SCIENCE CHINA
Information Sciences

• **Supplementary File** •

Leveraging Implicit Social Structures for Recommendation via Bayesian Generative Model

Huafeng Liu[1,2], Jingxuan Wen[1,2], Liping Jing[1,2*] & Jian Yu[1,2]

[1]*Beijing Jiaotong University, No.3 Shangyuancun, Haidian District, Beijing,* 100044*, China;*
[2]*Beijing Key Lab of Traffic Data Analysis and Mining, No.3 Shangyuancun, Haidian District, Beijing,* 100044*, China*

Appendix A Theoretical Analysis

In this section, we introduce the min-max framework in statistical decision theory [1] to analyze the proposed model, and provide bounds on the risk of the **SoMA** estimator.

Appendix A.1 Notations

Here we define some notations used in following theoretical analysis.
Matrix and set: Given rating matrix $\mathbf{R} \in \mathbb{R}^{n\times m}$ and social matrix $\mathbf{S} \in \mathbb{R}^{n\times n}$, let $\mathbf{M} = [\mathbf{R}, \mathbf{S}] \in \mathbb{R}^{n\times(n+m)}$ be their row concatenation and $[\mathbf{M}]$ be its column space. Let $\mathbb{M}_d^{n\times\cdot}$ be a set of rank-d matrices with row dimensions n and arbitrary column dimensions, and $\mathbb{M}_d^{n\times m} \subseteq \mathbb{M}_d^{n\times\cdot}$ be its subset with column dimensions m. Let $\mathbb{G}_d^n$ be the Grassmannian defined as the set of d-dimensional subspace in $\mathbb{R}^n$. Note each element in $\mathbb{G}_d^n$ is a subspace. Let $\mathbb{O}_d^n$ be the set of orthonormal matrices in $\mathbb{R}^{d\times n}$.
Metric between subspaces: Suppose $S_1 \in \mathbb{G}_d^n$ and $S_2 \in \mathbb{G}_d^n$ are subspaces of $\mathbb{R}^n$. The *distance* between these two subspaces is defined as $d(S_1, S_2) = ||P_1 - P_2||_F$, where P_i is the orthogonal projection onto S_i. The distance between a pair of subspaces can be characterized by the blocks of a certain orthogonal matrix.
The min-max risk of SoMA estimator: The rating matrix $\mathbf{R}$ and social matrix $\mathbf{S}$ admit generalized factorization $\mathbf{R} = \boldsymbol{\theta}^\top\boldsymbol{\phi}$ and $\mathbf{S} = \boldsymbol{\theta}^\top\boldsymbol{\upsilon}$ where $\boldsymbol{\phi}$ and $\boldsymbol{\upsilon}$ are the corresponding transform matrices. Since $\mathbf{R}$ and $\mathbf{S}$ admit a unique $\boldsymbol{\theta}$ in factorization, we have a mapping $f : \mathbb{R}_d^{n\times\cdot} \to \mathbb{G}_d^n$ such that $f(\mathbf{R}) = \boldsymbol{\theta}$. Then, the **SoMA** estimator can be defined as $\hat{f} : \{\mathbf{M}_\Omega\} \to \mathbb{G}_d^n$ where $\Omega = \{\Omega_R, \Omega_S\}$ is the index set of observed entries in $\mathbf{R}$ and $\mathbf{S}$. The quality of mapping $\hat{f}$ can be evaluated via $l(\hat{f}|\mathbf{M}_\Omega) = \frac{1}{2}\big[d(\hat{f}(\mathbf{M}_\Omega), f(\mathbf{R})) + d(\hat{f}(\mathbf{M}_\Omega), f(\mathbf{S}))\big]$ and the maximum risk of **SoMA** estimator can be calculated by

$$\mathcal{R}(\hat{f}) = \sup \mathbb{E}_\Omega l(\hat{f}|\mathbf{M}_\Omega)$$

where expectation $\mathbb{E}_\Omega$ is taken over the randomness of $\mathbf{M}_\Omega$.
Packing: For any set S equipped with a distance metric d_S, let $\{S_w\}_{w\in W}$ be an arbitrary subset of S indexed by set W. For $\delta > 0$, we say this subset is a δ-packing of S with respect to d_S if $d_S(S_v, S_{v'}) \geqslant \delta$ whenever $v \neq v'$.
Hypothesis testing: To lower bound $\mathcal{R}(\hat{f})$, we employ the classic estimation-to-testing reduction method [2–4]. We design a new hypothesis testing problem. Let $\{S_w\}_{w\in W}$ be a 2δ-packing of $\mathbb{G}_d^n$ indexed by a finite set W, and $\mathcal{V}$ and $\mathcal{V}'$ be two random variables taking values w and w' from W. Thus the hypothesis testing problem is stated as following steps: (1) randomly choose $\mathcal{V}$ and $\mathcal{V}'$ from W independently and uniformly; (2) conditioned on $\mathcal{V}$ and $\mathcal{V}'$, randomly choose $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ satisfying $[\boldsymbol{\theta}] = S_w$ and $[\boldsymbol{\theta}'] = S_{w'}$; (3) generate $\mathbf{R} = \boldsymbol{\theta}^\top\boldsymbol{\phi}$ and $\mathbf{S} = \boldsymbol{\theta}^\top\boldsymbol{\upsilon}$ with some random matrices $\boldsymbol{\phi}$ and $\boldsymbol{\upsilon}$; then generate observation $\mathbf{M}_\Omega = [\mathbf{R}_{\Omega_R}, \mathbf{S}_{\Omega_S}]$. (4) apply a testing function $T : \{\mathbf{M}_\Omega\} \to W$ defined as $T(\mathbf{M}_\Omega) = \arg\min_{w\in W} d(\hat{f}(\mathbf{M}_\Omega), S_w)$.

Appendix A.2 A Bound for SoMA

Here we give the primary lower bound of **SoMA** estimator.

Proposition 1. Suppose $\mathbb{G}_d^n$ admits a 2δ-packing indexed by a finite set W, and $\mathcal{V}$ is a uniform random variable on W. Then, the **SoMA** estimator satisfies

$$\mathcal{R}(\hat{f}) \geqslant \frac{\delta}{2}\Big(1 - \frac{1}{|W|}(1 - p(T(\mathbf{M}_\Omega) \neq \mathcal{V}))\Big)$$

where the probability is defined over the random choice of $\mathbf{M}_\Omega = [\mathbf{R}_{\Omega_R}, \mathbf{S}_{\Omega_S}]$ and W.

* Corresponding author (email: lpjing@bjtu.edu.cn)

Proof. Write $\overline{\mathcal{V}}$ for $(\mathcal{V}, \mathcal{V}^{'})$ and $\overline{w}$ for $(w, w^{'})$. Let notation $\vee$ denote the logical disjunction. Following standard min-max arguments, we have

$$\begin{aligned}\sup \mathbb{E}_{\Omega}\big[d(\hat{f}(\mathbf{M}_{\Omega}), f(\mathbf{R})) + d(\hat{f}(\mathbf{M}_{\Omega}), f(\mathbf{S}))\big] &\geqslant \sup \mathbb{E}[\delta \cdot \mathbf{1}\{d(\hat{f}(\mathbf{M}_{\Omega}), f(\mathbf{R})) \geqslant \delta \vee d(\hat{f}(\mathbf{M}_{\Omega}), f(\mathbf{S})) \geqslant \delta\}] \\ &= \delta \cdot \sup p\{d(\hat{f}(\mathbf{M}_{\Omega}), f(\mathbf{R})) \geqslant \delta \vee d(\hat{f}(\mathbf{M}_{\Omega}), f(\mathbf{S})) \geqslant \delta\}\end{aligned} \tag{A1}$$

where the inequality is based on the fact that total distance is greater than δ if any one distance is greater than δ.

Reducing the above estimation problem into the hypothesis testing problem (with a 2δ-packing $\{f_w\}_{w\in W}$), we have

$$\begin{aligned}\sup p\{d(\hat{f}(\mathbf{M}_{\Omega}), f(\mathbf{R})) \geqslant \delta \vee d(\hat{f}(\mathbf{M}_{\Omega}), f(\mathbf{S})) \geqslant \delta\} \geqslant \frac{1}{|W|^2}\sum_{\overline{w}} p\{d(\hat{f}(\mathbf{M}_{\Omega}), f_w) \\ \geqslant \delta \vee d(\hat{f}(\mathbf{M}_{\Omega}), f_{w^{'}}) \geqslant \delta | \overline{\mathcal{V}} = \overline{w}\}\end{aligned}$$

where the coefficient is based on the uniform sampling assumption on W so that $p(\overline{\mathcal{V}} = \overline{w}) = \frac{1}{|W|^2}$.

$$\begin{aligned}&p\{d(\hat{f}(\mathbf{M}_{\Omega}), f_w) \geqslant \delta \vee d(\hat{f}(\mathbf{M}_{\Omega}), f_{w^{'}}) \geqslant \delta | \overline{\mathcal{V}} = \overline{w}\} \\ &= p\{d(\hat{f}(\mathbf{M}_{\Omega}), f_w) \geqslant \delta \vee d(\hat{f}(\mathbf{M}_{\Omega}), f_{w^{'}}) \geqslant \delta | w \neq w^{'}, \overline{\mathcal{V}} = \overline{w}\} \cdot p\{w \neq w^{'}, \overline{\mathcal{V}} = \overline{w}\} \\ &\quad + p\{d(\hat{f}(\mathbf{M}_{\Omega}), f_w) \geqslant \delta \vee d(\hat{f}(\mathbf{M}_{\Omega}), f_{w^{'}}) \geqslant \delta | w = w^{'} \overline{\mathcal{V}} = \overline{w}\} \cdot p\{w = w^{'}, \overline{\mathcal{V}} = \overline{w}\} \\ &= 1 \cdot p\{w \neq w^{'}, \overline{\mathcal{V}} = \overline{w}\} + p\{d(\hat{f}(\mathbf{M}_{\Omega}), f_w) \geqslant \delta \vee d(\hat{f}(\mathbf{M}_{\Omega}), f_{w^{'}}) \geqslant \delta | w = w^{'} \overline{\mathcal{V}} = \overline{w}\} \cdot p\{w = w^{'}, \overline{\mathcal{V}} = \overline{w}\}\end{aligned} \tag{A2}$$

where the second equality is based on the geometric argument, i.e. if $w \neq w^{'}$, then no $\hat{f}$ can be simutaneously δ-close to both f_w and $f_{w^{'}}$, which implies $p\{d(\hat{f}(\mathbf{M}_{\Omega}), f_w) \geqslant \delta \vee d(\hat{f}(\mathbf{M}_{\Omega}), f_{w^{'}}) \geqslant \delta\} = 1$.

Since $d(\hat{f}, f_w) \geqslant \delta$ as implied by $T(\mathbf{M}_{\Omega}) \neq w$ and average over all possible $\overline{w}$, we have

$$\begin{aligned}\sup \mathbb{E}_{\Omega}\big[d(\hat{f}(\mathbf{M}_{\Omega}), f(\mathbf{R})) + d(\hat{f}(\mathbf{M}_{\Omega}), f(\mathbf{S}))\big] &\geqslant \delta\Big(p(w \neq w^{'}) + \frac{1}{|W|} p\{T(\mathbf{M}_{\Omega}) \neq \mathcal{V} | \mathcal{V} = \mathcal{V}^{'}\}\Big) \\ &\geqslant \frac{\delta}{2}\Big(1 - \frac{1}{|W|}(1 - p(T(\mathbf{M}_{\Omega}) \neq \mathcal{V}))\Big)\end{aligned} \tag{A3}$$

Now we will give a detailed lower bound covering the recovery error.

Proposition 2. Let S, $S^{'}$ respectively be the column space of any $\mathbf{R}, \mathbf{R}^{'} \in \mathbb{M}_d^{n\times m}$. Let $s_d(\mathbf{R})$ be the smallest non-zero singular value of $\mathbf{R}$. Then

$$d(S, S^{'}) \leqslant \frac{\sqrt{2}||\mathbf{R} - \mathbf{R}^{'}||_F}{s_d(\mathbf{R})} \tag{A4}$$

For simplicity, we focus on a set $\widetilde{\mathbb{M}}_d^{n\times \cdot} \subseteq \mathbb{M}_d^{n\times \cdot}$ whose matrices have their smallest non-zero singular values bounded away from zero.

Let $\widetilde{\mathbb{G}}_d^n \subseteq \mathbb{G}_d^n$ be the set induced from $\widetilde{\mathbb{M}}_d^{n\times \cdot}$ such that for every $S \in \widetilde{\mathbb{G}}_d^n$ there is an matrix $\mathbf{R} \in \widetilde{\mathbb{M}}_d^{n\times \cdot}$ satisfying $f(\mathbf{R}) = S$. For a matrix $\mathbf{R}$ and a latent factor $\hat{\boldsymbol{\theta}}$ estimated from its observation $\mathbf{R}_{\Omega}$, we define the recovery error as

$$E_{\mathbf{R}}(\hat{\boldsymbol{\theta}}) = \min ||\mathbf{R} - \hat{\boldsymbol{\theta}}^{\top}\boldsymbol{\phi}||_F^2 \tag{A5}$$

and we can define the recovery loss for both rating matrix and social matrix as $l(\hat{f}|\mathbf{M}_{\Omega}) = \frac{1}{2}[E_{\mathbf{R}}(\hat{f}(\mathbf{M}_{\Omega}) + E_{\mathbf{S}}(\hat{f}(\mathbf{M}_{\Omega})]$ and the maximum risk of **SoMA** estimator as $\mathcal{R}(\hat{f}) = \sup \mathbb{E}_{\Omega} l(\hat{f}|\mathbf{M}_{\Omega})$. Thus, the recovery bound is given as follows

Theorem 1. Given a $\widetilde{\mathbb{M}}_d^{n\times \cdot}$ and its induced $\widetilde{\mathbb{G}}_d^n$ that admits a 2δ-packing indexed by a finite set W, Let $\mathcal{V}$ be a uniform random variable on W. Then, there is a constant $c > 0$ (depending on $\widetilde{\mathbb{M}}_d^{n\times \cdot}$) bounded away from zero such that **SoMA** estimator $\hat{f}$ satisfies

$$\mathcal{R}(\hat{f}(\mathbf{M}_{\Omega}) \geqslant \frac{c\delta}{2\sqrt{2}}\Big(1 - \frac{1}{|W|}(1 + p(T(\mathbf{M}_{\Omega}) \neq \mathcal{V}))\Big) \tag{A6}$$

where the probability defined over the random choice of $\mathbf{M}_{\Omega} = [\mathbf{R}_{\Omega_R}, \mathbf{S}_{\Omega_S}]$ and W.

Proof. Note that $s_d(\mathbf{R})$ is the smallest non-zero singular value of matrix $\mathbf{R}$. Constant $c = \inf_{\mathbf{R}\in\widetilde{\mathbb{M}}_d^{n\times \cdot}} s_d(\mathbf{R})$ is positive and bounded away from zero. Combing with Proposition 2, this implies that any matrix $\mathbf{R}$, $\mathbf{R}^{'} \in \widetilde{\mathbb{M}}_d^{n\times \cdot}$ satisfy $d(S, S^{'}) \leqslant \frac{\sqrt{2}||\mathbf{R}-\mathbf{R}^{'}||_F}{c}$ and $E_{\mathbf{R}}(\hat{f}(\mathbf{M}_{\Omega})) \geqslant \frac{c\cdot d(\hat{f}(\mathbf{M}_{\Omega}), f(\mathbf{R}))}{\sqrt{2}}$. Hence over all $\mathbf{M} \in \widetilde{\mathbb{M}}_d^{n\times \cdot} \times \widetilde{\mathbb{M}}_d^{n\times \cdot}$, we have

$$\sup \mathbb{E}_{\Omega}[E_{\mathbf{R}}(\hat{f}(\mathbf{M}_{\Omega})) + E_{\mathbf{S}}(\hat{f}(\mathbf{M}_{\Omega}))] \geqslant \frac{c}{\sqrt{2}} \sup \mathbb{E}_{\Omega}\big[d(\hat{f}(\mathbf{M}_{\Omega}), f(\mathbf{R})) + d(\hat{f}(\mathbf{M}_{\Omega}), f(\mathbf{S}))\big]$$

Applying proposition 1, we can obtain the bound stated above.

Theorem 1 shows a recovery error bound that maintains the same order as the estimation error bound.

Appendix B Scalable Stochastic Variational Inference for SoMA

The key problem in the parameter estimation of Bayesian model is computing the posterior distribution over the latent variables given the observed data. For **SoMA**, it is intractable to obtain the exact inference of the posterior distribution, thus, we adopt mean-field stochastic variational Bayesian method to approximate the full posterior distribution.

Appendix B.1 Variational Inference for SoMA

Variational methods aim to approximate the posterior distribution (i.e. $p(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{z}, \boldsymbol{\gamma}, \boldsymbol{\pi}|\mathbf{R}, \mathbf{S}, \alpha, \beta, \lambda, \sigma_{\phi}, \sigma_R)$) via a factorized form

$$q(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{z}, \boldsymbol{\gamma}, \boldsymbol{\pi}) = \prod_{i=1}^{n} q(\boldsymbol{\gamma}_i|\eta_i) \prod_{c=1}^{d} q(\boldsymbol{\pi}_c|\delta_c) \prod_{i=1}^{n}\prod_{c=1}^{d} q(\boldsymbol{\theta}_{ic}|\zeta_{ic}) \times \prod_{j=1}^{m} q(\boldsymbol{\phi}_j|\mu_j, \sigma^2) \prod_{<i,k>}\prod_{c=1}^{d} q(\boldsymbol{z}_{ikc}|\boldsymbol{\xi}_{ic}) \tag{B1}$$

with $q(\boldsymbol{\pi}_c|\delta_c) = Beta(\boldsymbol{\pi}_c; \delta_c)$, $q(\boldsymbol{\theta}_{ic}|\zeta_{ic}) = Beta(\boldsymbol{\theta}_{ic}; \zeta_{ic})$, $q(\boldsymbol{\gamma}_i|\eta_i) = Exp(\boldsymbol{\gamma}_i; \eta_i)$ $q(\boldsymbol{\phi}_j|\mu_j, \sigma^2) = N(\boldsymbol{\phi}_j; \mu_j, \sigma^2)$ and $q(\boldsymbol{z}_{ikc}|\xi_{ic}) = B((\boldsymbol{z}_{ikc}; \xi_{ic})$. The variational inference can be implemented by minimizing the Kullback-Leibler (KL) divergence between the true posterior $p(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{z}, \boldsymbol{\gamma}, \boldsymbol{\pi})$ and the factorized variational distribution $q(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{z}, \boldsymbol{\gamma}, \boldsymbol{\pi})$. It has been proved that minimizing the KL divergence is equal to optimizing an *Evidence Lower BOund* (ELBO) $\mathcal{L}$:

$$\begin{aligned}
\mathcal{L} &= \mathbb{E}_q[\log p(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{z}, \boldsymbol{\gamma}, \boldsymbol{\pi}, \mathbf{R}, \mathbf{S})] - \mathbb{E}_q[\log q(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{z}, \boldsymbol{\gamma}, \boldsymbol{\pi})] \\
&= \sum_{i=1}^{n}\sum_{c=1}^{d} \mathbb{E}_q[\log p(\boldsymbol{\theta}_{ic}|\alpha)] - \mathbb{E}_q[\log q(\boldsymbol{\theta}_{ic}|\zeta_{ic})] + \sum_{j=1}^{m} \mathbb{E}_q[\log p(\boldsymbol{\phi}_j|\sigma_{\phi})] - \mathbb{E}_q[\log q(\boldsymbol{\phi}_j|\mu_j, \sigma^2)] \\
&+ \sum_{i=1}^{n} \mathbb{E}_q[\log p(\boldsymbol{\gamma}_i|\lambda)] - \mathbb{E}_q[\log q(\boldsymbol{\gamma}_i|\eta_i)] + \sum_{c=1}^{d} \mathbb{E}_q[\log p(\boldsymbol{\pi}_c|\beta)] - \mathbb{E}_q[\log q(\boldsymbol{\pi}_c|\delta_c)] + \sum_{\mathbf{S}_{ik}} \mathbb{E}_q[\log p(\mathbf{S}_{ik}|\boldsymbol{z}_{ik}, \boldsymbol{\pi})] \\
&+ \sum_{<i,k>}\sum_{c=1}^{d} \mathbb{E}_q[\log p(\boldsymbol{z}_{ikc}|\boldsymbol{\theta}_{ic})] - \mathbb{E}_q[\log q(\boldsymbol{z}_{ikc}|\boldsymbol{\xi}_{ic})] + \sum_{\mathbf{R}_{ij}\neq 0} \mathbb{E}_q[\log p(\mathbf{R}_{ij}|\boldsymbol{\theta}_i, \boldsymbol{\phi}_j, \sigma_R)]
\end{aligned} \tag{B2}$$

All these variables can be roughly divided into *global* hidden variables and *local* hidden variables [5]. The first three lines in the right of (B2) contain the summations over all user groups, users, and items, thus they are denoted as global terms, and the related variables $(\boldsymbol{\zeta}, \boldsymbol{\mu}, \boldsymbol{\eta}, \boldsymbol{\delta})$ are called as global variables. The remaining parts are local terms, and $\boldsymbol{\xi}$ is taken as local variable. Next, we will optimize ELBO via stochastic optimization.

Appendix B.2 Stochastic Optimization

Stochastic variational inference is a coordinate ascent algorithm that iteratively updates local and global variables. For each iteration, given the current settings of the global variables, we subsample the rating information $\mathbf{R}$ and compute optimal local variables. Then the global variables can be updated by using the stochastic natural gradient which is computed from the subsampled data and local variables.

For (B2), the gradients of ELBO with respect to the global variables $\boldsymbol{\zeta}, \boldsymbol{\mu}, \boldsymbol{\eta}, \boldsymbol{\delta}$ can be calculated via

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \zeta_{ic,1}} &= \alpha_1 - \zeta_{ic,1} + \sum_{\mathbf{R}_{ij}\in D} \boldsymbol{\theta}_i^{\top}\boldsymbol{\phi}_j + \sum_{k=1}^{n}\sum_{c=1}^{d} \mathbf{S}_{ik}\boldsymbol{\theta}_{ic} + \sum_{k=1}^{n}\sum_{c=1}^{d}(1-\mathbf{S}_{ik})\boldsymbol{\theta}_{ic} \\
\frac{\partial \mathcal{L}}{\partial \mu_j} &= -\frac{\mu_j}{\sigma_{\phi}^2} + \sum_{\mathbf{R}_{ij}\in D}(\mathbf{R}_{ij} - \boldsymbol{\theta}_i^{\top}\boldsymbol{\phi}_j) \qquad \frac{\partial \mathcal{L}}{\partial \delta_{c,1}} = -\delta_{c,1} + \beta_1 + \sum_{k=1}^{n}(\mathbf{S}_{ik} - g(\hat{\mathbf{S}}_{ik}))g^{'}(\hat{\mathbf{S}}_{ik}) \\
\frac{\partial \mathcal{L}}{\partial \eta_i} &= -\eta_i + \lambda + \sum_{\mathbf{R}_{ij}\in D}(\mathbf{R}_{ij} - \boldsymbol{\theta}_i^{\top}\boldsymbol{\phi}_j) + \sum_{k=1}^{n}(\mathbf{S}_{ik} - g(\hat{\mathbf{S}}_{ik}))g^{'}(\hat{\mathbf{S}}_{ik})
\end{aligned}$$

where the mini-batch of observed rating set is denoted as $D = \{(i,j) : R_{ij} > 0\}$. Then we can update these variables via stochastic gradient ascent technique with different learning rates $(\rho_i^{(\tau)}, \rho_j^{(\tau)}, \rho_c^{(\tau)})$ for different variables. Note $\rho^{(\tau)} = (\tau_0 + \tau)^{-\kappa}$, where $\kappa \in (0.5, 1]$ is the learning rate and $\tau_0 \leqslant 0$ downweights early iterations. In a similar way, we can obtain natural gradient for $\zeta_{ic,2}$, stochastic gradient for σ and $\delta_{c,2}$, we omit it because of page limitation.

The local parameter $\boldsymbol{\xi}$ can be updated via coordinate ascent technique:

$$\boldsymbol{\xi}_{ic} \propto \exp\Big\{\mathbb{E}_q[\log \boldsymbol{\theta}_{ic}\boldsymbol{\gamma}_i] + \sum_{k=1}^{n} \mathbf{S}_{ik}\mathbb{E}_q[\log \pi_c] + \sum_{\mathbf{R}_{ij}\in D}(\mathbf{R}_{ij} - \boldsymbol{\theta}_i^{\top}\boldsymbol{\phi}_j)\Big\}. \tag{B3}$$

By iteratively updating these variables until coverage, we can get the stable results. The full stochastic variational inference is summarized in Algorithm B1.

Algorithm B1 The stochastic variational inference for **SoMA**

Input: Rating matrix $\mathbf{R}$ with n users and m items, social matrix $\mathbf{S}$, initialized variational parameters.
while ELBO is not converged **do**
  Sample a mini-batch D of rating matrix.
  Update local variables
    Update ξ_{ic} for each user i related to D via (B3)
  Update global variables
    Update ζ_{ic} for each user i related to D;
    Update μ_j for each item j related to D;
    Update η_i for each user i related to D;
    Update δ_c for each group c;
  Set $\rho_i^{(\tau)}$ for each user, $\rho_j^{(\tau)}$ for each item, and $\rho_c^{(\tau)}$ for each group; $\tau = \tau + 1$.
end while
Output: stable value of latent variables $\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{z}, \boldsymbol{\gamma}, \boldsymbol{\pi}$.

Complexity analysis: The main computation complexity of **SoMA** is to iteratively update global and local variables. **SoMA** has to model both observed social relations connecting each user and unobserved social relations about the current user. In order to alleviate the computational complexity, we randomly sample a subset of unobserved social relations. Updating the local variables costs $O(d(nnz(\mathbf{R}) + nnz(\mathbf{S}) + r(n^2 - nnz(\mathbf{S}))))$, where r is the sampling ratio for unobserved social relations. Updating the global variables will cost $O(d(n + m))$. The total complexity of **SoMA** is $O(td(nnz(\mathbf{R}) + nnz(\mathbf{S}) + r(n^2 - nnz(\mathbf{S}))))$, where t is the number of iterations.

Appendix C Experiments

In this section, we evaluate the proposed **SoMA** on four datasets by comparing with existing methods.

Appendix C.1 Experimental Setting

Datasets: In experiments, four widely used social recommendation datasets, *Ciao*[1)], *Epinions*[2)], *Douban*[3)] and *Yelp*[4)], are used to test the recommendation performance. Their ratings are ordinal values on the scale 1 to 5, more information is summarized in Table C1.

Table C1 Experimental datasets.

	Ciao	*Epinions*	*Douban*	*Yelp*		*Ciao*	*Epinions*	*Douban*	*Yelp*
♯ users (n)	7,375	49,290	129,490	1,182,626	♯ relations	111,781	487,183	1,692,952	13,811,526
♯ items (m)	106,997	139,738	58,541	156,638	SDensity	0.2055%	0.0201%	0.0202%	0.0009%
♯ ratings	284,086	664,824	16,830,839	4,731,265					
RDensity	0.036%	0.010%	0.222%	0.0026%					

Metrics for Rating Prediction: Two well-known evaluation metrics, mean absolute error $(\mathrm{MAE} = \sum_{(i,j)\in R_t} |\mathbf{R}_{ij} - \hat{\mathbf{R}}_{ij}|/|R_t|)$ and root mean square error $(\mathrm{RMSE} = \sqrt{\sum_{(i,j)\in R_t} (\mathbf{R}_{ij} - \hat{\mathbf{R}}_{ij})^2/|R_t|})$ are adopted to evaluate the prediction accuracy, where R_t is the testing set. $\mathbf{R}_{ij}$ and $\hat{\mathbf{R}}_{ij}$ are the ground truth and predicted rating given by the i-th user to the j-th item. Smaller RMSE and MAE values indicate better result. Five-fold cross-validation technique is used and their averaged results are reported.

Baselines: Eleven existing methods are used as baselines. Among them, PMF [6] is a classical probabilistic matrix factorization based collaborative filtering method, and NCF [7] is a deep neural networks based collaborative filtering method. Locabal [8], TrustSVD [9] and SIACC [10] consider the explicit social relations. PSLF [11], MFC [12], SoDimRec [13], and UniWalk [14] consider implicit social relations and are implemented via a two-stage strategy. NSCR [15], DeepRec [16] and GraphRec [17] are social recommendation methods modeling user and item features via deep neural networks. Note that we also conducted experiments on SoRec [18], TrustMF [19], SocialMF [20] and SoReg [21]. However, we do not list the experimental results since they were proved inferior to the selected baselines.

Parameter setting: The optimal experimental settings for each method are determined either by experiments or suggested by previous works. For **SoMA**, the hyperparameters are set with $\alpha_1 = 0.5, \alpha_2 = 1, \beta_1 = 5, \beta_2 = 1, \lambda = 2, \sigma_\phi = 0.5, \sigma_R = 10$ for all datasets. The learning parameters for inference algorithm are set with $\tau_0 = 50000, \kappa = 0.5$.

1) https://www.cse.msu.edu/ tangjili/trust.html
2) http://www.trustlet.org/downloaded_epinions.html
3) https://www.cse.cuhk.edu.hk/irwin.king.new/pub/data/douban
4) https://www.yelp.com/dataset/challenge

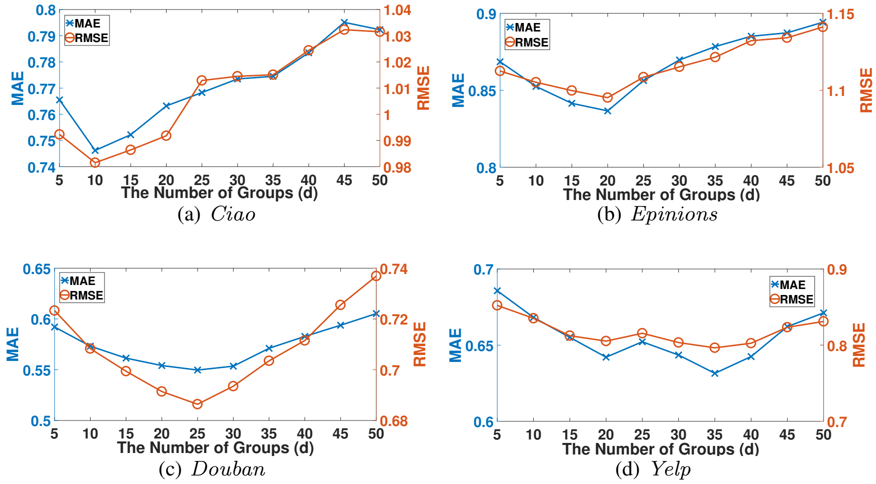


Figure C1 Effect of d (the number of user groups) on **SoMA** for (a) *Ciao*, (b) *Epinions*, (c) *Douban*, and (d) *Yelp*.

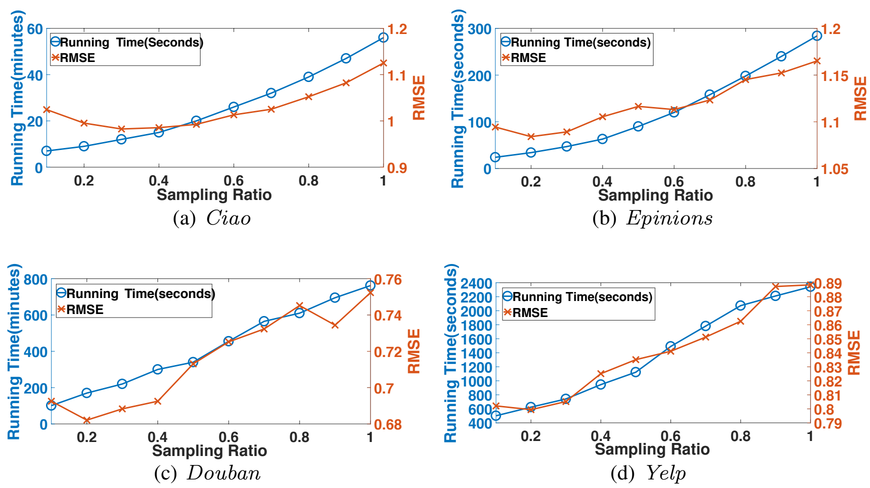


Figure C2 Effect of sampling ratio r (to select the unobserved social relations for each user) on **SoMA** in terms of running time (seconds) and RMSE for (a) *Ciao*, (b) *Epinions*, (c) *Douban*, and (d) *Yelp*.

Appendix C.2 Results and Discussion

In this section, we investigate **SoMA** from several views. Firstly, a series of experiments are conducted to test the effects of two parameters (the number of user groups d and the sampling ratio for unobserved relations r). Secondly, **SoMA** is compared with baselines from two facets including *All Users* and *Near-cold-start Users* in term of rating and ranking prediction. *All Users* indicates that all ratings are used as the testing set. *Near-cold-start Users* view means that the users who rate less than five items will be involved in the testing set. Thirdly, we investigate how the social relations affect the corresponding social recommendation methods and the effect of global social structures. Fourthly, the generalization ability analysis demonstrates the efficiency for **SoMA**. Then, we analysis model scalability and complexity by comparing with several comparative social recommendation methods. Finally, we show the interpretable recommendation results provided by **SoMA** with the aid of auxiliary information.

Appendix C.2.1 *Effect of parameters*

The first experiment is conducted to demonstrate the effect of number of social groups (i.e., latent space size d) on the proposed model. Figure C1 shows the effect of number of user groups d on four datasets in term of MAE and RMSE. For each dataset, **SoMA** performs better as d increases, reaches the best value, and then decreases in performance as d grows larger. Note that larger d does not output better recommendation performance. The main reason is that the observed rating matrix is extremely sparse. Increasing the number of groups would keep splitting the large user group into more focused small groups, which may lead to data overfitting and further destroy the recommendation performance. Meanwhile, a larger value d aggravates the computational complexity. It can be seen that the optimal d in *Ciao*, *Epinions*, *Douban*, and *Yelp* are 10, 20, 25 and 30 respectively, and they are almost proportional to the number of users.

For each user, **SoMA** has to model not only the observed social relations connecting this user, but also the unobserved social relations about him/her. From Table C1, we can see that each user only has few observed social relation (e.g., in *Yelp*, $\overline{s}$ is 12 for each user, however there are total $1,182,626$ users). To efficiently handle this, we adopt sampling technique to select a small subset of unobserved social relations for each user by a fixed ratio r (from 0.1 to 1 with step 0.1). Figure C2 lists the running time and prediction accuracy under varying the sampling ratio. This demonstrates that considering unobserved social relation is beneficial to social recommendation. However, they aggravate the computational complexity and may introduce noisy social relations when considering more unobserved social relations. Fortunately, the performance becomes better when the sampling ratio is in [0.2,0.3] and the computational complexity is acceptable.

Table C2 Comparing different recommendation methods on testing *All Users* and *Near-cold-start Users*.

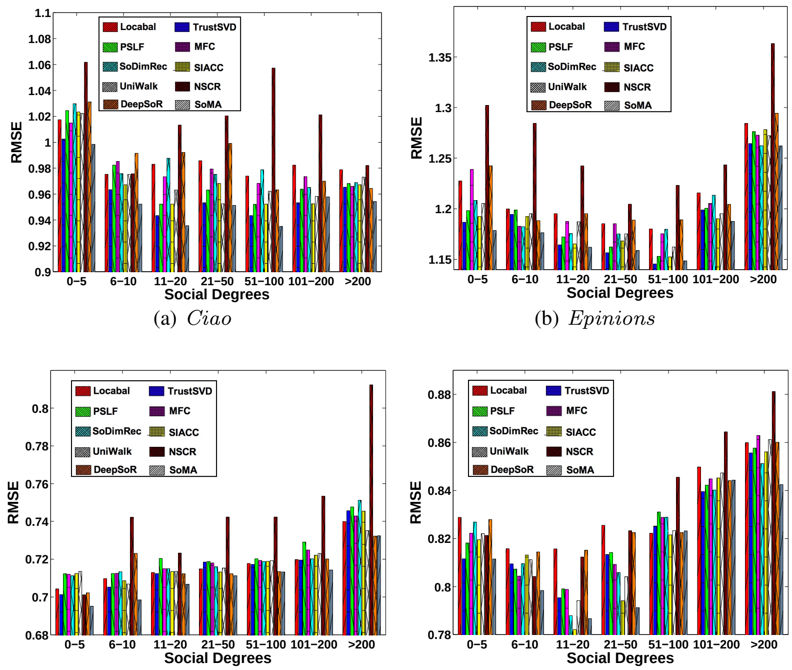
	Datasets	*Ciao*		*Epinions*		*Douban*		*Yelp*	
	Metrics	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
All Users	PMF	0.8257	1.1310	1.1206	1.3654	0.6230	0.7699	0.6684	0.8557
	NCF	0.7832	1.0189	1.0433	1.2894	0.6046	0.7432	0.6686	0.8455
	Locabal	0.7561	1.0214	0.9374	1.1548	0.5756	0.7190	0.6572	0.8331
	TrustSVD	$\underline{0.7516}$	$\underline{0.9854}$	0.9145	1.1332	$\underline{0.5563}$	$\underline{0.6985}$	$\underline{0.6393}$	$\underline{0.8116}$
	PSLF	0.7654	0.9963	0.9251	1.1486	0.5731	0.7126	0.6448	0.8252
	MFC	0.7601	0.9865	0.9257	1.1495	0.5798	0.7197	0.6512	0.8271
	SoDimRec	0.7694	0.9995	0.9305	1.1524	0.5751	0.7148	0.6537	0.8296
	SIACC	0.7554	1.0138	$\underline{0.8515}$	$\underline{1.1115}$	0.5679	0.7183	0.6531	0.8287
	UniWalk	0.7525	0.9868	0.9042	1.1251	0.5636	0.7163	0.6542	0.8313
	NSCR	0.7765	1.0044	1.0142	1.2643	0.5916	0.7325	0.6496	0.8297
	DeepSoR	0.7639	0.9973	0.9322	1.1563	0.5763	0.7173	0.6453	0.8239
	SoMA	**0.7451**	**0.9813**	**0.8357**	**1.0943**	**0.5484**	**0.6857**	**0.6307**	**0.7957**
Near-cold-start Users	PMF	0.9344	1.1877	1.3508	1.4477	0.8433	1.0239	0.8752	1.0722
	NCF	0.8732	1.1233	1.2836	1.3954	0.7733	0.9631	0.7922	0.9912
	Locabal	0.8296	1.0775	1.0877	1.2651	0.7244	0.8836	0.7487	0.9152
	TrustSVD	0.7954	1.0135	$\underline{1.0532}$	1.2452	0.6751	0.8453	0.7228	0.8952
	PSLF	0.8122	1.0321	1.0721	1.2762	0.7011	0.8685	0.7412	0.9122
	MFC	0.8134	1.0334	1.0702	1.2678	0.7093	0.8766	0.7385	0.9025
	SoDimRec	0.8278	1.0523	1.0689	1.2606	0.6909	0.8606	0.7461	0.9194
	SIACC	0.8094	1.0021	1.0656	$\underline{1.1971}$	0.6389	0.8323	0.7523	0.9253
	UniWalk	0.8188	1.0414	1.0677	1.2322	0.6935	0.8685	0.7344	0.9011
	NSCR	0.8517	1.0893	1.2534	1.3499	0.7534	0.9354	0.7814	0.9689
	DeepSoR	0.8333	1.0821	1.2432	1.3663	0.7364	0.8946	0.7659	0.9381
	GraphRec	$\underline{0.8041}$	$\underline{1.0016}$	1.0561	1.2034	$\underline{0.6313}$	$\underline{0.8296}$	$\underline{0.7269}$	$\underline{0.9008}$
	SoMA	**0.7859**	**0.9988**	**1.0506**	**1.1890**	**0.6278**	**0.8256**	**0.7147**	**0.8838**

Appendix C.2.2 *Recommendation performance*

A series of experiments are conducted to compare the proposed **SoMA** with several baselines on *All Users* and *Near-cold-start Users* (in terms of RMSE and MAE), as shown in Table C2. The best and second results are marked in bold and underlined, respectively. As expected, social recommendation methods work better than recommendation method with solely rating information (PMF and NCF) which only exploits the rating information in most cases. Although NCF and PMF only utilize the rating information, NCF obtains much better performance than PMF since the power of neural networks. From the view of *All Users*, TrustSVD outperforms others on *Ciao*, *Douban* and *Yelp* dataset. The main reason is that TrustSVD sufficiently exploits the social relations with two strategies: adjusting the user latent factor with the user's social neighbors, and enforcing the rating matrix and social matrix share the same user latent factor. SIACC performs well on *Epinions* dataset because it simultaneously considers global and local social relations when determining the co-clusters among users and items. Note that deep social recommendation methods (NSCR, DeepSoR and GraphRec) also achieve comparative performance due to the non-linear feature learning.

Obviously, **SoMA** performs better than the existing social recommendation methods, which confirms that considering both observed and unknown social relations are helpful to extract social structure and leverage training the social recommendation model. Even though SoDimRec and UniWalk make use of observed and unobserved social relations, they adopted a two-phase separated strategy, i.e., they firstly mine the indirect social relations and then integrate them to matrix factorization based social recommendation model. Such strategy will ignore the interaction between social structure identification and recommendation model training, which results in that these two methods are inferior to the proposed unified Bayesian learning framework **SoMA**. Moreover, although PSLF learns latent user factors and indirect social relations in a unified model, **SoMA** outperforms PSLF on all datasets. Unlike *Dirichlet* distribution used in PSLF, **SoMA** adopts *Beta* distribution to model community membership, which can make sure that each user has large affiliation strengths with multiple communities if he or she are simultaneously and strongly related to these communities.

Although the relative improvements are small, small improvements can lead to significant differences of recommendations in practice [22]. Thus, we conduct t-test between **SoMA** and each baseline with five-fold cross-validation results. The p-



(a) *Ciao* (b) *Epinions*

(c) *Douban* (d) *Yelp*

Figure C3 Comparison of social recommendation methods with different social degrees.

values in all cases are less than 10^{-5}, which indicates that the improvements are statistically significant at the 5% level. Therefore, based on these observations, we can say **SoMA** consistently outperforms the state-of-the-art baselines and significantly improves the recommendation performance.

Appendix C.2.3 *Effect of social information*

In order to investigate how the social relationships affect the social recommendation with different strategies, we analyze the social relationships from the perspective of social degrees. Specifically, all users are split into seven subsets according to their social degrees (i.e., the number of user's neighbors in social network): 0-5, 6-10, 11-20, 21-50, 51-100, 101-200, and >200. All social recommendation methods are compared in seven user-subsets with different social degrees for each dataset, as shown in Figure C3 (in terms of RMSE). It is interesting that all methods on each dataset have the similar trends with respect to different social degrees, which confirms that social degree plays an important role in social recommendation systems. As expected, **SoMA** obtains the best performance for all social degrees on all datasets because of its great strategy to sufficiently exploit social information. From Figure C3, it can be seen that all methods on *Douban* and *Ciao* have different trends from other two datasets. Recall the statistic information in Table C1, the rating information in *Douban* is much denser than others, while the social information in *Ciao* is much denser than others. This result indicates that social recommendation performance may be affected by the denser information to a larger extent. Especially, the overall RMSE becomes better and better with the increasing of social degree on *Ciao* datasets, which implies that social relations positively promote recommendation performance. However, all methods output worse performances on the users with higher social degrees (>200) (except for *Ciao*). One possible reason is that the social ties of such users are casual because they tend to add more and more friends without real intentions, and this kind of behaviors may destroy the quality of social information.

Secondly, the social recommendation with implicit social structures (MFC, SoDimRec, PSLF and **SoMA**) are investigated by analyzing the properties of the mined implicit social structures. From the experimental results, it can be seen that the proposed **SoMA** is superior to MFC, SoDimRec and PSLF. The main difference between them is the processing of determining implicit social structures. MFC and SoDimRec can be taken as two-phase recommendation framework, where the community detection approach (Bigclam [23]) is adopted to identify the implicit social structures from the social network in the first phase, and then the learned implicit social struuresct are employed into the recommendation phase. In other words, MFC and SoDimRec only take the social network into account for mining the social relations. **SoMA** and PSLF utilized both social network and rating information to identify the implicit social structures (such social structure are used to learn the user and item latent factors), meanwhile, the learned latent factors affect the implicit social structures identification. In this case, we can say **SoMA** and PSLF are unified model, rather than a two-phase process like MFC and SoDimRec, while the difference between **SoMA** and PSLF is the generative process in modeling implicit social structure.

Thus, in experiments, we investigated the user communities (i.e., the implicit social structures) properties learned from MFC, SoDimRec, PSLF and **SoMA**. (Note that MFC and SoDimRec have the same user communities results because

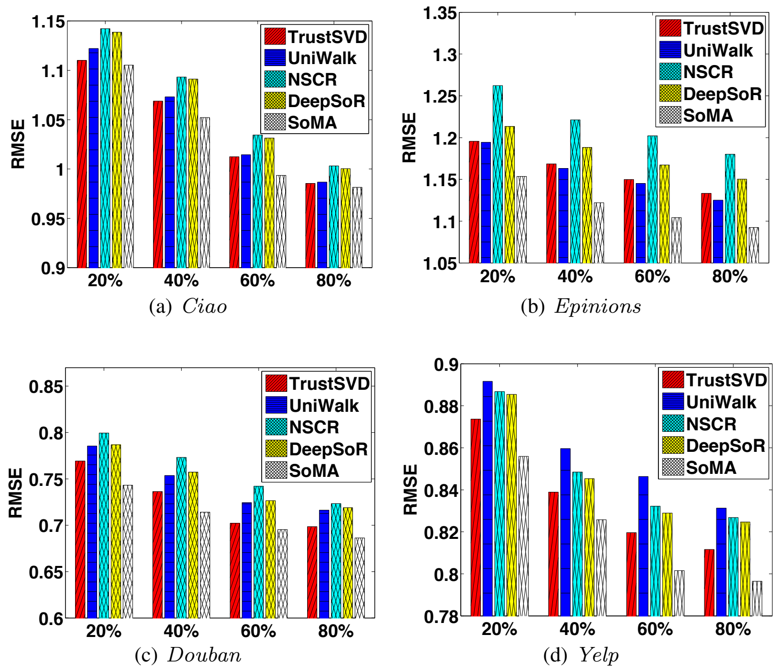


Figure C4 RMSE comparison of TrustSVD, UniWalk, NSCR, DeepSoR and **SoMA** in terms of different size of training sets.

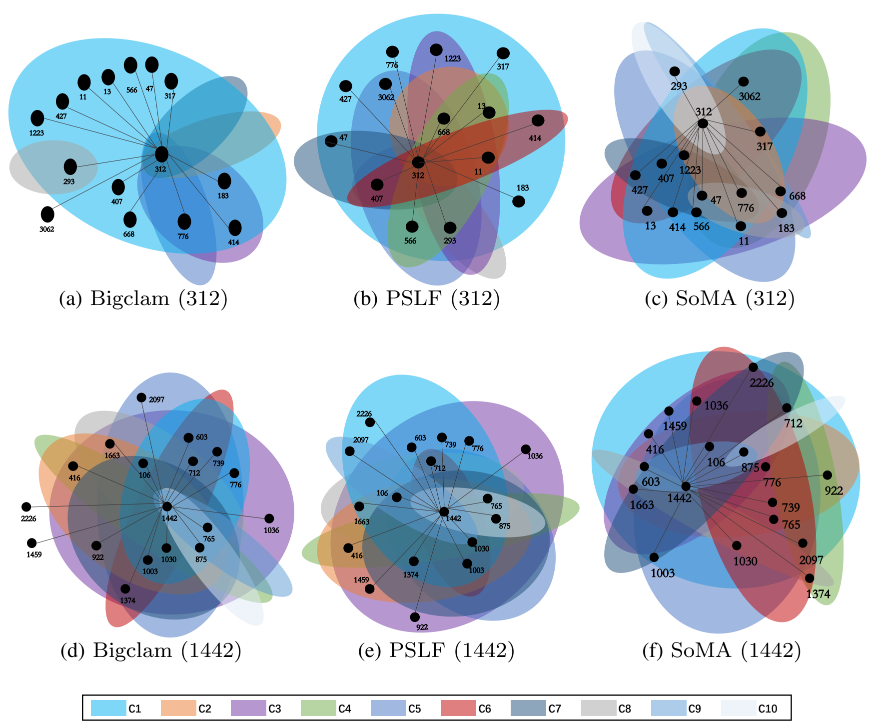


Figure C5 Social communities obtained via Bigclam, PSLF and **SoMA** for two kinds of representative users on *Ciao*.

they adopt the same community detection method Bigclam [23]. We determine whether user i belongs to community c by comparing $\boldsymbol{\theta}_{ic}$ with a predefined threshold $\epsilon = \sum_{i,c} \boldsymbol{\theta}_{ic}/nr$ for all methods). Fig. C5 shows the community of two kinds of representative users from *Ciao* dataset learned via Bigclam, PSLF and **SoMA**. In order to show the difference between them, we investigated the user communities properties from three different views.

View 1: The users are assigned into more communities by the proposed **SoMA** than by Bigclam and PSLF. Taking the user 312 as an example, it belongs to 10 communities as shown in Fig. C5(b) and Fig. C5(c), but belongs to 5 communities as shown in Fig. C5(a). These three subfigures contain the same subset of users. It can be seen that most users are assigned into the community C1 by Bigclam, while they are divided into different communities by PSLF and **SoMA**. By investigating their rating behavior, there is almost no relevance between user 47, 317, 427 and 566. Thus, it is better to separate them (as done by **SoMA**) rather than put them into a big community (as done by Bigclam). In this case, the latent factor of user 312 will be affected by 5 gross communities in MFC and SoDimRec, while by 10 delicate communities in PSLF and **SoMA**, which may be one of the reasons to improve the final recommendation accuracy. Compared with PSLF, **SoMA** has ability to determine the fine-grained preference influence. Table C3 lists the recommendation accuracy on three representative users to demonstrate that these users benefit from the proposed **SoMA**.

Table C3 Recommendation accuracy (RMSE) on representative users in Fig. C5.

View	*View 1*			*View 2*			*View 3*		
userID	427	566	312	1003	922	1442	1459	2226	3062
MFC	0.9761	0.9677	0.9832	0.9871	0.9677	0.9782	0.9973	1.0324	1.0244
SoDimRec	0.9851	0.9719	0.9713	0.9893	0.9788	0.9645	0.9855	1.0315	1.0341
PSLF	0.9731	0.9688	0.9705	0.9733	0.9714	0.9652	0.9831	1.0233	1.0261
SoMA	**0.9724**	**0.9653**	**0.9696**	**0.9621**	**0.9648**	**0.9589**	**0.9788**	**0.9983**	**0.9996**

View 2: The users are assigned into fewer communities by the proposed **SoMA** than by Bgiclam and PSLF. Taking the user 1442 as an example, as shown in Fig. C5(d), (e) and (f), it belongs to 5 communities in **SoMA**, but belongs to 10 communities in Bigclam and 8 communities in PSLF. User 1442 has nothing to do with the communities C1, C6, C7, C8 and C9 in **SoMA**. Similarly, by investigating the rating behavior, we find that the rating preference similarities between user 1442 and the users in these four communities are almost zero, thus it is necessary to split user 1442 from such communities. In other words, the proposed method has ability to eliminate the unnecessary social relations, which may be one of the reasons to improve the final recommendation accuracy. Table C3 lists the recommendation accuracy on three representative users, which further confirms that the proposed social recommendation framework is useful to improve the recommendation performance.

View 3: The users cannot be assigned into any community by Bigclam, like user 3062 in Fig. C5(a), and user 1459 and user 2226 in Fig. C5(d). Note that these users can be grouped into the corresponding communities by PSLF and **SoMA**, which indicate that user preference can be influenced. For example, user 3062 is assigned into C1 and C5 in PSLF and assigned into C1, C5 and C7 in **SoMA**. Once the user belonging to the proper communities, his or her latent factor can be identified by the supervision of the indirect social relations. Table C3 lists the recommendation accuracy on three representative users in *View 3*, where **SoMA** is better than PSLF, MFC and SoDimRec. Although both PSLF and **SoMA** are unified graphical model, **SoMA** can determine better user community membership.

Appendix C.2.4 *Effect of global social structure*

Popularity is important for users on social networks especially for professional users who are following more serious goals of increasing visibility, turnover, sells, and so on. Reaching those goals through social media can be directly affected by the number of followers. In the proposed **SoMA** model, we introduce a global social structure variable γ_i to model the influence of user popularity. Thus, we conduct experiments to investigate the effect of user popularity on user preference learning. We remove the user popularity variable γ from the **SoMA** and it degenerates into **SoMA-G**, i.e., user preference is modeled via $\mathbf{S}_{ik} \sim B(\mathbf{p}_{ik})$. Table C4 lists the recommendation performance obtained via **SoMA** and **SoMA-G** on four datasets in terms of RMSE. Obviously, **SoMA** performs better than **SoMA-G**, which indicates considering global social structure among all users is useful for recommendation.

Table C4 Comparison of **SoMA-G** and **SoMA**.

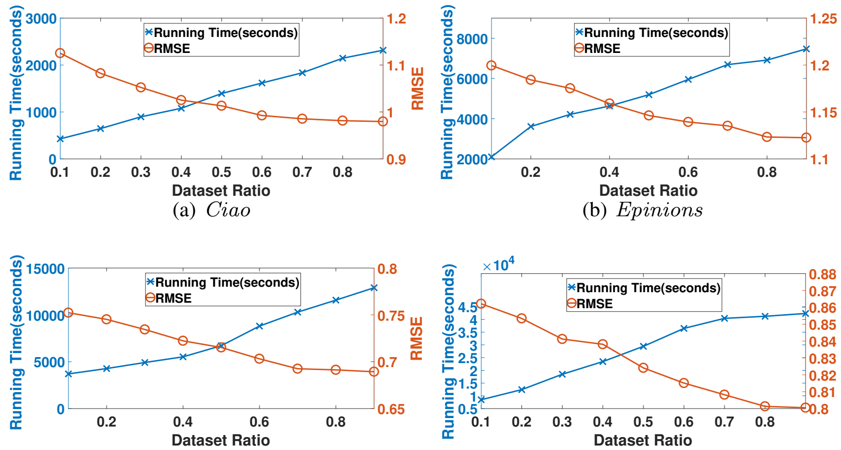
Datasets	**SoMA-G**	**SoMA**
Ciao	0.9910	**0.9813**
Epinions	1.1126	**1.0943**
Douban	0.6967	**0.6857**
Yelp	0.8127	**0.7957**

Table C5 Running time (seconds) of recommendation methods with implicit social structures.

Datasets	*Ciao*	*Epinions*	*Douban*	*Yelp*
MFC	1875.38	6322.17	11561.79	38161.32
SoDimRec	2988.52	11204.54	17228.14	60958.71
PSLF	3782.31	18852.32	31231.41	62331.48
SoMA	2315.12	7455.21	13221.53	42331.51

Appendix C.2.5 *Generalization ability analysis*

In order to evaluate the generalization ability of the proposed model, we choose different training sets with different sizes {20%, 40%, 60%, 80%} and fix the same 20% as the testing set. We guarantee that small size training set is included in large size training set. Since TrustSVD and UniWalk obtains competitive performance among all baselines and NSCR and DeepSoR belongs to deep social recommendation, the following comparisons and analysis focus on **SoMA**,



(a) *Ciao* (b) *Epinions* (c) *Douban* (d) *Yelp*

Figure C6 Scalability analysis of **SoMA**.

TrustSVD, UniWalk, NSCR and DeepSoR. Figure C4 shows the RMSE comparison in terms of different training set on four datasets. Since we have similar observations in terms of other metrics, we only show the results in RMSE. Obviously, the recommendation performance becomes better and better with the increasing of training data and **SoMA** is superior to TrustSVD and UniWalk for all training set on four datasets. Note that *Epinions* and *Yelp* is extremely sparse, Figure C4 (b) and (d) show that **SoMA** significantly outperforms TrustSVD, UniWalk, NSCR and DeepSoR, especially for small size training sets (20%, 40%). This result further demonstrates that **SoMA** has a strong generalization ability compared with existing social recommendation methods.

Appendix C.2.6 *Complexity and scalability analysis*

In this part, we evaluate the efficiency of the designed parallel graph computing algorithm used in **SoMA**. The algorithm is implemented in C++ at the hosts with Intel(R) Xeon(R) 2.0GHz CPU E7-4820 v2 having four processors, each processor has eight cores and the memory is 64GB. The operating system is Ubuntu 16.04.2 LTS. Meanwhile, the Intel(R) parallel studio XE 2016 composer edition for cpp is used to compile all codes so that the inter-procedural optimizations can be performed to improve the parallel computing.

Table C5 lists the average running time (for 5-fold cross validation) of different social recommendation methods. For the existing two-stage social recommendation methods (MFC and SoDimRec), the running time contains two parts, one for running Bigclam to find the implicit social structures, and the other for learning the latent user/item factors. As expected, **SoMA** with scalable variational inference is much faster than them. Meanwhile, although PSLF learns latent user factors and implicit social structures in a unied model, the inference algorithm, expectation maximization, has to be used to approximate the complex probabilistic generative process. Fortunately, the proposed model **SoMA** can be directly approximated with the aid of parallel gradient-based technique, which are more efficient and effective than PSLF.

Additionally, we investigate the scalability of the **SoMA** model. We randomly select a subset of ratings as training set according to a fixed ratio (from 0.1 to 0.9 with step 0.1) and x the testing set. For each ratio, ten subsets are extracted as training data and the averaged results (running time and RMSE) are recorded in Figure C6. Obviously, the recommendation performance becomes better and better with the increasing of training data size. Meanwhile, the training computational complexity (i.e., running time) is linearly scalable to the training data size.

Appendix C.2.7 *Interpretable social groups*

SoMA exploits both rating and social information to determine the latent space. By assigning users to different social groups, each group contains users with similar taste, which is helpful to describe the corresponding latent feature. Once obtaining the user-group membership vector for each user ($\boldsymbol{\theta}_i = \{\boldsymbol{\theta}_{ic}\}_{c=1}^{d}$ and $0 \leqslant \boldsymbol{\theta}_{ic} \leqslant 1$), we can get the user-group assignment matrix $\boldsymbol{\theta}$, and know the set of like-minded users. In this case, each social group can be explicitly described by these like-minded users, which will be helpful to understand the corresponding latent factor.

Specifically, we set a threshold ϵ_θ for affiliation strengths with their averaged value, which is defined by $\epsilon_\theta = \frac{1}{nd}\sum_{i=1}^{n}\sum_{c=1}^{d}\boldsymbol{\theta}_{ic}$. Based on this threshold, for each group, we can select its representative users by $SU_c = \{\mathrm{USER}_i | \boldsymbol{\theta}_{ic} > \epsilon_\theta\}$. It can be seen that the users in SU_c have strong relations with the c-th group. In other words, we can explain each latent feature (i.e., social group) via the information about items that the representative users (SU_c) rated. To confirm this, we investigate the semantical information of each user group for *Yelp* dataset, where each item (i.e., business) is marked by one or more categorical labels (there are total 1240 categories). According to the frequency that each category appears in all items, we selected top 20 categories including *Restaurants*, *Shopping*, *Food*, *Beauty&Spas*, *Home Services*, *Health&Medical*, *Nightlife*, *Bars*, *Automotive*, *Local Services*, *EP&Services*, *Active Life*, *Fashion*, *Sandwiches*, *Fast Food*, *American*, *Pizza*, *Coffee&Tea*, *Hair Salons*, and *Hotels&Travel*.

Table C6 Semantical information for 5 selected user groups

Strong Group ID	1	2	3	4	5
Semantical Labels	*Restaurants* *Pizza, Food*	*Fashion* *Beauty&Spas*	*Hotels&Travel*	*Health&Medical* *Active Life*	*Local Service* *Home Service*

For each social group, we collect the representative items which are rated by the representative users (i.e., the users in SU_c for the c-th group) and have rating values greater than 4. By counting the frequency of each category appearing in the representative items of each social group, we can select the representative categories and take them as the semantic information of the corresponding social group. Table C6 lists the representative categories of five user groups. By taking advantage of such semantical information, **SoMA** can provide explainable recommendation reason for each predicted rating. Taking the 89-th business (*Pizza Company* with label {*Restaurants*, *Pizza*, *Food*}) and the 105-th user (named by Jenifer) in *Yelp* as an example, the predicated rating value that Jenifer gives to item *Pizza Company* is 4.5, and the recommendation reason can be given as follows.

To Jenifer:

We recommend business *Pizza Company* to you because you are strongly associated with the social group {*Restaurants*, *Pizza*, *Food*} and *Pizza Company* is a business simultaneously related to categories *Restaurants*, *Pizza* and *Food*.

References

1 James O Berger. *Statistical decision theory and bayesian analysis.* Springer Science & Business Media, 2013.

2 Lucien LeCam et al. Convergence of estimates under dimensionality restrictions. *The Annals of Statistics*, 1(1):38–53, 1973.

3 Bin Yu. *Assouad, Fano, and Le Cam*, pages 423–435. Springer New York, New York, NY, 1997.

4 Lucien Le Cam. *Asymptotic methods in statistical decision theory.* Springer Science & Business Media, 2012.

5 Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *JMLR*, 14(1):1303–1347, 2013.

6 Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proc. of ICML*, pages 880–887. ACM, 2008.

7 Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proc. of WWW*, pages 173–182, 2017.

8 J. Tang, X. Hu, H. Gao, and H. Liu. Exploiting local and global social context for recommendation. In *Proc. of IJCAI*, pages 264–269, Sep. 2013.

9 G. Guo, J. Zhang, and N. Yorke-Smith. Trustsvd: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In *Proc. of AAAI*, pages 123–129, 2015.

10 Xixi Du, Huafeng Liu, and Liping Jing. Additive co-clustering with social influence for recommendation. In *Proc. of ACM RecSys*, pages 193–200. ACM, 2017.

11 Y. Shen and R. Jin. Learning personal+social latent factor model for social recommendation. In *Proc. of ACM SIGKDD*, pages 1303–1311, 2012.

12 H. Li, D. Wu, W. Tang, and N. Mamoulis. Overlapping community regularization for rating prediction in social recommender systems. In *Proc. of ACM RecSys*, pages 27–34. ACM, 2015.

13 J. Tang, S. Wang, X. Hu, D. Yin, Y. Bi, Y. Chang, and H. Liu. Recommendation with social dimensions. In *Proc. of AAAI*, pages 251–257, 2016.

14 Haekyu Park, Hyunsik Jeon, Junghwan Kim, Beunguk Ahn, and U Kang. Uniwalk: explainable and accurate recommendation for rating and network data. In *Proc. of SIAM SDM*, 2018.

15 Xiang Wang, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. Item silk road: Recommending items from information domains to social users. In *Proc. of ACM SIGIR*, pages 185–194. ACM, 2017.

16 Wenqi Fan, Qing Li, and Min Cheng. Deep modeling of social relations for recommendation. In *Proc. of AAAI*, 2018.

17 Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *Proc. of WWW*, pages 417–426. ACM, 2019.

18 H. Ma, H. Yang, M. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *Proc. of ACM CIKM*, pages 931–940. ACM, 2008.

19 B. Yang, Y. Lei, D. Liu, and J. Liu. Social collaborative filtering by trust. In *Proc. of IJCAI*, pages 2747–2753, 2013.

20 M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proc. of ACM RecSys*, pages 135–142. ACM, 2010.

21 H. Ma, D. Zhou, C. Liu, M. Lyu, and I. King. Recommender systems with social regularization. In *Proc. of ACM WSDM*, pages 287–296. ACM, 2011.

22 Y. Koren, R. Bell, and C. Kolinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

23 J. Yang and J. Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Proc. of WSDM*, pages 587–596, 2013.