Exploiting Implicit Trust and Geo-social Network for Recommendation

1st Feiyang Li
School of Computer Science and Technology
University of Science and Technology of China, P.R.china
mmmwhy@mail.ustc.edu.cn

2ndKai Han
School of Computer Science and Technology
Suzhou Institute for Advanced Study
University of Science and Technology of China, P.R.china
hankai@ustc.edu.cn

3rdYue Li
School of Mathematics and Physics
North China Electric Power University, Beijing, 102206, China
mmmwhy@hust.edu.cn

4th Jiahao Zhang
School of Computer Science and Technology
University of Science and Technology of China, P.R.china
jhcheung@mail.ustc.edu.cn

***Abstract*—Recommender system (RS) seeks to predict the rating or preference a user would give to an item, this system often relies on collaborative filtering (CF). CF suffers from the problems of data sparsity, cold start and location insensitive. Existing RSs do not consider the spatial extent of users, we analyze the users' location data from four four commercial websites, and conclude that people with close social relationships prefer to purchase in places that are also physically close. State-of-the-art recommendation algorithm TrustSVD extends RS with social trust information, we propose Trust-location SVD (TLSVD) by incorporating the location information and implicit trust into TrustSVD. The improved TLSVD helps to quantitatively analyze the spatial closeness and preference similarity between users. Experimental results indicate that the accuracy of our method is better than other multiple counterparts', especially when active users have location information or few ratings.**

***Index Terms*—Recommender system, Location-based social networks, Collaborative filtering, Implicit trust**

I. Introduction

Modern consumers are faced with a variety of choices, retailers offer a wide selection of products to meet all kinds of special needs and tastes. It's not easy to match consumers with most appropriate products, this emphasizes the importance of the recommender system (RS), which provides personalized recommendations for products that are suitable for users' tastes. Collaborative filtering (CF) is a widely accepted recommendation approach built upon the assumption that users are likely to prefer the same items which the similar users prefer [19]. For example, if user A and user B have the same rating habits, A is more likely to agree with B on an unknown issue than a randomly chosen person. Such models have great significance on many commercial websites like Yelp[1], Foursquare[2], Dianping[3] etc. The personalized advice that RS provides greatly increase the likelihood of a customer making a purchase compared to an unpersonalized one.

[1]https://www.yelp.com
[2]https://foursquare.com
[3]http://www.dianping.com

☆☆☆☆☆ **San Francisco, CA** **38** friends I had been looking for the best paella in San Francisco, and Paellas and Cos did not disappoint! Their paella is rich and flavorful and cooked to perfection.	Been here 5+ times [4] **West Village, NYC** Followers: **319** The Sabich Sandwich is my favorite vegetarian dish in NYC. It's loaded with eggplant, hard-boiled egg, hummus, and more but the amba sauce is the key.

Fig. 1. Two sample reviews

However, CF has three well-known problems: data sparsity, cold start [3] and location insensitive. The densities of available ratings in many commercial websites are often less than 1% [9], the second problem indicates that new users only give few or no ratings, the last one represents that many existing RSs ignore the importance of location information of users/items. These problems severely reduce the effectiveness of the RSs in modeling user preference and thus significantly reduce accuracy of prediction on the rating score of unknown items. To solve these problems, many researchers [5], [7], [8] gradually introduce trust relation into RSs, these methods furtherly improve prediction accuracy based on the theory that friends often influence each other by offering proposals. The most advanced trust-based model is TrustSVD [7], this model takes the ratings and explicit trust into account when recommending active users. Even though explicit trust is so sparse, TrustSVD can be far more advanced than other state-of-the-art methods that only based on user-item ratings.

As shown in **Fig.1**, two real comments from local service websites Yelp and Foursquare separately, the rating(1 star to 5 stars) shows user's overall satisfaction degree of an evaluation

[4]Actually, the Foursquare dataset contains ratings from users.

object, the evaluation contents not only indicate the preference of users but also contain location and social relationship. Studies in [10] shows that good friends with close locations prefer to purchase in the same place. Thus, we enhance the spatial-unaware TrustSVD by incorporating implicit trust and location information.

Summary of Contributions. This work makes the following main contributions.

1. Although the user can designate other users as trusted neighbors and indicate their degree of trust, publicly available datasets contain only binary values due to privacy concerns. TrustSVD relies on social trust relationships which are sparse and binary (i.e, trust links), in order to overcome the above disadvantages, we improve TrustSVD by incorporating the influence of implicit trust.
2. Furthermore, we propose a spatial-based recommendation method Trust-location SVD (TLSVD) builds on top of above improvements. To our knowledge, we are the first one to introduce location infomation into TrustSVD. The experimental results on four websites datasets show that our approach exceeds the state-of-the-art recommendation algorithm, especially when users have location information.

Organization. We review several major approaches of RS and spatial community search in Section II, then analyze implicit-trust and spatial information in Section III. Section IV presents our novel recommendation algorithm: TLSVD. The results of experiments will be presented in Section V, followed by the conclusions and future work in Section VI.

II. Related Work

In this section, we review several major recommendation methods, especially for CF, there are two types of CF: memory-based and model-based. Memory-based method is more popular and widely used in commercial RS [11], the most typical examples of memory-based collaborative filtering are user-based [12], [13] and item-based approach [14]. User-based method predicts the ratings according to the similar users, as illustrated in **Fig.2(a)**, the first user and the third user both like apple and watermelon, they have the similar preference. Base on the fact that the first user like strawberry, so we can recommend strawberry to the third user. Item-based approach bases on the similarity of items, as illustrated in **Fig.2(b)**, users who like apple usually like strawberry as well. We call these two fruits as similar items, so we can recommend strawberry to the user who like apple. About the model-based approach, models are developed by using different machine learning algorithms to predict users' ratings of unrated items.

Recommendation models above all are based on the assumption that users are independent and identically distributed (**i.i.d**), this is not consistent with the fact that we usually ask our friends for advice. Based on this fact, many researchers have started introducing trust into RS. Hao et al. proposed SoRec in [5] by considering the constraint of social relationship, his idea is to share a common user-feature matrix regularized by trust and ratings. Guo and Zhang presented a

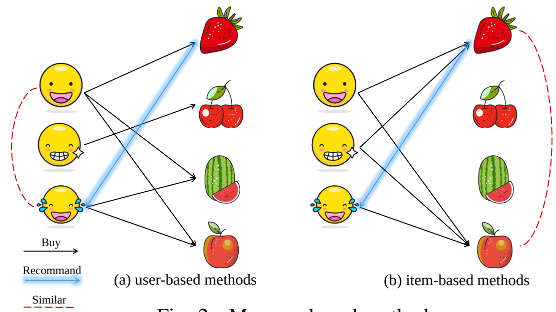

Fig. 2. Memory-based method

new model TrustSVD in [7] on the top of SVD++ by combing trust relation, they decomposed trust matrix into two user feature matrices.

On the one hand, many works show that a matrix decomposition method normalized by trust better than the one without trust. On the other hand, with the emergence of local based services, such as Yelp and Foursquare. Spatial community search has also received plently of attention [15]–[17], users are associated with each other through location information in these geo-social networks. Fang proposed a concept called spatial-aware community (SAC) which is a social unit of any size that shares common characteristics [18], he emphasized the significance of SAC in event recommendation and social marketing. In order to improve the accuracy of predictions in geo-social networks, it is meaningful to propose a new model to combine SAC with TrustSVD.

III. Implicit-trust and Spatial Analysis

In this section, we introduce the concepts of implicit trust and spatial relationship. **TABLE I** lists frequently used notations.

TABLE I
Notations and meanings

Notation	Meaning
b_u, b_i	biases of user u and item i
$\mu, r_{u,i}$	item global average rating, rating given by user u on item i
P, Q	user-feature matrix and item-feature matrix
R, T	user-item rating matrix, user-user trust matrix
S, L	user-user latent trust matrix, user-user SAC relationship matrix
I_u, U_i	set of items rated by user u, set of users who rate item i
T_u, T_v^+	set of users trusted by user u, set of users who trust user v
L_u, L_u^+	set of users in user u's SAC, set of user whose SAC include u
W, y_i	trustee-feature matrix, implict influence of items rated by u
w_v, m_x	user feature vector of user v,x explicit and implicit trusted by u
d_z	user latent feature vector of user z whose SAC include u

Implicit trust. User similarity is used as a distance metric to measure the closeness of two users, user-based and item-based approaches often use the Pearson correlation coefficient (PCC) algorithm as similarity computation methods.

$$s_{u,v} = \frac{\sum_{i \in I_{u,v}} (r_{u,i} - \overline{r}_u)(r_{v,i} - \overline{r}_v)}{\sqrt{\sum_{i \in I_{u,v}} (r_{u,i} - \overline{r}_u)^2} \sqrt{\sum_{i \in I_{u,v}} (r_{v,i} - \overline{r}_v)^2}} \quad (1)$$

Where $s_{u,v} \in [-1, 1]$ is the similarity between user u and user v, $I_{u,v}$ is the set of items rated by both u and v. Papagelis et al. used PCC to define the implict trust between users [19], due to the sparsity of trust information, it's a novel approach to extend recommendation model with latent trust.

SAC. The concept of SAC was proposed by Fang in [18], SAC is a high structure and spatial cohesiveness community. With the location information and social network of users, we can get a geo-social unit that shares common characteristics, the detail calculation procedure is given in section IV.

Four real-world datasets are used in our analysis and experiments: Yelp[5], Foursquare[6], Dianping[7], Gowalla[8]. All of the four datasets contain item ratings, social networks and locations of users or items, the spatial data in Yelp and Dianping contains shop location and checkin log. Gowalla only has checkin log, we use function *ln* to convert the checkin log into user preference, the dataset statistics are illustrated in **TABLE II**.

IV. TLSVD: A Location-based Model

A. Problem Definition

Based on user-item rating matrix, user-user location relation, user-user explicit and implicit trust matrix, the problem in this paper is to predict the rating that a user would give to an unknown item.

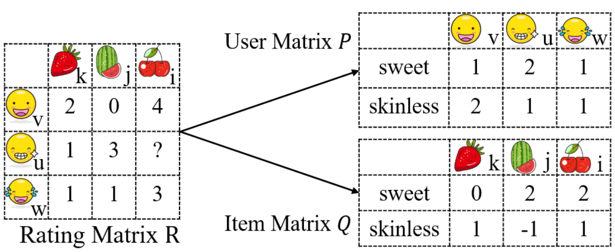

Fig. 3. The decomposition of user-item rating matrix

Let us first consider the typical matrix factorization RS in **Fig.3**, the key of matrix factorization is to find two low-dimension matrices: user-feature matrix P and item-feature matrix Q. The two matrices can adequately recover the rating matrix R, i.e $R \approx PQ^T$. The assumption of matrix factorization is that users and items can be characterized by some features like preferring sweet or skinless. Therefore, the rating on cherry for grin user can be predicted by inner product of $\hat{r}_{u,i} = q_i^T p_u$. The job of RS is to predict the rating $\hat{r}$ as accurate as possible, so we need loss function to quantify the amount by which the predictions deviate from the actual values, λ is used to control model complexity and avoid overfitting.

$$\min_{q*,p*} \sum_u \sum_{i \in I_u} (q_i^T p_u - r_{u,i})^2 + \lambda\Big(\sum_u ||p_u||_F^2 + \sum_i ||q_i||_F^2\Big)$$

[5]https://www.yelp.com/dataset

[6]Foursquare dataset provided by [6]

[7]Due to licence restrictions, we use spider to collect data and do not share.

[8]Gowalla dataset provided by [4]

Similarly, we use matrix T to describe the directed trust relationship between users, w_v is d-dimensional user latent feature vector of user v whom user u trusts. Respectively, let W denote trustee feature matrix. Thus, we can use $t_{u,v} = w_v^T p_u$ to predict the trust relationship between users, and learn P and D by minimizing the following loss function:

$$\min_{w*,p*} \sum_u \sum_{v \in T_u} (w_v^T p_u - t_{u,v})^2 + \lambda\Big(\sum_u ||p_u||_F^2 + \sum_i ||w_v||_F^2\Big)$$

One of the drawbacks of above equation is that large majority of datasets contain only binary trust values due to privacy concerns, this might decrease trust accuracy between users. As mentioned in section III, many researchers use PCC to measure closeness of two users, we use matrix S to describe this kind of closeness between users, m_x represents the feature vector of user x who subconsciously trusted by user u. Furthermore, we can get the latent trust by calculating $\hat{s}_{u,x} = m_x^T p_u$. Respectively, we define the loss function over the latent trust relationship:

$$\min_{m*,p*} \sum_u \sum_{x \in S_u} (m_x^T p_u - s_{u,x})^2 + \lambda\Big(\sum_u ||p_u||_F^2 + \sum_i ||m_x||_F^2\Big) \tag{2}$$

Meanwhile, the location-based service (LBS) and the personalization recommendation are two important trends in the development of electric commerce. However, many previous researches only emphasize on personalization RS, this paper uses the adjacency matrix L to measure the location relations between users, $l_{u,z}$ presents the extent of spatial relationship between users, we also propose an algorithm SACQuery to quantitatively calculate this parameter in the next subsection.

B. The TLSVD Model

Trust-location SVD (TLSVD) is based on Guo's TrustSVD [7], Guo's work is based on SVD++ which was proposed by Koren [2]. SVD++ proposes a baseline estimation theory to consider the user/item biases and item global average ratings as follows:

$$\hat{r}_{u,i} = b_u + b_i + \mu + q_i^T p_u$$

Normally, we use $b_{u,i}$ to represent $b_u + b_i + \mu$, Koren also provided an opinion about user rated items in the past will implicit influence the rating of unknown items in the future. Consequently, we can use the set of items that user rated to represent user's feature vector, finally use $(p_u + |I_u|^{-\frac{1}{2}} \sum_{j \in I_u} y_j)$ to replace simple p_u [2]. Guo merged the trust relationship into SVD++, and enumerated a theory that user feature vector can be interpreted as the set of users whom he trusts, i.e. $|T_u|^{-\frac{1}{2}} \sum_{v \in T_u} w_v$ [7]. As analyzed in the previous section, we explain that binary and sparsity of trust information will reduce accuracy of prediction. Hence, we use latent trust to enhance TrustSVD model. Similar to TrustSVD, user's eigenvector can be explained by the user whom she latent trusts. Accordingly, a more general prediction equation would be:

TABLE II
STATISTICS OF THE FOUR DATASETS

Statistics	Yelp	Foursquare	Dianping	Gowalla
Users	1,183,326	2,153,469	7,351	407,434
Items	156,639	1,143,090	76,271	2,844,076
Ratings	2,570,642	2,809,580	733,805	*
Social Relations	39,846,890	27,098,472	102,475	4,418,339
Locations (latitude and longitude)	156,639	3,296,559	76,271	36,001,959
Rating Density	0.000014	0.000011	0.0013	*
Ave. Friends Per User	33	12	14	11

$$
\begin{aligned}
\hat{r}_{u,i} =& b_{u,i} + q_i^T\big(p_u + |I_u|^{-\frac{1}{2}} \sum_{i \in I_u} y_i + |T_u|^{-\frac{1}{2}} \sum_{v \in T_u} w_v \\
&+ |S_u|^{-\frac{1}{2}} \sum_{x \in S_u} m_x\big)
\end{aligned} \tag{3}
$$

As usual, we use loss function to get the values of involved parameters.

$$
\begin{aligned}
&\mathcal{L} = \frac{1}{2}\sum_{u}\sum_{i \in I_u}\left(\hat{r}_{u,i} - r_{u,i}\right)^2 \\
&+ \frac{\lambda}{2}\big(\sum_u b_u^2 + \sum_i b_i^2 + \sum_u ||p_u||_F^2 \\
&+ \sum_i ||q_i||_F^2 + \sum_j ||y_j||_F^2 + \sum_u ||w_v||_F^2 + \sum_x ||m_x||_F^2\big)
\end{aligned} \tag{4}
$$

Algorithm 1: SACQuery

Input: G, T, q, k
Output: the user unit of user's SAC: L

```
1  Initialize Q, L ← φ, R ← φ, β ← k;
2  Q.add(q);
3  while |Q| > 0 do
4      p ← Q.push();
5      L.add(p, β/|p,q|);
6      for u ∈ T(p) do
7          if deg_G(u) ≥ k then
8              if |u,q| ≤ |p,q| then
9                  L.add(u, β/|u,q|);
10                 β = β − 1;
11             else if u ∉ R then
12                 Q.add(u);
13                 R.add(u);
14     if |L ∩ T(q)| ≥ k ∧ |L| ∩ T(p) ≥ k then
15         return L;
```

We try to incorporate users' SAC into TrustSVD, firstly, let us consider a geo-social network graph $G(V, E)$, which is a graph consists of vertex set V and edge set E. Vertices represnt the users and edges mean the relationships between users. There is a position$(v.x, v.y)$ for each vertex, normally we use longitude and latitude from datasets to describe it. The concept of SAC was proposed by Fang in [18], SAC is a subgraph G' of the graph G satisfying: (1) Connectivity: G' is connected. (2) Structure cohesiveness: the vertices of G' are linked intensively. (3) Spatial cohesiveness: all the vertices of G' are geography close to each other. Based on the theory of SAC, we propose an aglorithm to quantitatively calculate the degree of the spatial closeness between users. **Algorithm 1** presents SACQuery, firstly it initializes four parameters Q, L, R, β: Q is a priority queue of vertices, where vertices are sorted by desc order according the distance between vertices and q; L is the result of SACQuery; R recorded vertices which added into Q; β represents the closeness between users, apparently the first user who joins SAC is more close than the last one. Then, it adds q into Q (line 2). In the whole loop (line4-15), it first gets the nearest vertex p from Q, and adds p into L with the value $\frac{\beta}{|p,q|}$ where $|p,q|$ means the distance between p and q (line4-5). Next we consider the social neighbors of p (line6-13), for each neighbor u, if the distance from u to p is smaller than the distance between p and q, we put it into L. Otherwise, we add it into Q and R. The reason why we use R to decide whether u added into Q is that queue Q contains all vertices of dataset, if we use Q to calculate, this might have a very large time complexity. Finally, it checks whether there exists a subgraph SAC in graph G.

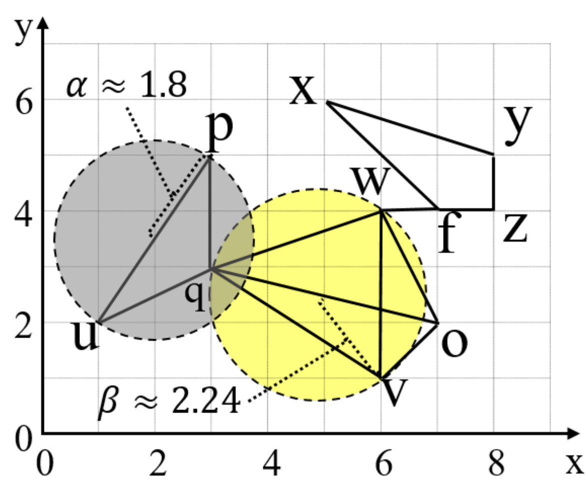

Fig. 4. The decomposition of user-item rating matrix

Fig.4 illustrates a geo-social network G contains 10 vertices, the solid lines denote their social relationships and $k = 2$.

Firstly, algorithm adds p to L and there is no SAC in subgarph. Then, it adds u into L, finds L with vertex set $\{q, p, u\}$. Compare L with the right SAC unit in **Fig.4**, L is far more cohesive. SACQuery based on Fang's theory of SAC in [18], we make some improvements to quantitatively calculate the proximity degree between users in SAC.

Time complexity. In **algorithm 1**, the while loop executes n times, and each loop costs $O(m)$, so the total time complexity is $O(nm)$.

As mentioned in section IV.A, let result L denote the user-user spatial matrix. For clarity, we present symbol L_u^+ as the set of users who is the member of u's SAC, let L_u^- denote the set of users whose SAC include u. We represent d_z as the d-dimensional user latent feature vector of user z who from L_u^+, thus we can predict spatial specific vector by $\hat{l}_{u,z} = d_z^T p_u$. Similarly, we have user spatial-feature matrix P and user spatial-feature matrix D, and recover the spatial matrix by $L \approx P^T D$. According to the theory proposed by Guo [7], we limit the spatial related users in L and the active user in the rating matrix to share the same user-feature space and bridge them together. Finally, we use loss function to learn P and D:

$$\min_{d*,p*} \sum_u \sum_{z \in L_u} (d_z^T p_u - l_{u,v})^2 + \lambda \Big(\sum_u ||p_u||_F^2 + \sum_i ||d_z||_F^2 \Big)$$

To get more accurate results, the importance of spatial information should be stressed. We enhance the spatial-unware TrustSVD model by using **algorithm 1** to add location relationship between users, and propose *TLSVD*. We take the geo-social network into account by modelling user preference in the same manner as trust, given by:

$$\begin{aligned} \hat{r}_{u,i} = b_{u,i} + q_i^T \big(p_u + |I_u|^{-\frac{1}{2}} \sum_{j \in I_u} y_j + |T_u|^{-\frac{1}{2}} \sum_{v \in T_u} w_v \\ + |S_u|^{-\frac{1}{2}} \sum_{x \in S_u} m_x + |L_u|^{-\frac{1}{2}} \sum_{z \in L_u} d_z \big) \end{aligned} \tag{5}$$

In a sense, **Equation 5** provides a 2-tier model for recommendation, it's the most important part in this paper. $b_{u,i}$ is the first tier to describe general properties of item and user which can be calculated by $b_{u,i} = b_u + b_i + \mu$, without considering any involved interactions. Take *Starbucks* as an example, the average score of all cafes is 3.7(μ), *Starbucks* is very popular all over the world, so the rating of *Starbucks* might be 0.5(b_i) higher than the average score. But our user Tom is very harsh on cafes, so the score may be 0.3(b_u) lower than the average. The next tier is the rest of formulate, it provides the interaction between user, item, trust, implicit trust and spatial profile. Considering the influence of spatial factors, with reference to **Equation 4**, we can derive the feature matrices T, S and L simultaneously by minimizing the following objective:

$$\begin{aligned} \mathcal{L} = \frac{1}{2} \sum_u \sum_{i \in I_u} (\hat{r}_{u,i} - r_{u,i})^w + \frac{\lambda}{2} \Big(\sum_u b_u^2 + \sum_i b_i^2 \\ + \sum_u ||p_u||_F^2 + \sum_i ||q_i||_F^2 + \sum_j ||y_j||_F^2 + \sum_u ||w_v||_F^2 \\ + \sum_x ||m_x||_F^2 + \sum_z ||d_z||_F^2 \Big) \end{aligned} \tag{6}$$

Adaptive Regularization. Yang proposed a technique called weighted λ regularization in [20], which helps objective function to avoid overfitting when learning parameters. For users who rated more items and for the items which got more ratings, Yang indicated more penalties on them. However, Guo proposed another idea [7], he argued that Yang's idea may force the model to be more biased on those popular users and items, Guo thought popular users should be less penalized and cold start users should be more regularized. After experiments, results show the Guo's idea is more suitable for our datasets. Recall the $e_{u,i} = \hat{r}_{u,i} - r_{u,i}$ represents the rating prediction error for user u on item i, $e_{u,v} = \hat{t}_{u,v} - t_{u,v}$ indicates the trust predict error for user u towards trustee v, $e_{u,z} = \hat{l}_{u,z} - l_{u,z}$ indicates the spatial-relation predict error for u towards user z whose SAC include u, as well as $e_{u,x} = \hat{s}_{u,x} - s_{u,x}$ for user u towards latent trustee x. After consider all relations, the new lose function to minimize is shown as:

$$\begin{aligned} \mathcal{L} = \frac{1}{2} \sum_u \sum_{i \in I_u} (\hat{r}_{u,i} - r_{u,i})^w + \frac{\lambda_t}{2} \sum_u \sum_{v \in T_u} \left(\hat{t}_{u,v} - t_{u,v} \right)^2 \\ + \frac{\lambda_s}{2} \sum_u \sum_{x \in S_u} (\hat{s}_{u,v} - s_{u,v})^2 + \frac{\lambda_l}{2} \sum_u \sum_{z \in L_u} \left(\hat{l}_{u,z} - l_{u,z} \right)^2 \\ + \frac{\lambda}{2} \sum_u |I_u|^{-\frac{1}{2}} b_u^2 + \frac{\lambda}{2} \sum_i |U_i|^{-\frac{1}{2}} b_i^2 \\ + \frac{1}{2} \sum_u \big(\lambda |I_u|^{-\frac{1}{2}} + \lambda_t |T_u|^{-\frac{1}{2}} \\ + \lambda_s |S_u|^{-\frac{1}{2}} + \lambda_l |L_u|^{-\frac{1}{2}} \big) ||p_u||_F^2 \\ + \frac{\lambda}{2} |T_v^+|^{-\frac{1}{2}} ||w_v||_F^2 + \frac{\lambda}{2} |S_x|^{-\frac{1}{2}} ||m_x||_F^2 + \frac{\lambda}{2} |L_z^+|^{-\frac{1}{2}} ||d_z||_F^2 \end{aligned} \tag{7}$$

Model Training. Gradient descent algorithms can be used to train the TLSVD model, model parameters are determined by minimizing the associated objective functions. We loop over all known ratings, trust matrix, implicit trust martix and spatial matrix. For a given training case $r_{u,i}$, we modify the parameters by moving in the opposite direction of the gradient. The gradients of $\mathcal{L}$ in **equation 8** with respect to b_u, b_i, p_u,q_i, y_i, w_v, m_x, d_z. Recall U_i, U_j represent the set of users who rate items i and j, T_u^+ is the set of users who trust user u, S_u is the set of users who correlate with user u, L_u^+ represents the set of users whose SAC include u.

$$
\begin{aligned}
b_u &\leftarrow b_u - \gamma\Big(\sum_{i\in I_u} e_{u,i} + \lambda|I_u|^{-\frac{1}{2}} b_u\Big)\\
b_i &\leftarrow b_i - \gamma\Big(\sum_{u\in U_i} e_{u,i} + \lambda|U_i|^{-\frac{1}{2}} b_i\Big)\\
p_u &\leftarrow p_u - \gamma\Big[\sum_{j\in I_u} e_{u,i} q_i\\
&\quad + \lambda_t \sum_{v\in T_u} e_{u,v} w_v + \lambda_s \sum_{x\in S_u} e_{u,x} m_x + \lambda_l \sum_{z\in L_u} e_{u,z} d_z\\
&\quad + \Big(\lambda|I_u|^{-\frac{1}{2}} + \lambda_t|T_u|^{-\frac{1}{2}} + \lambda_s|S_u|^{-\frac{1}{2}} + \lambda_t|L_u|^{-\frac{1}{2}}\Big) p_u\Big]\\
q_i &\leftarrow q_i - \gamma\Big[\sum_{u\in U_i} e_{u,i}\big(p_u + |I_u|^{-\frac{1}{2}} \sum_{i\in I_u} y_i + |T_u|^{-\frac{1}{2}} \sum_{v\in T_u} w_v\\
&\quad + |S_u|^{-\frac{1}{2}} \sum_{x\in S_u} m_x + |L_u|^{-\frac{1}{2}} \sum_{z\in L_u} d_z\big) + \lambda|U_i|^{-\frac{1}{2}} y_i\Big]\\
\forall j \in I_u, y_i &\leftarrow y_i - \gamma\Big[\sum_{i\in I_u} e_{u,i}|I_u|^{-\frac{1}{2}} q_i + \lambda|U_j|^{-\frac{1}{2}} q_j\Big]\\
\forall v \in T_u^+, w_v &\leftarrow w_v - \gamma\Big[\sum_{i\in I_u} e_{u,i}|T_u^+|^{-\frac{1}{2}} q_i\\
&\quad + \lambda_t e_{u,v} p_u + \lambda|T_v^+|^{-\frac{1}{2}} w_v\Big]\\
\forall x \in S_u, m_x &\leftarrow m_x - \gamma\Big[\sum_{i\in I_u} e_{u,i}|S_u|^{-\frac{1}{2}} q_i\\
&\quad + \lambda_t e_{u,x} p_u + \lambda|S_x|^{-\frac{1}{2}} m_x\Big]\\
\forall z \in L_u^+, d_z &\leftarrow d_z - \gamma\Big[\sum_{i\in I_u} e_{u,i}|L_u^+|^{-\frac{1}{2}} q_i\\
&\quad + \lambda_t e_{u,z} p_u + \lambda|L_z^+|^{-\frac{1}{2}} d_z\Big]
\end{aligned} \tag{8}
$$

V. Experiments and Results

Datasets. Four datasets are shown in **TABLE II**.

Cross-validation. Traditionally, we use 5-fold cross-validation for learning and testing, which means each dataset will split into five folds. In each iteration, four folds will be used to train model, the last fold is used as the test set. After 5 iterations, all folds will make sure to be tested.

$$
MAE = \frac{\sum_{u,i} |\hat{r}_{u,i} - r_{u,i}|}{N}, RMSE = \sqrt{\frac{\sum_{u,i} (\hat{r}_{u,i} - r_{u,i})^2}{N}}
$$

Evaluation Metrics. We use mean absolute error (MAE) and root mean square error (RMSE) as benchmark error evaluation metrics, N indicates the number of testing set, smaller values of MAE and RMSE represent better accuracy of prediction. Because all RSs try to reduce the square errors of predictions and real data, RMSE is more indicative than MAE. The results are shown in **Fig.5**, where we only illustrate the performance by RMSE and d = 10 because of space limitation.

Comparison Methods and Parameter Settings. We compared TSSVD and TLSVD with three models: (1)Baseline: widely used recommendation method, PMF (probabilistic Gaussian matrix factorization) [21] and ItemKNN [22]. (2)Trust-based model: Sorec [5] and TrustSVD [7]. (3)Rating-only model: SVD++ [2]. For each method, we set the optimal parameters respectively according to corresponding references, given in **TABLE III**. As **Fig.5** shows, there are significant differences among the models. TrustSVD shows the best performance in Yelp dataset, after analysis, we find that Yelp has the densest social network in our experimental datasets as shown in **Fig.6.(B)**, but our algorithms don't fall behind very much. On Dianping, ItemKNN performs better than other methods, this might because of Dianping's dataset features, users from dianping are very fond of rating as presented in **Fig.6.(A)**, more than 20% users in Dianping give twenty or more evaluations. ItemKNN is very suitable for processing the datasets with intensive rating data. While TLSVD performs best in Foursquare and Gowalla, this might because these two datasets have rich location data. Compared with other datasets, Foursquare and Gowalla suffer a lot from data sparsity.

TABLE III
PARAMETER SETTINGS

Methods	Optimal parameters
PMF [21]	$\lambda = 0.001$
ItemKNN [22]	$k = \infty$
SoRec [5]	$\lambda_c = 0.0001, 0.001, 0.01, 0.001$ for Yelp, Foursquare, Dianping, Gowalla
TrustSVD [7]	$\lambda = 0.0005, \lambda_t = 0.9$
SVD++ [2]	$\lambda = 0.0003, 0.0025, 0.1, 0.035$ respectively
TLSVD	$\lambda = 0.0001, \lambda_t = 0.5, \lambda_s = 0.2, \lambda_l = 0.4$

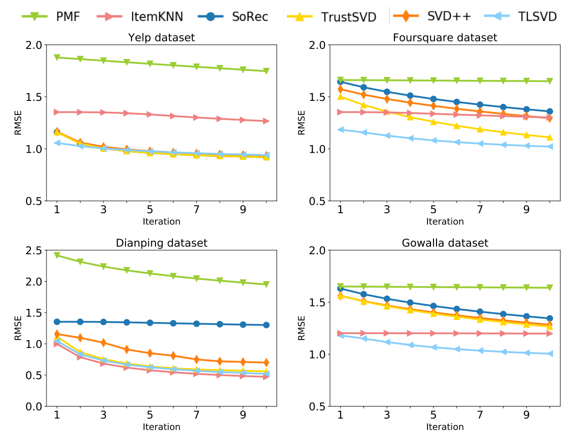

Fig. 5. RMSE of four datasets

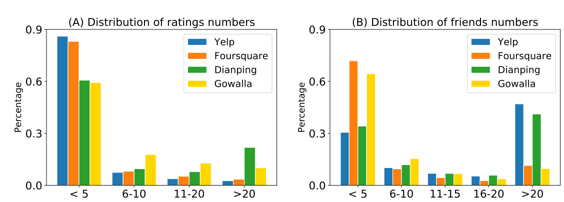

Fig. 6. Data distribution

For this kind of dataset structure, trust-based RSs which purely mine the user-item rating matrix and user-user trust matrix for recommendation, giving somewhat unsatisfactory results.

we are encouraged, even somehow excited about the results. As Koren pointed out that even a small number of improvements in RMSE translate into significant improvements of RS in real products [2].

VI. CONSLUSIONS AND FUTURE WORK

In this paper, we make two improvements to the TrustSVD. Firstly, we convert history rating matrix into implicit trust to quantitatively calculate similarity of users' preferences. Secondly, with the help of location information and social network, we can obtain the degree of spatial closeness between users. Based on these improvements, this paper introduces a novel RS model TLSVD. Our experiments on four datasets show that TLSVD is more accurate than the state-of-the-art recommendation algorithms, this work will significantly boost prediction accuracy of Location-based services. In the future, we intend to consider how the item location affects user preference.

REFERENCES

[1] Adomavicius, Gediminas, and Alexander Tuzhilin. "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions." IEEE transactions on knowledge and data engineering 17.6 (2005): 734-749.

[2] Koren, Yehuda. "Factorization meets the neighborhood: a multifaceted collaborative filtering model." Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2008.

[3] Guo, Guibing, Jie Zhang, and Daniel Thalmann. "A simple but effective method to incorporate trusted neighbors in recommender systems." International Conference on User Modeling, Adaptation, and Personalization. Springer, Berlin, Heidelberg, 2012.

[4] Liu, Yong, et al. "Exploiting geographical neighborhood characteristics for location recommendation." Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. ACM, 2014.

[5] Ma, Hao, et al. "Sorec: social recommendation using probabilistic matrix factorization." Proceedings of the 17th ACM conference on Information and knowledge management. ACM, 2008.

[6] Sarwat, Mohamed, et al. "LARS*: An efficient and scalable location-aware recommender system." IEEE Transactions on Knowledge and Data Engineering 26.6 (2014): 1384-1399.

[7] Guo, Guibing, Jie Zhang, and Neil Yorke-Smith. "TrustSVD: Collaborative Filtering with Both the Explicit and Implicit Influence of User Trust and of Item Ratings." AAAI. Vol. 15. 2015.

[8] Ma, Hao, et al. "Recommender systems with social regularization." Proceedings of the fourth ACM international conference on Web search and data mining. ACM, 2011.

[9] Sarwar, Badrul, et al. "Item-based collaborative filtering recommendation algorithms." Proceedings of the 10th international conference on World Wide Web. ACM, 2001.

[10] Manchanda, Puneet, Grant Packard, and Adithya Pattabhiramaiah. "Social dollars: The economic impact of customer participation in a firm-sponsored online customer community." Marketing Science 34.3 (2015): 367-387.

[11] Linden, Greg, Brent Smith, and Jeremy York. "Amazon. com recommendations: Item-to-item collaborative filtering." IEEE Internet computing 7.1 (2003): 76-80.

[12] Jin, Rong, Joyce Y. Chai, and Luo Si. "An automatic weighting scheme for collaborative filtering." Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2004.

[13] Xue, Gui-Rong, et al. "Scalable collaborative filtering using cluster-based smoothing." Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2005.

[14] Deshpande, Mukund, and George Karypis. "Item-based top-n recommendation algorithms." ACM Transactions on Information Systems (TOIS) 22.1 (2004): 143-177.

[15] Armenatzoglou, Nikos, Stavros Papadopoulos, and Dimitris Papadias. "A general framework for geo-social query processing." Proceedings of the VLDB Endowment 6.10 (2013): 913-924.

[16] Sarwat, Mohamed, et al. "LARS*: An efficient and scalable location-aware recommender system." IEEE Transactions on Knowledge and Data Engineering 26.6 (2014): 1384-1399.

[17] Zhang, Wei, Jianyong Wang, and Wei Feng. "Combining latent factor model with location features for event-based group recommendation." Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2013.

[18] Fang, Yixiang, et al. "Effective community search over large spatial graphs." Proceedings of the VLDB Endowment 10.6 (2017): 709-720.

[19] Papagelis, Manos, Dimitris Plexousakis, and Themistoklis Kutsuras. "Alleviating the sparsity problem of collaborative filtering using trust inferences." International Conference on Trust Management. Springer, Berlin, Heidelberg, 2005.

[20] Yang, Bo, et al. "Social collaborative filtering by trust." IEEE transactions on pattern analysis and machine intelligence 39.8 (2017): 1633-1647.

[21] Mnih, Andriy, and Ruslan R. Salakhutdinov. "Probabilistic matrix factorization." Advances in neural information processing systems. 2008.

[22] Deshpande, Mukund, and George Karypis. "Item-based top-n recommendation algorithms." ACM Transactions on Information Systems (TOIS) 22.1 (2004): 143-177.