A Clustering Based Social Matrix Factorization Technique for Personalized Recommender Systems

Completed Research

Divyaa L.R
Indian Institute of Technology Madras
lrdivyaa3011@gmail.com

Aniruddha Tamhane
Indian Institute of Technology Madras
aniruddha16293@gmail.com

Nargis Pervin
Indian Institute of Technology Madras
nargisp@iitm.ac.in

Abstract

Recently, a new paradigm of social network based recommendation approach has emerged wherein structural features from social network turned out to be an effective measure to improve the efficacy of the algorithms. However, these approaches assume a user is impacted by all his social connections and completely ignore their preferential similarity, which is crucial for personalized recommendations. Herein, we address this pivotal issue and propose a two-stage clustering based matrix-factorization algorithm, "**C**luster **RE**finement on **P**reference **E**mbedded MF (CREPE MF)" using a subgraph of social network that integrates the preferential similarity score. Clustering has been applied first on the user followed by the item based on ratings. The proposed algorithm has been systematically evaluated with state-of-the-art algorithms in terms of prediction accuracy and runtime complexity using real-world Yelp dataset. Gratifyingly, our approach outperforms the state-of-the-art algorithms with up to 12.97% and 29.60% improvements in RMSE and runtime, respectively.

Keywords

Probabilistic Matrix Factorization, Two-stage Clustering, Personalized Recommendations, Social Network, Preference Network, Recommender System

Introduction

Recommendation technology constitutes an alluring field of research owing to their extensive applications in the area of machine learning, data mining, sociology, and physicist communities (Vespignani, 2009; Dunbar, 2016; Newman et al., 2011). Many e-Marketplaces such as Amazon, Alibaba, eBay, etc. have integrated recommender systems at the core of their business models and also judiciously applied in the field books (Linden et al., 2003), movies (Lekakos and Caravelas, 2008), music (Davidson et al., 2010), etc.

Broadly, a Recommender System (RS) problem consists of a set of users, a set of items, and the preference of the users for those items in terms of ratings, reviews or other sources of information where the task is to predict the missing preferences for the users. In this context, Collaborative Filtering (CF) based RS is the most popular technique in use. The underlying assumption in CF is that if the preference of a user u is alike to that of another user v on a set of items, then u and v will have similar preference for other items as well; the similarity of a pair of users is computed with the common items they have rated or reviewed using well-known similarity measures (e.g., Pearson Correlation Coefficients, Cosine similarity, etc.). While traditional CF techniques are operationally simple and easy to implement, they suffer from well-known data-sparsity and cold-start problems for new users and items. Further, CF method does not scale well as the number of users and items grow exponentially and thus its adaptability in real-world applications is plagued.

To deal with these challenges of scalability and sparsity, Matrix Factorization (MF) techniques (Sarwar et al., 2002a; Mnih and Salakhutdinov, 2008) have gained significant attention, where the rating matrix $R_{m,n}$ is

approximated by product of lower dimensional matrices which capture the latent or unobserved attributes of the users and items. In particular, Probabilistic Matrix Factorization (PMF) bodes well with a large user-item rating matrix with good prediction accuracy for users with few ratings. Another popular approach dealing with the scalability issue is the clustering based CF techniques (Sarwar et al., 2002b; Ungar and Foster, 1998; Pham et al., 2011), which divides the user-item rating space into smaller clusters and the recommendations are generated using CF method in individual clusters. This approach manages to reduce the runtime complexity of the RSs, however, compromises the recommendation quality (accuracy). We envisage that the amalgamation of model-based approach with clustering will improve the scalability as well as accuracy to a great extent. To the best of our knowledge clustering has primarily been applied on memory-based recommendation approaches and implementation on model-based approach is scarce. Herein, we propose a two-stage clustering based probabilistic matrix factorization technique which incorporates user's *Preference Network (PN)* connections into PMF model and refer that as "**C**luster **RE**finement on **P**reference **E**mbedded MF (CREPE MF)". *Preference Network* has been defined as a graph where an edge exists between a pair of users if they have similar preference on a set of items. In fact, *PN* is a subgraph of actual social network. It is worth noting that unlike *PN* based RS, social recommendation systems (Ma et al., 2011; Jamali and Ester, 2010; Ma et al., 2008), which reinforce the user's social connections into RS system, assumes that users and all their social connections share similar preference for a set of items. However, in reality, among the myriad of online connections on social platform only a few exhibit similar taste (Dunbar, 2016; Brzozowski et al., 2008) and each of these connections have different level of significance in the focal user's decision making. Experiments have been conducted on a real-world dataset collected from Yelp dataset challenge 2017 and compared with four state-of-the-art algorithms as presented in (a) Jamali and Ester (2010), (b) Mnih and Salakhutdinov (2008),(c) Yang et al. (2013) and (d) Ma et al. (2011). The experimental results demonstrate the efficacy of our approach compared to baseline algorithms with a significant improvement of 12.97% in accuracy and 29.60% in runtime.

Literature Review

Collaborative Filtering

Collaborative Filtering (CF), a renowned approach to recommender systems, can be broadly classified into two categories: memory (neighbours) based (Linden et al., 2003) and model based (Lü et al., 2012; Mnih and Salakhutdinov, 2008). While the memory-based approaches are inefficient for large in-memory data, model-based CF approaches which extract information from the rating matrices are effective for real-time recommendation systems. Latent factor model, a class of traditional model-based approach, assumes that the large sparse rating matrix can be represented as the product of user and item latent feature matrices where the user and item ratings are assumed to depend on a set of implicit factors. For example, the preference of a user for a restaurant can depend over cuisines, cost, ambiance, etc., which are not transparent from the given ratings. Various latent factor models such as Singular Value Decomposition (SVD) and other variations of matrix factorization techniques have been proposed in the literature (Sarwar et al., 2002a; Mnih and Salakhutdinov, 2008; Jamali and Ester, 2010; Ma et al., 2008). The Probabilistic Matrix Factorization (PMF) technique (Mnih and Salakhutdinov, 2008) is the state-of-the-art latent factor model which factorizes the rating matrix into product of two lower-dimensional matrices to improve accuracy. It also scales linearly with the number of users and performs well on large, sparse datasets.

Social Network Based Recommender Systems

Recent trust based social recommender systems aim to increase the coverage of recommendations while preserving the quality of predictions (Massa and Avesani, 2004; Wang et al., 2011). Several latent factor techniques have been proposed (Ma et al., 2008; Jamali and Ester, 2010) where trust has been propagated using user and item features, incorporating users' social network information in the PMF model. Ma et al. (Ma et al., 2011) introduced a social regularization to the Matrix Factorization approach, where the influence of users' friends on the users' latent feature is weighted differently based on the rating similarity between the users and their friends. Yang et al. (Yang et al., 2013) proposed a Location Based Social Matrix Factorization (LBSMF) approach which incorporates the item-item similarity network based on item attributes along with social network influence to the PMF model. This approach facilitates the propagation of influence through the social network of users and the network of similar items. The underlying assumption in such methods

is that all of the online friends in social network circle influence the active users' preference for items. However, in reality majority of the social network friends do not share similar interests (Dunbar, 2016) and hence they probably have very little impact on active users' decisions. Therefore, computing similarity among the social connections based on their true item preference is crucial and thus a novel methodology that exploits preference-similarity network in the PMF model to capture common preferences between the active user and other social connections is highly desirable.

Clustering Based Recommender Systems

Grouping users into clusters based on the rated items have turned out to be a scalable approach in memory-based CF models (Pham et al., 2011; Ungar and Foster, 1998; Sarwar et al., 2002b). An early work by Ungar and Foster (Ungar and Foster, 1998) proposed a statistical approach based on repeated clustering (using K-means and other soft-clustering method) and Gibbs sampling revealed that the generative model using K-means compared to Gibbs sampling is faster, however, it does not improve the accuracy. Sarwar et al. (2002b) demonstrated that clustering based RSs are efficient compared to the traditional CF methods and are useful for online applications. Clustering has also been applied on social network based recommendations (Pham et al., 2011; Shepitsen et al., 2008) in order to find the communities of similar users from the friends' network which in turn reduces the sparsity issue.

In this paper, we propose a model based clustering technique that aims to overcome the sparsity and scalability issues with an improvement in prediction accuracy. Our proposed approach "**C**luster **RE**finement on **P**reference **E**mbedded MF (CREPE MF)" considers both user similarity network and venue similarity network.

Solution Overview

The aim of this study is to recommend venues to the users based on their preferential attributes mined from the ratings and social connections. The problem has been addressed in four steps - (1) Two stage clustering based on ratings, (2) Social preference network and inter-venue similarity graph computation, (3) Modeling user and item profiles (cluster-wise), (4) Recommendation of venues using two-stage cluster PMF.

Probabilistic Matrix Factorization (PMF) model (Mnih and Salakhutdinov, 2008) is a popularly used approach in recommendation literature. In PMF, the utility matrix $R_{m\times n}$ is factorized to two smaller dimensional matrices $U_{m\times l}$ and $V^T_{n\times l}$, user-latent feature matrix and item-latent feature matrix, respectively.

$$R_{m\times n} \approx U_{m\times l} \times V^T_{n\times l} \tag{1}$$

where l denotes the dimension of the latent feature space.

$$p(R|U,V,\sigma_R^2) = \prod_{u=1}^{m}\prod_{i=1}^{n}[\mathcal{N}(R_{u,i}|U_u \times {V_i}^T, \sigma_R^2)]^{I_{u,i}} \qquad \text{where } \boldsymbol{I}_{u,i} = \begin{matrix} 1\ if\ user\ u\ has\ rated\ item\ i \\ = 0\ otherwise \end{matrix} \tag{2}$$

Here $\mathcal{N}(x|\mu,\sigma^2)$ denotes the normal distribution with mean μ and variance σ^2. For both user and item latent features Gaussian prior distribution has been assumed which are shown below.

$$p(U|\sigma_U^2) = \prod_{u=1}^{m}\mathcal{N}(U_u|0,\sigma_U^2\boldsymbol{I}) \tag{3}$$

$$p(V|\sigma_V^2) = \prod_{i=1}^{n}\mathcal{N}(V_i|0,\sigma_V^2\boldsymbol{I}) \tag{4}$$

Based on Bayesian inference, the posterior probability distribution of U and V are given as:

$$p(U,V|R,\sigma_R^2,\sigma_U^2,\sigma_V^2) \propto p(R|U,V,\sigma_R^2)p(U|\sigma_U^2)p(V|\sigma_V^2) \tag{5}$$

From Equation 5, U and V can be learnt for predicting missing ratings. We have formulated the Probabilistic Matrix Factorization (PMF) approach that includes the effect of similar social connections and similar items' network which iteratively propagates the preferential influence in the similar user and venue latent features. To improve the scalability of the model, the preference based PMF technique has been applied in clusters derived in two stages. Herein, we introduce a *Preference Enhanced Social Network* for users and cluster the users and items with the following propositions:

Preference Network: To evaluate the influence propagation we consider users' direct as well as two-hop connections. Theory of homophily (Zuo et al., 2014) states that people with similar characteristics tend to form ties. On the other hand, principle of triadic closure coined by M. Granovetter (Granovetter, 1973)

postulates that two individuals with common friends are likely to become friends in the future. Therefore, we model the propagation of social influence through a subgraph of direct and two-hop connections, which we refer as *Preference Network*, can potentially improve quality of the recommendations. User's similarity is computed using several preference attributes including rating information.

Rating Bubbles: Online users are heavily inundated with historical reviews and ratings (Aral, 2014). Consequently, a user's opinion for an item follows the herd instinct and a highly rated item is susceptible to get a high rating. Similarly, the user who consistently rates items lower than the average is very likely to give a low rating to a different item and vice versa. For example, a highly rated restaurant is likely to get high ratings consistently and the users who usually rate high (low) will possibly give high (low) rating to the next restaurant they visits. Thus, users and items are trapped in *Rating Bubbles*. Inspired by this phenomenon, we hypothesize that clustering users and items based on their previous rating patterns can identify these rating bubbles and improve quality of recommendations.

Solution Details

Two-Stage Clustering: We propose that users and items should be grouped into clusters based on ratings to identify the similar rating patterns. This idea seems intuitive considering that users and items are trapped in rating bubbles (Aral, 2014). The numbers of user clusters and item clusters need to be carefully chosen since a very low number of clusters fails to identify all the rating bubbles, while a very high number of clusters will increase the sample variance, which will adversely lower the prediction accuracy. We define the rating pattern $\mathcal{U}_u$ for a user u and $\mathcal{I}_i$ for an item i as 2-dimensional vectors containing the mean and the standard deviation of the ratings by user u and item i respectively defined as:

$$\mathcal{U}_u = \left[\overline{R_u}, \quad \sqrt{\frac{\sum_{i=1}^{n} \boldsymbol{I}_{u,i}(R_{u,i} - \overline{R_u})^2}{\sum_{i=1}^{n} \boldsymbol{I}_{u,i} - 1}}\right] \tag{6}$$

$$\mathcal{I}_i = \left[\overline{R_i}, \quad \sqrt{\frac{\sum_{u=1}^{m} \boldsymbol{I}_{u,i}(R_{u,i} - \overline{R_i})^2}{\sum_{u=1}^{m} \boldsymbol{I}_{u,i} - 1}}\right] \tag{7}$$

Here $\overline{R_u}$ and $\overline{R_i}$ are the u^{th} user's and i^{th} item's mean ratings. The rating matrix is clustered in two-stages – in the first stage we get n_u user clusters, where users with similar rating pattern will be grouped together using K-Means (Hartigan and Wong, 1979) or any other soft clustering algorithms. The optimal number of user clusters (n_u) and item clusters (n_i) are chosen in a way such that the RMSE and runtime is minimized. The process also ensures to retain substantial number of similar users and similar items in each cluster. First, we varied the number of user clusters and chose the optimal number of clusters based RMSE and runtime. Next, we repeated the same process for determining the number of item clusters. However, to ensure that the final clusters are not too sparse because of over clustering, the second level item clustering is done only for those user clusters for which the cardinality exceeds 10% of the total interactions.

Preference Network Computation: The traditional user-user similarity computation is based on the common ratings. This approach inadequately captures user's personal culinary preferences. Therefore, we as well consider user's implicit preference attributes in the user preference similarity computation used in the construction of *Preference Network*. For every user u, the implicit Preference Vector (P_u) is defined as in Equation 8 where A_i is the d-dimensional item attribute vector for the i^{th} item. For example, if restaurant 1 serves Fast Food and Dessert, restaurant 2 serves Fast Food, Dessert, and Chinese, $A_1 = [1,\ 1,\ 0]$, $A_2 = [1,\ 1,\ 1]$ where A_{i1}, A_{i2}, A_{i3} correspond to Fast Food, Dessert, and Chinese respectively. The dimension of the attribute vectors ($d = 3$ in this example) depends on the total number of attributes for all the restaurants. We compute an average rating given by user u for all the items following Equation 9.

$$P_u = \frac{\sum_{i=1}^{n} A_i \boldsymbol{I}_{u,i}}{\sum_{i=1}^{n} \boldsymbol{I}_{u,i}} \tag{8}$$

$$\overline{R_u} = \frac{\sum_{i=1}^{n} R_{u,i} \boldsymbol{I}_{u,i}}{\sum_{i=1}^{n} \boldsymbol{I}_{u,i}} \tag{9}$$

The $(d+1)$ dimensional User Preference Vector $(\mathcal{P}_u)$ is updated by concatenating Equations 8 and 9 to reflect the user's implicit and explicit item preferences as $\mathcal{P}_u = P_u || \overline{R_u}$ (10) . Preference vector is further standardized to scale the attribute values in $[0, 1]$ and is denoted by $\hat{\mathcal{P}}_u$. We compute the user preference similarity between users u and v using the cosine similarity as:

$$PrefSim(u, v) = \frac{\hat{\mathcal{P}}_u^T \hat{\mathcal{P}}_v}{||\hat{\mathcal{P}}_u|| \, ||\hat{\mathcal{P}}_v||} \tag{11}$$

We define the *Preference Network (PN)* for the user u based on the preferences as:

$$S_{u,\mathcal{V}}(\theta_u) = \{v \in \mathcal{V} | PrefSim(u, v) \geq \theta_u\} \ \forall \, u \in \mathbf{U} \tag{12}$$

where θ_u is an experimentally fixed threshold on *PrefSim* so that two users are connected by an edge in the *PN* and $\mathcal{V}$ denotes the set of direct and second-hop connections from social network.

Item Network Computation: We compute the item-item similarity ($Sim(i,j)$) between the d-dimensional item attribute vectors A_i with A_j using cosine similarity as shown in Equation 11. Finally, we define the *Item similarity network (IN)* of item i, analogous to *PN* as:

$$S_{i,\mathcal{J}}(\theta_i) = \{j \in \mathcal{J} | Sim(i,j) \geq \theta_i\} \ \forall\, i \in \mathbf{I} \tag{13}$$

where $\mathcal{J}$ denotes the set of items. Hereafter, we simplified the notations of $S_{u,\mathcal{V}}(\theta_u)$ and $S_{i,\mathcal{J}}(\theta_i)$ as S_u and S_i, respectively.

Modeling User and Item Profiles (Cluster-wise): We assume that the user's (item's) latent vector be Gaussian distributed with the mean being the average of the other similar users' (items') latent features. Thus extending Equations 3 and 4:

$$p(U|S_{\mathbf{U}}, \sigma_U^2, \sigma_{S_{\mathbf{U}}}^2) = \prod_{u=1}^{m} \mathcal{N}(U_u|0, \sigma_U^2 \boldsymbol{I}) \times \prod_{u=1}^{m} \mathcal{N}\Big(U_u | \frac{\sum\limits_{s \in S_u} PrefSim(u,s)\, U_s}{\sum\limits_{s \in S_u} PrefSim(u,s)}, \sigma_{S_{\mathbf{U}}}^2 \boldsymbol{I}\Big) \tag{14}$$

$$p(V|S_{\mathbf{I}}, \sigma_V^2, \sigma_{S_{\mathbf{I}}}^2) = \prod_{i=1}^{n} \mathcal{N}(V_i|0, \sigma_V^2 \boldsymbol{I}) \times \prod_{i=1}^{n} \mathcal{N}\Big(V_i | \frac{\sum\limits_{z \in S_i} Sim(i,z)\, V_z}{\sum\limits_{z \in S_i} Sim(i,z)}, \sigma_{S_{\mathbf{I}}}^2 \boldsymbol{I}\Big) \tag{15}$$

From Equations 5,14,15 and using Bayesian inference, the posterior probability of the latent features can be written as:

$$\begin{aligned} p(U,V|R,S_{\mathbf{U}},S_{\mathbf{I}},\sigma_R^2,\sigma_U^2,\sigma_V^2,\sigma_{S_{\mathbf{U}}}^2,\sigma_{S_{\mathbf{I}}}^2) &= \prod_{u=1}^{m}\prod_{i=1}^{n}[\mathcal{N}(R_{u,i}|g(U_u V_i^T),\sigma_R^2)]^{\boldsymbol{I}_{u,i}} \quad \times \prod_{u=1}^{m}\mathcal{N}(U_u|0,\sigma_U^2\boldsymbol{I}) \\ \times \prod_{u=1}^{m}\mathcal{N}\Big(U_u|\frac{\sum\limits_{s\in S_u} PrefSim(u,s)\,U_s}{\sum\limits_{s\in S_u} PrefSim(u,s)}, \sigma_{S_{\mathbf{U}}}^2\boldsymbol{I}\Big) &\times \prod_{i=1}^{n}\mathcal{N}(V_i|0,\sigma_V^2\boldsymbol{I}) \times \prod_{i=1}^{n}\mathcal{N}\Big(V_i|\frac{\sum\limits_{z\in S_i} Sim(i,z)\,V_z}{\sum\limits_{z\in S_i} Sim(i,z)}, \sigma_{S_{\mathbf{I}}}^2\boldsymbol{I}\Big) \end{aligned} \tag{16}$$

where $g(x)$ is the modified logistic function used for keeping the predictions in range [1, 5] defined as:

$$g(x) = \min_{u\in\mathbf{U}, i\in\mathbf{I}} R_{u,i} + \frac{\max_{u\in\mathbf{U}, i\in\mathbf{I}} R_{u,i} - \min_{u\in\mathbf{U}, i\in\mathbf{I}} R_{u,i}}{1+e^{-x}} \tag{17}$$

The log posterior probability can be obtained by taking the natural logarithm of Equation 16.

$$\begin{aligned} ln(p(U,V|R,S_{\mathbf{U}},S_{\mathbf{I}},\sigma_R^2,\sigma_{\mathbf{U}}^2,\sigma_V^2,\sigma_{S_{\mathbf{U}}}^2,\sigma_{S_{\mathbf{I}}}^2) &= -\frac{1}{\sigma_R^2}\Big(\frac{1}{2}\sum_{u=1}^{m}\sum_{i=1}^{n} I_{u,i}(R_{u,i}-g(U_uV_i^T))^2 + \frac{\sigma_R^2}{\sigma_{\mathbf{U}}^2}\frac{1}{2}\sum_{u=1}^{m}\|U_u\|^2 + \frac{\sigma_R^2}{\sigma_V^2}\frac{1}{2}\sum_{i=1}^{n}\|V_i\|^2 \\ + \frac{\sigma_R^2}{\sigma_{S_{\mathbf{U}}}^2}\frac{1}{2}\sum_{u=1}^{m}\|U_u - \sum_{s\in S_u}\frac{PrefSim(u,s)\,U_s}{\sum_{s\in S_u} PrefSim(u,s)}\|^2 + \frac{\sigma_R^2}{\sigma_{S_{\mathbf{I}}}^2}\frac{1}{2}\sum_{i=1}^{n}\|V_i - \sum_{z\in S_i}\frac{Sim(i,z)\,V_z}{\sum_{z\in S_i} Sim(i,z)}\|^2\Big) + \mathcal{C} \end{aligned} \tag{18}$$

Let us assume that C is the number of clusters. Hence, for each cluster $c \in \{1,2,...,C\}$, our objective becomes minimizing the following equation (after minor rearrangement of Equation 18):

$$\begin{aligned} Z^c(R^c,S_{\mathbf{U}}^c,S_{\mathbf{I}}^c,U^c,V^c) &= \frac{1}{2}\sum_{u\in M^c}\sum_{i\in N^c} I_{u,i}(R_{u,i}^c - g(U_u^c V_i^{cT}))^2 + \frac{\gamma_U^c}{2}\sum_{u\in M^c}\|U_u^c\|^2 + \frac{\gamma_V^c}{2}\sum_{i\in N^c}\|V_i^c\|^2 \\ + \frac{k_U^c}{2}\sum_{u\in M^c}\|U_u^c - \sum_{s\in S_u^c}\frac{PrefSim(u,s)\,U_s^c}{\sum_{s\in S_u^c} PrefSim(u,s)}\|^2 + \frac{k_V^c}{2}\sum_{i\in N^c}\|V_i^c - \sum_{z\in S_i^c}\frac{Sim(i,z)\,V_z^c}{\sum_{z\in S_i^c} Sim(i,z)}\|^2 \end{aligned} \tag{19}$$

where $\|\cdot\|$ denotes the Frobenius norm. S_u^c and S_i^c denote the set of similar users for user u and set of similar items for item i in cluster c, respectively. The hyperparameters related to the Bayesian variances are as follows: $\gamma_U^c = \frac{\sigma_R^{c2}}{\sigma_U^{c2}}$, $\gamma_V^c = \frac{\sigma_R^{c2}}{\sigma_V^{c2}}$, $k_U^c = \frac{\sigma_R^{c2}}{\sigma_{S_{\mathbf{U}}}^{c2}}$, $k_V^c = \frac{\sigma_R^{c2}}{\sigma_{S_{\mathbf{I}}}^{c2}}$

Here, R^c, U^c, V^c are the user-item ratings matrix, user, and item latent features in cluster c. $\gamma_U^c, \gamma_V^c, k_U^c, k_V^c$ are the hyper-parameters and M^c, N^c are the sets of users and items with respect to the cluster c. Also, $S_u^c = S_u \cap M^c$ and $S_i^c = S_i \cap N^c$.

The details of the Two Stage Rating Pattern Clustering Algorithm has been presented in Algorithm 1. The algorithm elucidates a method of clustering the m users, n items into n_u user-clusters and n_i item-clusters

Algorithm 1 Two-Stage Rating Pattern Clustering

1: **Input**: R, n_u, n_i, δ
2: **Return**: **M**, **N**, **R**
3: Use K-Means to cluster M users into $\{UC_1, UC_2, ..., UC_{n_u}\}$ clusters in $\mathcal{U}$ space
4: **Initialize**: $c \leftarrow 1$, $M \leftarrow \emptyset$, $N \leftarrow \emptyset$, $R \leftarrow \emptyset$
5: **for** $u \in \{1, ..., n_u\}$ **do**
6: $temp \leftarrow \{j | R_{v,j} \neq 0 \quad \forall v \in UC_u\}$
7: **if** $|UC_u| \leq \delta$ **then**
8: $M^c \leftarrow UC_u$
9: $N^c \leftarrow temp$
10: $R^c \leftarrow [R_{m,n}]_{|M^c| \times |N^c|}$ ***s.t.*** $\boldsymbol{I}_{m,n} \neq 0$
11: $c \leftarrow c + 1$
12: **else**
13: Use K-Means to cluster $temp$ items into $\{IC_1, IC_2, ..., IC_{n_i}\}$ clusters in $\mathcal{I}$ space
14: **for** $i \in \{1, ..., n_i\}$ **do**
15: $M^c \leftarrow UC_u$
16: $N^c \leftarrow IC_i$
17: $R^c \leftarrow [R_{m,n}]_{|M^c| \times |N^c|}$ ***s.t.*** $\boldsymbol{I}_{m,n} \neq 0$
18: $c \leftarrow c + 1$
19: $c \leftarrow c - 1$
20: $\mathbf{M} \leftarrow \{M^1, M^2, .., M^c\}$, $\mathbf{N} \leftarrow \{N^1, N^2, .., N^c\}$, $\mathbf{R} \leftarrow \{R^1, R^1, .., R^c\}$
21: ***return***$(\mathbf{M}, \mathbf{N}, \mathbf{R})$

based on the rating pattern (Equations 6, 7). The user rating pattern space is clustered using K-Means clustering algorithm (line 3, Algorithm 1) in order to discover the rating bubbles.Each of the user clusters has a corresponding set of items rated by its users. We aim to cluster these items into n_i item clusters. However, in case of a user cluster with cardinality below the predefined threshold value (line 7, Algorithm 1), we do not cluster the item set. Rather, we assign the complete item set to the first item cluster with respect to that particular user cluster and generate the corresponding user-item rating matrix (lines 8-10, Algorithm 1). This step helps us to avoid over-fitting of user-clusters for small datasets. For a significantly large user-cluster, the set of items are further clustered using K-Means algorithm (Equation 7) into n_i item clusters (lines 13-18, Algorithm 1) such that the cluster maintains a sufficient number of items. The outputs of the algorithm are a set of user clusters, a set of item clusters and the associated rating matrix (line 21, Algorithm 1).

Gradient Optimization: The partial derivatives of the objective function (Equation 19) with respect to the latent features are derived in Equations 20 and 21.

$$\frac{\partial Z^c}{\partial U^c_u} = \sum_{i \in N^c} \boldsymbol{I}_{u,i} V^c_i g'(U^c_u V^{cT}_i)(g(U^c_u V^{cT}_i) - R^c_{u,i}) + \gamma^c_U U^c_u + k^c_U \Big(U^c_u - \frac{\sum_{s \in S^c_u} PrefSim(u,s) U^c_s}{\sum_{s \in S^c_u} PrefSim(u,s)} \Big) - k^c_U \frac{\sum_{\{s | u \in S^c_s\}} PrefSim(s,u) \Big(U^c_s - \frac{\sum_{p \in S^c_s} PrefSim(s,p) U^c_p}{\sum_{p \in S^c_s} PrefSim(s,p)} \Big)}{\sum_{\{s | u \in S^c_s\}} PrefSim(s,u)} \tag{20}$$

$$\frac{\partial Z^c}{\partial V^c_i} = \sum_{u \in M^c} \boldsymbol{I}_{u,i} U^c_u g'(U^c_u V^{cT}_i)(g(U^c_u V^{cT}_i) - R^c_{u,i}) + \gamma^c_V V^c_i + k^c_V \Big(V^c_i - \frac{\sum_{z \in S^c_i} Sim(i,z) V^c_z}{\sum_{z \in S^c_i} Sim(i,z)} \Big) - k^c_V \frac{\sum_{\{z | i \in S^c_z\}} Sim(z,i) \Big(V^c_z - \frac{\sum_{r \in S^c_z} Sim(z,r) U^c_r}{\sum_{r \in S^c_z} Sim(z,r)} \Big)}{\sum_{\{z | i \in S^c_z\}} Sim(z,i)} \tag{21}$$

$g'(x) = \frac{(\max_{u \in \mathbf{U}, i \in \mathbf{I}} R_{u,i} - \min_{u \in \mathbf{U}, i \in \mathbf{I}} R_{u,i}) e^{-x}}{(1 + e^{-x})^2}$, is the derivative of the logistic function.

Using Equations 20 and 21, U^c_u and V^c_i can be updated iteratively in the Gradient Optimization approach (eg: Gradient Descent, Stochastic Gradient Descent, etc.). The graphical representation for CREPE MF has been presented in Figure 1. We have incorporated preference network and item network in the basic PMF model and for an arbitrary cluster c trust propagation through similar users and similar items has been depicted.

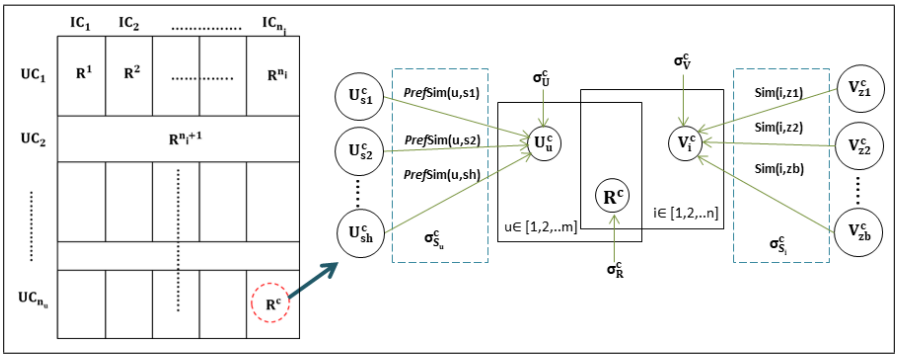

Figure 1: Graphical Representation of CREPE MF

Dataset Description

The data has been obtained from Yelp dataset challenge 2017 [1] which includes the business information as well as the interactions of the users with these businesses in the form of ratings and reviews. The businesses covered are broadly classified into roughly 1191 categories such as restaurants, real estate, doctors, religious organizations, etc. The users are spread across states in the United States of America (USA), Canada, Germany and the United Kingdom. In this study, we focus only on restaurant recommendations; therefore the original dataset has been subsetted to capture the restaurant-user interactions by including only those businesses that are tagged under the "Restaurants" category located in two big cities, Phoenix (Arizona, USA) and Toronto (Ontario, Canada). We have considered only those users who have given at least 10 reviews globally. The statistics of the datasets are given in Table 1.

Description	Phoenix	Toronto
# Users	14,198	8,082
# Businesses	3,233	6,193
#Friends	99,464	38,923
# Nodes in Preference-Network	197,564	153,887
%Friends who are similar	9.5%	14.3%
# Nodes in Item-Similarity Network	171,510	517,118
# Reviews	130,881	143,673
Ratings Sparsity	99.72%	99.71%

Table 1: Dataset Description

Yelp provides a large set of restaurant attributes which describes the restaurant's cuisine details and facilities available. However, many of these attributes are closely associated. To reduce this sparsity we group the attributes into ten broad categories, describing the restaurant's cuisines in a broader spectrum. For example: *American New, American Traditional, Bagels, Barbeque, Burger*, etc. are grouped into *'American'* category. For grouping the similar restaurant attributes into ten distinct categories, we employ two experts to independently categorize the attributes. Final category mappings were derived after mutual agreement between two experts. Information regarding users' visits to restaurants based on these categories are used to generate the *Preference Network*. For generating item similarity network, we build a binary vector of item attributes using these ten categories and compute the similarity using cosine similarity. The dataset also includes the users' social network connections (99,464 and 38,923 edges in Phoenix and Toronto dataset, respectively, Table 1). A careful investigation on the Yelp dataset reveals that on an average only a small percentage (9.5% in Phoenix and 14.3% in Toronto) of users' social network friends have similar preferences.

Experimental Details

In this section, we discuss the experimental results evaluating the accuracy and scalability of our algorithm, "CREPE MF". All the experiments have been performed on the datasets described in previous section and implemented in JAVA 1.8 on a Windows system with 3.4 GHz Intel i3 processor and 4 GB RAM. We have chosen datasets that are diverse in geography to ensure that the findings are not dataset dependent. The optimized parameters have been depicted in Table 2.

Evaluation Metrics

Accuracy: Two popular metrics are used for evaluation of accuracy: Root Mean Square Error (*RMSE*) and Mean Absolute Error (*MAE*).

[1]https://www.yelp.com/dataset_challenge

Parameters	Phoenix- Restaurants					Toronto- Restaurants					Phoenix - Other Businesses				
	I	II	III	IV	V	I	II	III	IV	V	I	II	III	IV	V
α	0.006	0.0012	0.001	0.0013	0.0028	0.005	0.0013	0.0013	0.0016	0.0018	0.011	0.0014	0.0009	0.001	0.0016
γ_U	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
γ_V	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
k_U	NA	5	0.000001	1.3	28	NA	1	0.000001	0.5	20	NA	15	0.000001	0.001	10
k_V	NA	NA	NA	10	10	NA	NA	NA	5	4	NA	NA	NA	0.00001	0.00001
# Iterations	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300

*NA implies the parameter is Not Applicable to the respective algorithm

I- PMF; II-Social MF; III- SocReg; IV- LBSMF; V- CREPE MF

Table 2: Specification of Parameters (α is the learning rate of Gradient Descent method)

$$RMSE = \sqrt{\frac{1}{|T|}\sum_{R_{u,i}\in T}(R_{u,i} - \hat{R}_{u,i})^2} \quad (22) \qquad MAE = \frac{1}{|T|}\sum_{R_{u,i}\in T}|(R_{u,i} - \hat{R}_{u,i})| \quad (23)$$

$|T|$ denotes the number of datapoints in the test dataset T. $R_{u,i}$ and $\hat{R}_{u,i}$ are the observed ratings and the predicted ratings by user u for item i, respectively. Lower *RMSE* and *MAE* values imply a better accuracy.

Scalability: We have used the average runtime as a metric of scalability where runtime refers to the time taken during model training. A scalable algorithm is expected to have lower runtime.

Experimental Findings

The proposed method *CREPE MF* has been compared with the following state-of-the-art algorithms to demonstrate its efficacy in terms of prediction accuracy and runtime.

1. Probabilistic Matrix Factorization (PMF) (Mnih and Salakhutdinov, 2008): This algorithm predicts missing ratings by factorizing the user-item matrix into two matrices of predetermined dimensions.
2. Social Matrix Factorization (SocialMF) (Jamali and Ester, 2010): This algorithm incorporates trust propagation through user's social network in the Probabilistic Matrix Factorization approach.
3. Social Regularization (SocReg) (Ma et al., 2011): This algorithm quantifies the similarity between users and their friends and accordingly incorporates a weighted social network trust propagation.
4. Location Based Social Matrix Factorization (LBSMF) (Yang et al., 2013): This algorithm extends SocReg algorithm by incorporating an item-item similarity network in the recommender system.

All the experiments have been performed with dimension of latent space (k) set to two standard values, 5 and 10. The datasets have been randomly divided with 80% in the training and remaining 20% in the test set. The experiments have also been repeated with a 90% and 10% division of the training and test set, respectively. The random selection was carried out 5 times independently and the average results are reported ensuring the robustness of our findings. The results are summarized in Table 3.

Evaluation of Accuracy: Table 3 shows that PMF has the highest RMSE, intuitively because the users and items are inherently assumed to be *i.i.d.* (independent and identically distributed). SocialMF and SocReg account for the user interdependence by including the effect of social network trust propagation resulting in a lower RMSE. LBSMF further improved accuracy by incorporating inter-item influence along with user's social network connections. These observations affirm that social networks and inter-item influence have a positive impact on recommendation accuracy. Gratifyingly, CREPE MF outperforms the other baseline algorithms. For latent space dimension of 5 and 80%-20% division of the dataset, CREPE MF achieves 28.54%, and 16.46% RMSE improvement compared to PMF and 12.97%, and 7.66% RMSE improvement compared to LBSMF for Phoenix and Toronto, respectively. Similar results have been observed for all other experimental settings. These outperformance of "CREPE MF" can be attributed to: i) enhancing the social network effect by including direct and second-hop connections with similar preferences and ii) building a homogeneous set of users and items by clustering them to discover the underlying rating bubbles. A further improvement in accuracy may be obtained through a cluster-specific hyperparameter tuning.
To test the statistical significance of performance of CREPE MF in terms of RMSE and MAE, paired t-test is performed for different dimensions of latent space (k= 5 and k= 10) and training-test split sets (80%-20% and 90%-10%). All the results obtained suggest that CREPE MF outperforms other benchmark models at 1% significance level. Due to space constraint, we have not reported the table here.

Dataset	Training	Metric	k = 5					k = 10				
			PMF	**SocialMF**	**SocReg**	**LBSMF**	**CREPE MF**	**PMF**	**SocialMF**	**SocReg**	**LBSMF**	**CREPE MF**
Phoenix-Restaurants	**80%**	RMSE	1.423	1.173	1.178	1.168	1.017	1.364	1.172	1.177	1.167	1.017
		Improvement	*28.54%*	*13.31%*	*13.68%*	*12.97%*		*25.44%*	*13.23%*	*13.59%*	*12.85%*	
		MAE	0.952	0.932	0.941	0.928	0.804	0.982	0.934	0.941	0.927	0.804
		Improvement	*15.55%*	*13.73%*	*14.56%*	*13.34%*		*18.13%*	*13.92%*	*14.56%*	*13.31%*	
	90%	RMSE	1.312	1.176	1.181	1.170	1.019	1.291	1.176	1.182	1.168	1.019
		Improvement	*22.34%*	*13.36%*	*13.73%*	*12.94%*		*21.07%*	*13.35%*	*13.79%*	*12.76%*	
		MAE	0.955	0.929	0.938	0.924	0.806	0.950	0.933	0.942	0.925	0.806
		Improvement	*15.59%*	*13.23%*	*14.06%*	*12.78%*		*15.13%*	*13.59%*	*14.41%*	*12.85%*	
Toronto-Restaurants	**80%**	RMSE	1.186	1.086	1.090	1.073	0.991	1.189	1.092	1.099	1.077	0.990
		Improvement	*16.46%*	*8.77%*	*9.10%*	*7.66%*		*16.74%*	*9.34%*	*9.92%*	*8.08%*	
		MAE	0.901	0.883	0.885	0.868	0.787	0.906	0.887	0.891	0.871	0.787
		Improvement	*12.61%*	*10.83%*	*11.03%*	*9.29%*		*13.13%*	*11.27%*	*11.67%*	*9.64%*	
	90%	RMSE	1.147	1.074	1.079	1.060	0.986	1.166	1.082	1.091	1.067	0.988
		Improvement	*14.04%*	*8.19%*	*8.62%*	*6.98%*		*15.27%*	*8.69%*	*9.44%*	*7.40%*	
		MAE	0.879	0.866	0.869	0.852	0.781	0.895	0.872	0.878	0.857	0.784
		Improvement	*11.15%*	*9.82%*	*10.13%*	*8.33%*		*12.40%*	*10.09%*	*10.71%*	*8.52%*	

Table 3: Accuracy Comparison with Benchmark Models

Evaluation of Run-time: An investigation of the runtime observations reveals that PMF has the lowest runtime, followed by SocialMF, SocReg, CREPE MF and LBSMF (Table 4). This order is in agreement with the increasing level of model complexity and it is maximum in case of LBSMF and CREPE MF where both social and item similarity networks are included. Nevertheless, CREPE MF has up to 29.60% lower runtime than LBSMF. It is worth noting that these results were obtained by running the CREPE MF clusters in series. However, a considerable reduction in run-time can be achieved by executing the clusters in parallel. For example, the run-time for the biggest cluster in Phoenix dataset (among 20 clusters) is 271.2 seconds which is much lower than SocialMF, SocReg and LBSMF (with 63.99%, 70.94% and 86.33% improvement respectively).

	Phoenix-Restaurants	Toronto-Restaurants
PMF	8.20	9.40
SocialMF	753.20	252.60
SocReg	933.20	433.60
LBSMF	1984.20	7007.80
CREPE MF	1752.60	4933.40

Table 4: Runtime Comparison (in sec)

Conclusion and Future Work

With the emergence of Web2.0, social recommender systems have gained an extensive attention from the academia as well as e-commerce communities. The underlying assumption of these models that all of the social connections of a user have an impact on user's decision does not hold true in many scenarios. In fact only a small fraction of user's social connections exhibit similar preferences to that of the active user. Therefore, we argue that incorporating social connections only with similar preference, referred as *Preference Network* in the model is highly rewarding and would drastically improve the system performance. Moreover, online users are heavily influenced by historical reviews and ratings and hence, high (low) rated items tend to receive high (low) ratings dividing the rating space into rating bubbles. In this work, we have disclosed a two-stage cluster based matrix factorization technique integrating *Preference Network* and inter-item similarity network into the PMF model, referred as "CREPE MF". The two-stage clustering on the rating space enables identification of the user and item rating bubbles as well as dimensionality reduction effectively.

The experimental results showed that CREPE MF outperforms the state-of-the-art recommender systems both in terms of prediction accuracy and run-time. The findings affirm that compared to social network, *Preference Network* can better capture and propagate the user's preference in the user latent feature matrix resulting in a lower RMSE (improvement up to 12.97%). Furthermore, two-stage clustering strategy reduces the rating-space to lower dimension and inherently improves putative data-sparsity and scalability issues as indicated by a lower runtime of CREPE MF (improvement up to 29.60%). In the current experimental settings, all the clusters have been executed in series. However, the clusters are independent and can be easily parallelized to achieve a reduced runtime. Extending the scope of recommending other businesses apart from restaurants is going in parallel and priliminary results are in favor of CREPE MF model. Further investigation on the sentiment of the users over restaurant attributes for enhancing the *Preference Network* is currently ongoing.

REFERENCES

Aral, S. 2014. "The problem with online ratings," *MIT Sloan Management Review* (55:2), p. 47.

Brzozowski, M. J., Hogg, T., and Szabo, G. 2008. "Friends and foes: ideological social networking," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* pp. 817–820.

Davidson, J., Liebald, B., Liu, J., Nandy, P., Van Vleet, T., Gargi, U., Gupta, S., He, Y., Lambert, M., Livingston, B., and Sampath, D. 2010. "The YouTube video recommendation system," *Proceedings of the fourth ACM conference on Recommender systems* pp. 293–296.

Dunbar, R. I. 2016. "Do online social media cut through the constraints that limit the size of offline social networks?" *Open Science* (3:1), p. 150,292.

Granovetter, M. S. 1973. "The strength of weak ties," *American journal of sociology* (78:6), pp. 1360–1380.

Hartigan, J. A., and Wong, M. A. 1979. "Algorithm AS 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society Series C (Applied Statistics)* (28:1), pp. 100–108.

Jamali, M., and Ester, M. 2010. "A matrix factorization technique with trust propagation for recommendation in social networks," *Proceedings of the fourth ACM conference on Recommender systems* pp. 135–142.

Lekakos, G., and Caravelas, P. 2008. "A hybrid approach for movie recommendation," *Multimedia Tools Appl* (36:1-2), pp. 55–70.

Linden, G., Smith, B., and York, J. 2003. "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," *IEEE Internet Computing* (7:1), pp. 76–80.

Lü, L., Medo, M., Yeung, C. H., Zhang, Y.-C., Zhang, Z.-K., and Zhou, T. 2012. "Recommender systems," *Physics Reports* (519:1), pp. 1–49.

Ma, H., Yang, H., Lyu, M. R., and King, I. 2008. "Sorec: social recommendation using probabilistic matrix factorization," *Proceedings of the 17th ACM conference on Information and knowledge management* pp. 931–940.

Ma, H., Zhou, D., Liu, C., Lyu, M. R., and King, I. 2011. "Recommender systems with social regularization," *Proceedings of the fourth ACM international conference on Web search and data mining* pp. 287–296.

Massa, P., and Avesani, P. 2004. "Trust-aware collaborative filtering for recommender systems," *CoopIS/DOA/ODBASE (1)* (3290), pp. 492–508.

Mnih, A., and Salakhutdinov, R. R. 2008. "Probabilistic matrix factorization," *Advances in neural information processing systems* pp. 1257–1264.

Newman, M., Barabasi, A.-L., and Watts, D. J. 2011. *The structure and dynamics of networks*, Princeton University Press.

Pham, M. C., Cao, Y., Klamma, R., and Jarke, M. 2011. "A clustering approach for collaborative filtering recommendation using social network analysis," *Journal of Universal Computer Science* (17:4), pp. 583–604.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. 2002a. "Incremental singular value decomposition algorithms for highly scalable recommender systems," *Fifth International Conference on Computer and Information Science* pp. 27–28.

Sarwar, B. M., Karypis, G., Konstan, J., and Riedl, J. 2002b. "Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering," *Proceedings of the fifth international conference on computer and information technology* (1).

Shepitsen, A., Gemmell, J., Mobasher, B., and Burke, R. 2008. "Personalized recommendation in social tagging systems using hierarchical clustering," *Proceedings of the 2008 ACM conference on Recommender systems* pp. 259–266.

Ungar, L. H., and Foster, D. P. 1998. "Clustering methods for collaborative filtering," *AAAI workshop on recommendation systems* (1), pp. 114–129.

Vespignani, A. 2009. "Predicting the behavior of techno-social systems," *Science* (325:5939), pp. 425–428.

Wang, J., Yin, J., Liu, Y., and Huang, C. 2011. "Trust-based collaborative filtering," *Fuzzy Systems and Knowledge Discovery (FSKD), 2011 Eighth International Conference on* (4), pp. 2650–2654.

Yang, D., Zhang, D., Yu, Z., and Wang, Z. 2013. "A sentiment-enhanced personalized location recommendation system," *Proceedings of the 24th ACM Conference on Hypertext and Social Media* pp. 119–128.

Zuo, X., Blackburn, J., Kourtellis, N., Skvoretz, J., and Iamnitchi, A. 2014. "The influence of indirect ties on social network dynamics," *International Conference on Social Informatics* pp. 50–65.