. & Salakhutdinov, R. Neighbourhood Components Analysis. *Advances In Neural Information Processing Systems 17*. pp. 513-520 (2005)
[19] Weinberger, K. & Saul, L. Distance Metric Learning for Large Margin Nearest Neighbor Classification. *Journal Of Machine Learning Research*. **10**, 207-244 (2009), http://jmlr.org/papers/v10/weinberger09a.html
[20] Musto, C. Enhanced Vector Space Models for Content-Based Recommender Systems. (Association for Computing Machinery,2010)
[21] Semeraro, G., Lops, P., Basile, P. & Gemmis, M. Knowledge Infusion into Content-Based Recommender Systems. (Association for Computing Machinery,2009)
[22] Kipf, T. & Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. (2017)
[23] Defferrard, M., Bresson, X. & Vandergheynst, P. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. (Curran Associates Inc.,2016)
[24] Hamilton, W., Ying, R. & Leskovec, J. Inductive Representation Learning on Large Graphs. (Curran Associates Inc.,2017)
[25] Ma, Y., Wang, S., Aggarwal, C., Yin, D. & Tang, J. Multi-dimensional graph convolutional networks. *SIAM International Conference On Data Mining, SDM 2019*. pp. 657-665 (2019), 19th SIAM International Conference on Data Mining, SDM 2019
[26] MARSDEN, P. & FRIEDKIN, N. Network Studies of Social Influence. *Sociological Methods Research*. **22**, 127-151 (1993)
[27] McPherson, M., Smith-Lovin, L. & Cook, J. Birds of a Feather: Homophily in Social Networks. *Review Of Sociology*. **27** pp. 415-444 (2001)
[28] Tieleman, T. & Hinton, G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks For Machine Learning*. **4**, 26-31 (2012)
[29] Leskovec, J. & Mcauley, J. Learning to Discover Social Circles in Ego Networks. *Advances In Neural Information Processing Systems 25*. pp. 539-547 (2012)
[30] Zhao, G., Qian, X. & Xie, X. User-Service Rating Prediction by Exploring Social Users' Rating Behaviors. *IEEE Transactions On Multimedia*. **18**, 496-506 (2016)
[31] Tang, J., Gao, H. & Liu, H. MTrust: Discerning Multi-Faceted Trust in a Connected World. *Proceedings Of The Fifth ACM International Conference On Web Search And Data Mining*. pp. 93-102 (2012)
[32] Guo, G., Zhang, J. & Yorke-Smith, N. A Novel Bayesian Similarity Measure for Recommender Systems. (AAAI Press,2013)
[33] Ma, H., Zhou, D., Liu, C., Lyu, M. & King, I. Recommender systems with social regularization. *Proceedings Of The Fourth ACM International Conference On Web Search And Data Mining*. pp. 287-296 (2011)