TopRec: Domain-Specific Recommendation through Community Topic Mining in Social Network

Xi Zhang, Jian Cheng*, Ting Yuan, Biao Niu, Hanqing Lu
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
Beijing, China
{xi.zhang, jcheng, tyuan, bniu, luhq}@nlpr.ia.ac.cn

ABSTRACT

Traditionally, Collaborative Filtering assumes that similar users have similar responses to similar items. However, human activities exhibit heterogenous features across multiple domains such that users own similar tastes in one domain may behave quite differently in other domains. Moreover, highly sparse data presents crucial challenge in preference prediction. Intuitively, if users' interested domains are captured first, the recommender system is more likely to provide the enjoyed items while filter out those uninterested ones. Therefore, it is necessary to learn preference profiles from the correlated domains instead of the entire user-item matrix. In this paper, we propose a unified framework, TopRec, which detects topical communities to construct interpretable domains for domain-specific collaborative filtering. In order to mine communities as well as the corresponding topics, a semi-supervised probabilistic topic model is utilized by integrating user guidance with social network. Experimental results on real-world data from Epinions and Ciao demonstrate the effectiveness of the proposed framework.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information filtering; H.3.5 [**Online Information Services**]: Web-based services

General Terms

Algorithms, Experimentation

Keywords

Recommender Systems, Collaborative Filtering, Social Network, Probabilistic Topic Modeling

1. INTRODUCTION

As electronic commerce becomes increasingly popular, large amounts of available information for products are flooding the Web. To avoid customers inundated with choices, nowadays recommender systems take a central role by selecting the potential enjoyed products and filtering out uninterested ones. Through taking personalized recommendations for products, some famous online shopping websites such as Amazon and Netflix have expanded their marketing successfully [7, 15].

Most of these commercial systems are based on Collaborative Filtering(CF), which is an effective recommendation approach with fundamental assumption that *two users have similar tastes on one item if they have rated other items similarly* [29, 24]. Due to the collaboration effects, CF only relies on users' history behaviors without collecting content information for all users and items. Another benefit of CF is that unexpected products could be recommended to users by mining user-product interactions. Those products are difficult to be discovered by merely analyzing the contents.

Although CF approaches have superior characteristics and have been applied to many real-world systems, there still exist drawbacks which limit their performance. On the one hand, traditional CF considers collaboration effects among users but ignores the variety across different domains. Generally, customers have similar tastes in one domain could not infer that they have similar tastes in other domains. An impressive example is involving Epinions[1], which is one of the largest product review websites with several less-correlated domains such as "Movies", "Music", "Home & Garden", and so on. On the site, two users who love movies of the same type are probably to have totally different preference in "Home & Garden" domain. In this sense, users show heterogenous features across multiple domains. As a consequence, training *domain-specific recommendation* model is more reasonable than training a model on the entire user-item matrix.

On the other hand, recommender systems in practice have to face a main challenge of data sparsity. Typically, there are usually thousands of products on e-commerce websites but most of them are long-tailed. In this case, recommendation based on CF methods is inclined to suggest well-known products rather than those cold ones [19]. Hence, it is quite difficult to predict user preference accurately from the whole item set. To alleviate this problem, an intuitive scheme is to take users' interested topic into account. Considering a scenario that if a user's attitude toward domains is captured first, one who expresses great interest in "Music" would probably receive a recommendation list with more music-related products than popular products in other domains. Compared to prediction in the whole item set, a user's preferred long-tailed items are more likely to be dug out via prediction in the interested domains.

*corresponding author

[1] http://www.epinions.com

WWW 2013, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2035-1/13/05.

Nevertheless, the information of users' interested topics is absent in most existing consumer review sites. One straightforward solution is to count how many times a user purchased or clicked products in one domain, then whether the user is interested in a domain or not would be inferred by her statistical history reviews. However, users without interest interactions but with social relations in the domain might probably be ignored. Also, since the distribution of review number over users is quite imbalanced, it is unclear how to define a unified criterion to partition users into different domains. Instead, we change our perspective to mining interpretable topics for communities. The process is equivalent to finding user clusters meanwhile align them into the predefined domains, and therefore can be viewed as *community topic mining*.

With all the concerns aforementioned, we develop a novel recommendation framework integrating community topic mining with domain-specific recommendation, which is called *TopRec* for short. As we known, social community detection itself is a very important research task with great challenges in data mining [4, 30]. The crucial difference is, in our TopRec framework, we not only focus on community detection algorithm, but also investigate:(1) *how to align these extracted user communities without explicit topic to the known domains*, and (2) *how to take advantage of the natural social network structure to assist user clustering.* To the best of our knowledge, these issues have not been studied in recommender systems before.

To address above two issues, we propose a unified probabilistic topic model in the context of social network in this paper, by combining semi-supervised methods of user-guided and structure-constrained clustering. Then the accuracy of the top-n recommendation task can be improved by domain-specific CF algorithm. The main contributions of this work are summarized below:

- Proposing a novel recommendation framework, TopRec, for multi-category datasets with trust connections to deal with the limitations of conventional CF methods.
- Introducing user guidance as a sort of prior knowledge in the probabilistic topic model to detect communities and align them with existing domains at the same time.
- Embedding social links as complementary data resource in the probabilistic topic model to satisfy connectivity coherency.
- Employing domain-specific CF approach to formulate heterogenous latent features corresponding to the interested domains for user.

The rest of the paper is organized as follows. A new recommendation framework, TopRec, is introduced in Section 2. A unified probabilistic model for community topic mining is described in Section 3 and a domain-specific CF model for recommendation is presented in Section 4. Then we analyze experimental results on benchmark datasets in Section 5. Finally, we discuss the related work in Section 6 and conclude this work in Section 7.

2. A NEW FRAMEWORK - TOPREC

In this section, we introduce an overview of our recommendation framework, TopRec, and give the problem definition formally.

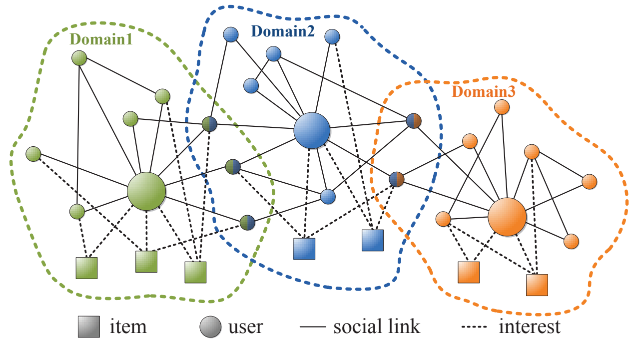


Figure 1: A simple illustration of interest interactions and social network within domains. The larger nodes represent experts in domains.

2.1 Framework Overview

There are three types of data in our framework: the review records and the social network for all users, as well as the category information for all items. Without loss of generality, Figure 1 depicts a typical topology of a heterogeneous graph containing both interest interactions and social network. The essential of our goal is to construct domains by the given data and community topic mining, then conduct domain-specific recommendation, which consists of following two stages:

Stage 1: Community Topic Mining. In this study, we tackle this step with probabilistic topic model for three main considerations. First, individuals in real life are usually multi-faceted. As their interests cannot be captured by one topic, discrete mixed membership models like Probabilistic Latent Semantic Analysis(PLSA) [10] and Latent Dirichlet Allocation(LDA) [1], which can represent objects as distribution over topics, are suitable in our scenario. Second, in consumer review sites such as Epinions, we can always find active users in each domain(the larger node in Figure 1), who can supply prior knowledge of the probabilistic model for topic alignment. Third, probabilistic topic model allows to connect similar users in social relations to enhance community meanwhile indicates why we connect them.

Stage 2: Domain-Specific Collaborative Filtering. After extracting topical communities, users are divided into domains according to their interested topics. To predict user preference, we exploit normal CF in each domain and then return a rank list according to the predicted rating scores. In our case, the particular algorithm utilized is Probabilistic Matrix Factorization(PMF) [21], a state-of-the-art CF approach with promising recommendation results. As a matter of fact, the TopRec is a general framework and has potential to combine other CF methods in practice.

One similar problem has been formulated in Multiclass Co-Clustering(MCoC) model recently [27], which groups users and items into topical subsets to achieve better recommendation results. However, TopRec boosts performance by exploring interpretable domains while the topics in MCoC are implicit.

2.2 Problem Definition

Here we present the notations and definitions to be used in this paper. Suppose there are N users and M items in the given dataset, $\mathcal{U} = \{u_1, \cdots, u_N\}$ denote the user set and $\mathcal{V} = \{v_1, \cdots, v_M\}$ denote the item set. Let $\mathbf{R} = (\mathbf{r}_1, \cdots, \mathbf{r}_N)^T \in \mathbb{R}^{N\times M}$ represent rating matrix, where $\mathbf{r}_i$ is column vector including ratings from user u_i towards item set $\mathcal{V}$. Note that rating vector $\mathbf{r}_i$ can be viewed as feature representation for user u_i. Then a user u_i in collection $\mathcal{U}$ can be denoted by a series of items $\{v_j\}_{j=1}^M$. Let $\mathbf{T} = (\mathbf{t}_1, \cdots, \mathbf{t}_N)^T \in \mathbb{R}^{N\times N}$ denote a binary trust matrix, where $\mathbf{t}_i$ is column vector indicating whether user u_i trust others in user set $\mathcal{U}$. We also model this user-user relationship as a graph $\mathcal{G}$ with adjacency matrix $\mathbf{W}$ to encode this trust relations into community mining model.

Definition 1. **Domain**. Suppose the number of domains is K, a domain $\mathcal{D}_k$ is composed of item subset $\mathcal{V}_k$ and user subset $\mathcal{U}_k$, where $k = 1, \cdots, K$, then we have N_k users and M_k items in each domain. Thus, a set of domain can be represented as $\mathcal{D} = \{\mathcal{D}_1, \cdots, \mathcal{D}_K\}$.

Definition 2. **Problem Definition**. Given review matrix $\mathbf{R}$, social context $\mathbf{T}$ and item-domain information $\mathcal{V}_k = \{v_1, \cdots, v_{M_k}\}$, our goal is: In stage 1, discover domains $\mathcal{D}_k = \{\mathcal{U}_k, \mathcal{V}_k\}$ by mining $\mathcal{U}_k = \{u_1, \cdots, u_{N_k}\}$, where $k = 1, \cdots, K$. In stage 2, for each domain $\mathcal{D}_k$, we train PMF model using the observed ratings of user-item pairs (u_i, v_j), where $u_i \in \mathcal{U}_k$ and $v_j \in \mathcal{V}_k$. Finally, by utilizing the learnt domain-specific latent features in prediction, the personalized rank lists for users in set $\mathcal{U}$ towards the whole item set $\mathcal{V}$ are returned.

3. COMMUNITY TOPIC MINING

In this section, we propose a semi-supervised probabilistic topic model with expert guidance and network structure. By modeling the generative process of user set, the model could explicitly mine their interest topics from both historical ratings and social relationships.

3.1 Basic Idea

Probabilistic generative models such as PLSA have been widely used in many text mining tasks [10, 3, 16], and also play a significant role in recommendation tasks due to their ability in dealing with dyadic data [11, 26]. Following the previous works on probabilistic models, we treat user as a mixture of topics, where each topic is a multinomial distribution over all the items. For instance, a distribution that assigns high probabilities to items such as "iPhone","iPad","Kindle" would suggest that the user loves "Electronics" topic. In order to identify multiple interest topics for user, we model users in a mixture model with K topics and estimate the model parameters so that the likelihood of user collection $\mathcal{U}$ is maximized. In this way, the statistical topic model could capture the co-occurences of items and encourage to group users into communities.

Intuitively, user communities grouped by basic PLSA model can represent interest topics towards item categories. However, these extracted topics are latent variables without explicit meaning and cannot be regarded as the given categories. Thus, simply using PLSA cannot ensure the obtained topic is well-aligned to the specific domains. To overcome this limitation, we introduce user guidance as a priori into the generative clustering process. In reality, most users in consumer review sites are extremely cold. That is, there is little useful information could guide community mining. Fortunately, there exit a few representative and trustable users in each domain, who review and comment frequently meanwhile own a large number of trustors. In this study, we regard these users as *experts* of each community and their rating distribution over items as prior knowledge for clustering.

Another problem is that the social network structure might be neglected. Generally, people not only make interactions with online products but also have trusts relation with other users. Thus only using rating data cannot guarantee that well-connected users are clustered in the same topical community. Besides, more information about cold users can be analyzed from the perspective of their local trusting neighbors. For example, one might do not have enough reviews, while has more trust relations such that mining her interest topic from social view is possible. Accordingly, these linked structures are quite useful in community topic mining.

3.2 Experts-Guided Topic Modeling

To model the user clustering procedure with experts guidance, we introduce a topic variable $z_k \in \{z_1, \cdots, z_K\}$ with each observation of a item $v_j \in \{v_1, \cdots, v_M\}$ is rated by a particular user $u_i \in \{u_1, \cdots, u_N\}$. Each topic z_k is corresponding to the k-th domain. Let $c(u_i, v_j) = R_{ij}$ denote the rating of user u_i giving item v_j to express how much the user like the item.

Given a user collection $\mathcal{U}$, experts-guided topic modeling is to discover topical communities meanwhile label them the known domains, and then users and items with a similar topic could be mapped into the same subgroup. To this end, we take the rating vectors of experts as prior knowledge for each user cluster. To make sure the effect of the guidance, the experts are manually chosen by two criteria:

- **Informative**. The experts should be the persons who contribute a large quantity of reviews in one particular item category, so that their attitude can cover items in the category as more as possible.

- **Reliable**. The experts would better have a great number of followers, which implies their reviews and comments are trustable to some certain extent.

Once the experts are selected, we model such prior as a Dirichlet distribution to enforce the topics to be as close as possible to the predefined domains. Specifically, for each given topic z_k, its probabilistic distribution over items $p(v_j|z_k)$ is assumed to be a multinomial distribution, which is generated from some Dirichlet distribution. We define this Dirichlet prior as $z_k : Dir(\{\sigma_k p(v_j|\bar{u}_k) + 1\}_{v_j \in \mathcal{V}})$, where σ_k is confidence parameter of the prior distribution for topic k, and $\bar{u}_k \in \mathcal{U}$ denote expert corresponding to the topic k.

Similar to PLSA, parameters in our model are $\{p(v_j|z_k), p(z_k|u_i)\}$. We set all the parameters as Θ for succinct in the following paragraph. In general, the prior on the parameters can be presented as

$$P(\Theta) \propto \prod_{k=1}^{K}\prod_{j=1}^{M} p(v_j|z_k)^{\sigma_k p(v_j|\bar{u}_k)} \tag{1}$$

where the prior $p(v_j|\bar{u}_k)$ involves the rating distribution of experts over the item set $\mathcal{V}$, and can be obtained by

$$p(v_j|\bar{u}_k) = \frac{c(\bar{u}_k, v_j)}{\sum_{j'=1}^{M} c(\bar{u}_k, v_{j'})} \tag{2}$$

A large σ_k equals to high confidence on prior of topic k. When $\sigma_k = 0$, the prior of $p(v_j|z_k)$ boils down to a uniform distribution, which means that no guidance is introduced in the clustering process.

In fundamental PLSA model, the log-likelihood of user collection $\mathcal{U}$ is

$$\mathcal{L}(\mathcal{U}) = \log p(\mathcal{U}|\Theta) = \sum_{i=1}^{N}\sum_{j=1}^{M} c(u_i, v_j) \log \sum_{k=1}^{K} p(v_j|z_k)p(z_k|u_i) \tag{3}$$

We may use Bayesian estimation, so the parameter Θ can be estimated by maximizing Eq.(3), which is

$$\hat{\Theta} = \arg\max_{\Theta} \log p(\mathcal{U}|\Theta) \tag{4}$$

With the prior incorporated, Maximum A Posterior (MAP) estimator is used instead of Maximum Likelihood estimator. That is, $\mathcal{L}(\mathcal{U}) = \log(p(\mathcal{U}|\Theta)p(\Theta))$. Therefore, Θ is obtained by

$$\begin{aligned}\hat{\Theta} &= \arg\max_{\Theta} \log(p(\mathcal{U}|\Theta)p(\Theta)) \\ &= \arg\max_{\Theta}\{\sum_{i=1}^{N}\sum_{j=1}^{M} c(u_i, v_j) \log \sum_{k=1}^{K} p(v_j|z_k)p(z_k|u_i) \\ &+ \sum_{k=1}^{K}\sum_{j=1}^{M} \sigma_k p(v_j|\bar{u}_k) \log p(v_j|z_k)\}\end{aligned} \tag{5}$$

3.3 Network-Constrained Topic Modeling

Furthermore, we propose to regularize the user-specific feature space by social network during clustering process. Given a binary trust matrix $\mathbf{T} = (\mathbf{t}_1, \cdots, \mathbf{t}_N)^T \in \mathbb{R}^{N\times N}$, a social graph $\mathcal{G} = (\mathcal{U}, \mathcal{E}, \mathbf{T})$ could be constructed, where $\mathcal{U}$ is a set of N vertices representing users, $\mathcal{E} \subseteq \mathcal{U}\times\mathcal{U}$ is a set of edges representing trust similarity between users in neighborhood, and $\mathbf{T}$ can be regarded as a matrix in which each row corresponds to a vector of feature values of a user. In particular, user relationships are embodied by defining adjacency matrix $\mathbf{W}$ on user graph as

$$W(u_i, u_{i'}) = \begin{cases} sim(u_i, u_{i'}), & \text{if } u_{i'} \in \mathcal{N}(u_i) \text{ or } u_i \in \mathcal{N}(u_{i'}) \\ 0, & \text{otherwise.}\end{cases} \tag{6}$$

where $\mathcal{N}(u_i)$ is the k-nearest neighbor of u_i, and $sim(u_i, u_{i'})$ represents the trust similarity measurement. For simplicity, we use cosine distance between vectors $\{\mathbf{t}_i\}_{i=1}^N$ here. Though above equation, we transform the original directed trust graph $\mathbf{T}$ to an undirected trust similarity graph $\mathbf{W}$, which is able to formulate into topic model as regularization term directly.

The essential idea of network-constrained user clustering is on the basis of one simple assumption: *users who have a strong connection with each other in social network should have similar preference on topics*. Inspired by this fact, we adopt the following formulation as a constraint for topic model

$$R(\mathcal{U}, \mathcal{G}) = \frac{1}{2}\sum_{\langle u_i, u_{i'}\rangle\in\mathcal{E}} W(u_i, u_{i'}) \sum_{k=1}^{K}(p(z_k|u_i) - p(z_k|u_{i'}))^2 \tag{7}$$

Therefore, our aim is to minimize Eq.(7). Then we utilize this constraint into log-likelihood function generated by PLSA, which means to maximize

$$\mathcal{J}(\mathcal{U}, \mathcal{G}) = \mathcal{L}(\mathcal{U}) - \lambda R(\mathcal{U}, \mathcal{G}) \tag{8}$$

where λ is a regularization parameter which controls the influence of smoothness on topic distribution over network.

3.4 The Unified Model

Combining above two parts of topic modeling, we have a joint objective function with concerning both topical consistency and connectivity coherency at the same time. By substituting the posterior probability formula in Eq.(5) and the regularizer in Eq.(7) into Eq.(8), the final objective function can be written as

$$\begin{aligned}\mathcal{J}(\mathcal{U}, \mathcal{G}) = &\sum_{i=1}^{N}\sum_{j=1}^{M} c(u_i, v_j) \log \sum_{k=1}^{K} p(v_j|z_k)p(z_k|u_i) \\ &+ \sum_{k=1}^{K}\sum_{j=1}^{M} \sigma_k p(v_j|\bar{u}_k) \log p(v_j|z_k) \\ &- \frac{\lambda}{2}\sum_{\langle u_i, u_{i'}\rangle\in\mathcal{E}} W(u_i, u_{i'}) \sum_{k=1}^{K}(p(z_k|u_i) - p(z_k|u_{i'}))^2\end{aligned} \tag{9}$$

3.5 Parameter Estimation

For Maximum Likelihood Estimation (MLE) procedure, the Expectation Maximization (EM) algorithms is commonly used. However, in our MAP case with the combination of regularizer, parameter estimation becomes difficult to handle by the standard EM. Hence, we apply the Generalized Expectation Maximization algorithm (GEM) to estimate the parameter Θ. Parameter in n-th iteration are denoted as Θ_n.

Formally, we conduct a two-step iterative algorithm. In E-step, given all the users' reviews data and parameter Θ_n, the distribution of the topics can be computed simply by the same formula as PLSA

$$p(z_k|u_i, v_j, \Theta_n) = \frac{p_n(v_j|z_k)p_n(z_k|u_i)}{\sum_{k'=1}^{K} p_n(v_j|z_{k'})p_n(z_{k'}|u_i)} \tag{10}$$

In M-step, the algorithm searches better parameters through optimizing Q-function: $\Theta_{n+1} = \arg\max_{\Theta} Q(\Theta; \Theta_n)$, which is present by

$$\begin{aligned}Q(\Theta; \Theta_n) = &\mathcal{J}(\Theta_n) + \sum_{i=1}^{N}\alpha_i(\sum_{k=1}^{K} p(z_k|u_i) - 1) \\ &+ \sum_{k=1}^{K}\alpha_k(\sum_{j=1}^{M} p(v_j|z_k) - 1)\end{aligned} \tag{11}$$

where α_i and α_k are Lagrange multipliers of the constraints $\sum_k p(z_k|u_i) = 1$ for all users and $\sum_j p(v_j|z_k) = 1$ for all topics, respectively.

Computation of $\mathbf{p_n(v_j|z_k)}$ To optimize Eq.(11) with respect to $p_n(v_j|z_k)$, we need to consider the original PLSA likelihood function and the user guidance term. By taking partial derivative of $p_n(v_j|z_k)$ to Eq.(11), its updating can be got as

$$p_{n+1}(v_j|z_k) = \frac{\sum_{i=1}^{N} c(u_i, v_j)p(z_k|u_i, v_j, \Theta_n) + \sigma_k p(v_j|\bar{u}_k)}{\sum_{j'=1}^{M}\sum_{i'=1}^{N} c(u_{i'}, v_{j'})p(z_k|u_{i'}, v_{j'}, \Theta_n) + \sigma_k} \tag{12}$$

This formulation can be understood easily. In addition to the data of user collection, the opinions of experts on item distribution over topics are also imposed. Then parameter updating are decided by collaboration of the both factors, which is also consistent with our intuition.

Computation of $p_n(z_k|u_i)$ Optimizing Eq.(11) with respect to $p_n(z_k|u_i)$ directly is more complicated even though it is only related with the terms of data likelihood and network regularization. Similar to [3], we take the strategy in GEM to satisfy $Q(\Theta_{n+1}) \geq Q(\Theta_n)$ in each step, which finds a better Θ rather than finds a globally optimal solution. We apply the normal updating method in standard PLSA to maximize $\mathcal{L}(\mathcal{U})$ in Eq.(8) which can find a start value Θ^1_{n+1}. Then $R(\mathcal{U}, \mathcal{G})$ is increased by

$$p^{t+1}_{n+1}(z_k|u_i) = \tau p^t_{n+1}(z_k|u_i) + (1-\tau)\frac{\sum_{i'=1}^{N} W(u_i, u_{i'})p^t_{n+1}(z_k|u_{i'})}{\sum_{i'=1}^{N} W(u_i, u_{i'})} \tag{13}$$

where τ is Newton step parameter to limit the effect of smoothness by $R(\mathcal{U}, \mathcal{G})$. We repeat this iteration until the Q-function is beginning to drop. Also, we judge the output Θ^t_{n+1}. If $Q(\Theta^t_{n+1}) \geq Q(\Theta_n)$, we adopt the proposal of Θ^t_{n+1}, otherwise, reject it. The E-step and M-step equations are alternated until achieving some termination conditions. In this way, we obtain the estimated parameters $\{p(v_j|z_k), p(z_k|u_i)\}$.

Now, to group each user into more than one topical communities, we set a natural clustering criterion as: *a user u_i is interested in a meaningful topic z_k, if and only if function $f(z_k, u_i)$ is satisfied by $f(z_k, u_i) > \varepsilon$.* Thus, given a user, the function $f(\cdot)$ measures attractiveness of a domain on her, which is defined as

$$f(z_k, u_i) = \sum_{j=1}^{M} c(u_i, v_j)p(v_j|z_k) \tag{14}$$

Clearly, Eq.(14) could be viewed as a kind of similarity measurement between users and topics.

4. DOMAIN-SPECIFIC COLLABORATIVE FILTERING

In this section, we illustrate domain-specific collaborative filtering in detail. The overall procedure consists of two steps, which are model training and top-n recommendation.

4.1 Model Training

After community topic mining, the observed user-item pairs are allocated into different domains. Let $\mathbf{R}^k \in \mathbb{R}^{N_k \times M_k}$ denote the rating matrix for the k-th domain, where $k = 1, \ldots, K$. M_k and N_k are the number of items and users in each domain respectively. Let $\mathbf{P}^k \in \mathbb{R}^{d \times N_k}$ and $\mathbf{Q}^k \in \mathbb{R}^{d \times M_k}$ denote the latent feature matrices in k-th domain, with column $\mathbf{p}^k_i$ and $\mathbf{q}^k_j$ represent the latent feature vectors of users and items respectively, where d denotes the dimension of latent feature. Adopting PMF model in different domains, the model is trained on rating data by minimizing the square error

$$\frac{1}{2}\sum_{i=1}^{N_k}\sum_{j=1}^{M_k} I^k_{ij}(R^k_{ij} - (\mathbf{p}^k_i)^T\mathbf{q}^k_j)^2 + \frac{\beta}{2}(\sum_{i=1}^{N_k}\|\mathbf{p}^k_i\|^2 + \sum_{j=1}^{M_k}\|\mathbf{q}^k_j\|^2) \tag{15}$$

where I^k_{ij} indicates the training data of user-item pairs belonged to domain k, $\|\cdot\|^2$ denotes the Frobenius norm to make the solution more robust, and β is the regularization coefficient. One important difference between the PMF and our model is that we consider the training process across each domain. Therefore, we have K objective functions in total. The parameters $\mathbf{p}^k_i$ and $\mathbf{q}^k_j$ in Eq.(15) can be minimized by Alternating Least Square(ALS) method, which performs the following two updates alternatively.

First, optimizing Eq.(15) with respect to $\mathbf{p}^k_i$ for $i = 1, 2, \ldots, N_k$ in domain k and fixing all $\mathbf{q}^k_j$ leads to

$$\mathbf{p}^k_i = (\sum_{j=1}^{M_k} I^k_{ij}\mathbf{q}^k_j(\mathbf{q}^k_j)^T + \beta\mathbf{I}_d)^{-1}(\sum_{j=1}^{M_k} I^k_{ij}R^k_{ij}\mathbf{q}^k_j) \tag{16}$$

Then, optimizing with respect to $\mathbf{q}^k_j$ for $j = 1, 2, \ldots, M_k$ in domain k and fixing all $\mathbf{p}^k_i$ leads to

$$\mathbf{q}^k_j = (\sum_{i=1}^{N_k} I^k_{ij}\mathbf{p}^k_i(\mathbf{p}^k_i)^T + \beta\mathbf{I}_d)^{-1}(\sum_{i=1}^{N_k} I^k_{ij}R^k_{ij}\mathbf{p}^k_i) \tag{17}$$

In order to avoid overfitting on test data, we use the weighted-regularization $\sum_{j=1}^{M_k} m^k_{v_j}\|\mathbf{q}^k_j\|^2$ and $\sum_{i=1}^{N_k} n^k_{u_i}\|\mathbf{p}^k_i\|^2$ instead of the original regularization terms in Eq.(15) in experiment, where $m^k_{v_j}$ and $n^k_{u_i}$ denote the number of ratings of item v_j and user u_i in $\mathcal{D}_k$, respectively.

4.2 Recommendation with Domains

Since top-n recommendation is to produce a ranking list of items with high to low preference, it is necessary to predict ratings for users towards items in each domain first. Corresponding to training process, we consider a two-step filtering to recommend items in the framework of TopRec.

Given a user-item pair (u_i, v_j), the first step is to judge whether the user u_i is interested in the specific domain $\mathcal{D}_k$ where the item v_j belongs to. If she is interested in $\mathcal{D}_k$, the following rating prediction step is implemented using the learnt domain-specific user and item latent feature parameters $\mathbf{p}^k_i$ and $\mathbf{q}^k_j$. Otherwise, we do not predict rating for the user-item pair. In detail, the whole process can be summarized in a unified form

$$\hat{R}_{ij} = \begin{cases} (\mathbf{p}^k_i)^T\mathbf{q}^k_j + r^k & \text{if } u_i \in \mathcal{D}_k \cap v_j \in \mathcal{D}_k \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

where $r^k \in \mathbb{R}$ denotes the global offset of $\mathcal{D}_k$ by averaging observed ratings in the domain. For each node, we sort the predicted rating $\hat{R}_{ij}$ of all the K domains with a decreasing order and then the top-n item list is eventually generated. Basically, TopRec can filter out lots of items for users according to their uninterested domains and then recommend items with high predicted rating score in their interested domains.

5. EXPERIMENTS

In this section, we investigate the performance of TopRec in top-n recommendation task compared to other state-of-the-art algorithms on two real-world datasets. Also, we report the results for different settings of model parameters.

5.1 Datasets

We examine how the TopRec behaves on two multi-domain product review datasets with trust networks: Epinions and Ciao[2]. Both of them are well-known consumer opinion websites where users not only provide reviews to their familiar products but also maintain trust lists of their trusting users.

[2]http://www.ciao.co.uk

Table 1: Statistics of the Datasets

	Epinions	Ciao
# of Users	7,475	4,137
# of Items	140,434	72,198
# of Ratings	343,789	194,278
# of Trust Links	143,066	85,877
# of Domains	8	10
Ave Ratings per User	45.99	46.96
Ave Ratings per Item	2.45	2.69
Ave Trusts per User	19.14	20.75
Rating Sparsity	99.97%	99.93%
Trust Network Sparsity	99.74%	99.50%

The version of the two datasets[3] used in this study are published by the authors of [25] including data records until May 2011. To evaluate the effects of social network, we first remove users without trust relations. For the purpose of taking each user's top-n recommendation performance into account, we also prune users with fewer than ten reviews to ensure sufficient test data for each user. Then we use eight and ten top popular categories to define the domains for Epinions and Ciao respectively. The detailed statistics of the two datasets are showed in Table 1. Compared to Ciao, Epinions owns more users and products in their most representative categories, which results in more review data and trust links. While the scale of Epinions and Ciao are different, the sparsity of both datasets are comparable. Based on the presented statistics, the high sparsity is fairly noticeable in user-item interaction as well as trust relation. Especially, as we do not sift cold products from the original published datasets, the averaging rating per item is extremely small. Figure 2 shows the degree distribution of the two datasets. As we can see, the datasets are very sparse and suggest power law distribution.

In experiments, we randomly pick 80% of the review data to form the training set and the rest for the test set, and run ten times in each configuration.

5.2 Performance Measures

Because recommender systems in reality normally concerns about personalized ranking of entities but not rating prediction to all products, we analyze performance of each model by comparing the top suggestions in our experiments. For a consistent evaluation with the top-n recommendation literature, three classical measures commonly used are employed: MAP(*Mean Average Precision*), F-measure, and nDCG(*normalized Discounted Cumulative Gain*).

For each user, given a ranked list with n items, we denote $\text{prec}(j)$ as the precision at ranked position j, $\text{pref}(j)$ as a binary preference indicator at position j. AP is defined as

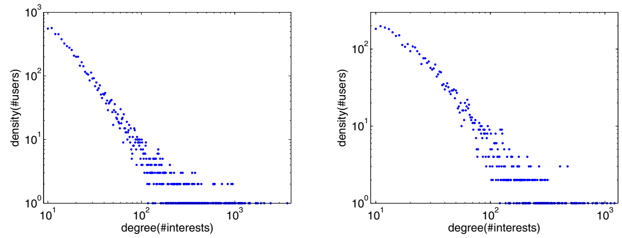
$$\text{AP}(u) = \frac{\sum_{j=1}^{n} \text{prec}(j) \times \text{pref}(j)}{\#\text{ of preferred items}} \tag{19}$$

$$\text{MAP} = \frac{\sum_{u \in \mathcal{U}} \text{AP}(u)}{|\mathcal{U}|} \tag{20}$$

To compute F-measure, let *precision* and *recall* denote the user-oriented averaging precision and recall with top-n list.

$$\text{F1} = \frac{2 \times precision \times \ recall}{precision + recall} \tag{21}$$

[3] http://www.public.asu.edu/ jtang20/



(a) Epinions (b) Ciao

Figure 2: Degree distribution of Epinions and Ciao.

To obtain nDCG, the preference indicator $\text{pref}(j)$ is also used.

$$\text{nDCG} = \frac{1}{\text{IDCG}} \times \sum_{j=1}^{n} \frac{2^{\text{pref}(j)-1}}{\log_2(j+1)} \tag{22}$$

where IDCG is produced by a perfect ranking algorithm. By this definition, nDCG gives larger credit to top-ranked entities. Higher MAP, F1 and nDCG implies better recommendation result.

5.3 Comparisons

Here we compare three variant methods on the basis of TopRec framework with four baseline methods to demonstrate the effectiveness of each part of our model.

- TopRec with Single Class (TopRec-S) In this single class model, we change user clustering criteria as: *a user* u_i *is interested in topic* z_k *if and only if* $\forall z_{k'}$, $s.t. f(z_k, u_i) > f(z_{k'}, u_i)$. Thus user membership hypothesis becomes that a user only interested in one topical domain. We set the model as a comparison of users' multi-faceted features assumption. $f(z_k, u_i)$ is still computed by Eq.(14) here.

- TopRec with Multiple Class (TopRec-M) The multiclass model based on TopRec framework believes that users are interested in multiple topical domain. Note that both of TopRec-S and TopRec-M are not added social networks.

- TopRec with Network (TopRec-Net) As to evaluate the contribution of social network, we embed network-constrained term on the basis of TopRec-M. This model is the unified model described in section 3.4.

- Probabilistic Matrix Factorization (PMF) [21]. PMF virtually is a low rank matrix factorization model and assumes that a user generates a rating for an item by adding Gaussian noise to the inner product $R_{ij} = (\mathbf{p_i})^T \mathbf{q_j}$, where $\mathbf{p_i} \in \mathbb{R}^d$ and $\mathbf{q_j} \in \mathbb{R}^d$ associate with latent factor vector of user and item.

- PMF with Domains (PMF-D). This model takes multiple domains information of items into consideration, so the PMF-D treats different domains independently but has N users in all domains.

- Multiclass Co-Clustering (MCoC) [27]. This method proposes a framework to extend traditional CF by dividing users and items into multiple subgroups. Different with our framework, it views this allocation procedure as a Multiclass Co-Clustering problem.

Table 2: **Performance comparisons of top-n recommendation on Epinions in terms of MAP, F1 and nDCG.**

Methods	n=5			n=10			n=15			n=20		
	MAP	F1	nDCG	MAP	F1	nDCG	MAP	F1	nDCG	MAP	F1	nDCG
RANDOM	0.2275	0.1156	0.1357	0.2248	0.1160	0.1109	0.2201	0.1072	0.0957	0.2162	0.0994	0.0855
PMF	0.2896	0.1714	0.1857	0.2911	0.2038	0.1709	0.2835	0.2010	0.1551	0.2759	0.1904	0.1418
PMF-D	0.3666	0.2045	0.2249	0.3593	0.2157	0.1919	0.3467	0.2032	0.1689	0.3360	0.1885	0.1520
MCoC	0.3736	0.1961	0.2492	0.3667	0.1990	0.2017	0.3628	0.1847	0.1714	0.3598	0.1726	0.1518
TopRec-S	0.2951	0.1569	0.1814	0.2904	0.1582	0.1487	0.2844	0.1555	0.1325	0.2779	0.1549	0.1234
TopRec-M	0.3953	0.2169	0.2485	0.3847	0.2206	0.2058	0.3739	0.2041	0.1781	0.3651	0.1882	0.1591
TopRec-Net	**0.4236**	**0.2386**	**0.2710**	**0.4111**	**0.2400**	**0.2235**	**0.3991**	**0.2200**	**0.1927**	**0.3896**	**0.2001**	**0.1709**

- Random Group Model (RANDOM) TopRec divides users into groups with sizes of $\{N_1, \dots, N_K\}$, so we randomly sample users into K groups with the same sizes of TopRec. The random group model is to create comparison with the process of community topic mining.

To make a fair comparison, we use PMF as the basic CF method either training on user-item domains or entire user-item matrix in all the experiments. For PMF, the latent dimensionality of low rank features is set to be $d = 10$, and the regularization coefficient is set as $\beta = 0.1$.

5.4 Experimental Protocol

To all the comparisons, we utilize the following experimental protocol.

We first notice that both of Epinions and Ciao are employed 5-star rating systems, which refers to user-item interaction are explicit. However, we also find that more than 70% ratings are 4 or 5, which means the rating distribution are fairly imbalanced. This positive ratings phenomenon also appears in many other online consumer rating datasets so that it is inevitably to train overly optimistic estimators by using the observed ratings directly. To address this phenomenon, we adopt a *bias correction* procedure mentioned in [28]. That is to draw uniformly from the set of the unobserved user-item pairs for each user as pseudo-negatives to balance the original training set. The assumption under bias correction is that the unobserved samples are less interested by users compared with observed rating data.

For the evaluation protocol, we follow the evaluation mechanism described in [28, 6]. For each user, as the total number of items is huge in the datasets, while the number of true preferred items is much smaller, it is prohibitive to take all the items as candidates and generate a total ordering of the whole item set. Our test methodology for top-n measure is: for each user, we randomly select S additional items that are not reviewed and mix them with the test data to construct a probe set. Thus we compute the predicted ratings over probe sets to find top-n products. The size of random probes per user is set as $S = 500$ in experiments.

5.5 Results and Analysis

Performance on Epinions. Table 2 shows the experimental results on the Epinions dataset with three different evaluation metrics: MAP, F-measure, nDCG, when we vary the number of returned items $n = 5, 10, 15, 20$. For the three variants of TopRec methods, we pick confidence parameter

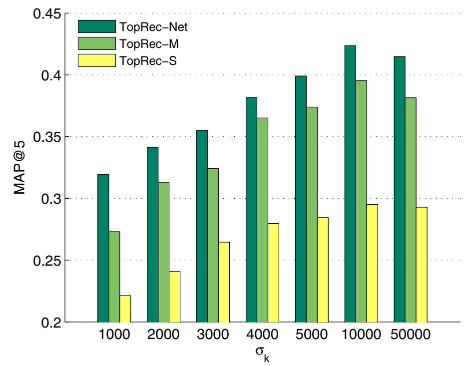


Figure 3: Impact of expert guidance confidence parameter σ_k on MAP@5 performance of Epinions.

$\sigma_k = 10000$, the number of experts 3 for each domain, and the value of regularization parameter $\lambda = 100$. Then we empirically set the number of nearest neighbors as 5, the value of Newton step parameter τ as 0.01.

From Table 2, TopRec-Net yields the best performance under all of the evaluation conditions. By looking at the trend along with the number of returned list n, we can see that all of the performance drops when the returned list n is increasing. This is mainly because that more than half of users only have less than five true interested items in test set. When n becomes large, recall improves while precision declines severely.

Compare models concerned about domain-specific CF (i.e. TopRec variants and PMF-D)with the original PMF, it is clear to conclude that the multiple domains do benefits on recommendation task. TopRec, which allocates users into their interested domains by user clustering and topic mining, outperforms PMF-D, which simply assumes that users are belonged to all the domains. By comparing the three variants of TopRec, we show that TopRec-M and TopRec-Net perform better than TopRec-S, which demonstrates the multi-faceted assumption for online users. Another phenomenon is that TopRec-Net outperforms TopRec-M consistently, which infers that network-constrained topic modeling can bring about performance improvement.

To illustrate the effectiveness of our community topic mining stage, we compare our methods with the state-of-the-art approach MCoC. Both of TopRec and MCoC aim to map users and items into subgroups, but we take advantage of the

Table 3: Performance comparisons of top-n recommendation on Ciao in terms of MAP, F1 and nDCG.

Methods	n=5			n=10			n=15			n=20		
	MAP	F1	nDCG	MAP	F1	nDCG	MAP	F1	nDCG	MAP	F1	nDCG
RANDOM	0.1915	0.0832	0.1139	0.1917	0.0781	0.0872	0.1891	0.0699	0.0727	0.1867	0.0642	0.0638
PMF	0.2296	0.1127	0.1415	0.2357	0.1484	0.1327	0.2312	0.1555	0.1246	0.2258	**0.1534**	0.1169
PMF-D	0.3002	0.1434	0.1760	0.3004	0.1567	0.1508	0.2943	0.1538	0.1339	0.2854	0.1483	0.1224
MCoC	0.3029	0.1265	0.1960	0.2999	0.1418	0.1619	0.2971	0.1329	0.1364	0.2930	0.1233	0.1197
TopRec-S	0.3277	0.1456	0.2011	0.3230	0.1282	0.1488	0.3173	0.1137	0.1231	0.3103	0.1052	0.1083
TopRec-M	0.3839	0.1732	0.2359	0.3787	0.1629	0.1811	0.3706	0.1468	0.1515	0.3634	0.1332	0.1325
TopRec-Net	**0.4025**	**0.1843**	**0.2501**	**0.3963**	**0.1766**	**0.1946**	**0.3878**	**0.1613**	**0.1641**	**0.3878**	0.1479	**0.1431**

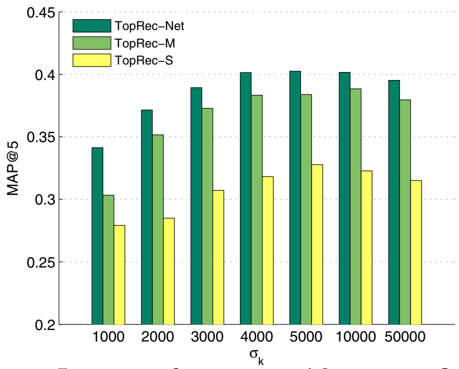
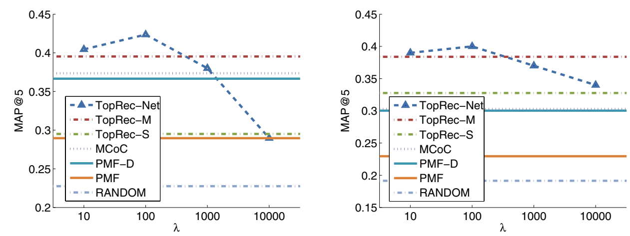


Figure 4: Impact of expert guidance confidence parameter σ_k on MAP@5 performance of Ciao.



(a) Epinions (b) Ciao

Figure 5: Impact of regularization parameter λ on MAP@5 performance

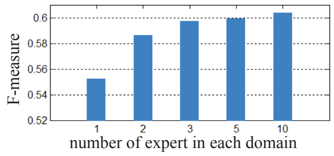
category relations and employ semi-supervised topic model for user clustering instead of user-item Co-Clustering. The experimental results illustrate that our model works better than MCoC. The reason is as follows. MCoC conducts Co-Clustering only by ratings. However, the user-item rating matrix we faced with is highly sparse, few observed interactions are available for MCoC model. To compensate this, our model combines rating matrix with complementary knowledge of item categories and social networks. At last, we show that user clustering and topic alignment are pivotal for overall recommendation performance by the comparison of TopRec and RANDOM.

Performance on Ciao. Ciao is a smaller dataset with few ratings. We select expert confidence parameter $\sigma_k = 5000$ according to the quantities of observed ratings. Other parameters mentioned above are set the same as Epinions. Experimental results on Ciao are reported in Table 3. It is evident that TopRec-Net outperforms the other methods in almost all cases on this dataset. Different from Epinions data, TopRec-S behaves better than MCoC and PMF-D this time. This is because more users are interested in one domain in the dataset. Similar to Epinions, the number of interested products for majorities is still less than five in test set, which leads to the performance getting worse as the recommended lists expending.

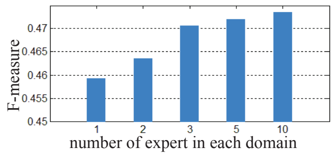
Parameter Study. The TopRec model mainly has two important parameters. We firstly study the effect of confidence parameter σ_k for expert guidance on Epinions and Ciao. The confidence weight of each domain is set from 1000 to 50000. The larger σ_k means the model has more confidence on experts. In Figure 3 and 4, MAP of the top 5 list are plotted as a function of σ_k for Epinions and Ciao respectively. As we increase σ_k, the performance in both Figure first increases and thereafter declines slightly. This observation coincides with the interpretation of experts-guided topic modeling: besides user collection $\mathcal{U}$, we adding a pseudo counts $\sigma_k p(v_j|\bar{u}_k)$ for item v_j, therefore it would be better that σ_k is equivalent to sample size. Yet, if we give a much higher confidence weight to experts, experimental results show that the performance would not raise any more. From the figures, we can see that the trends are quite similar on both of the datasets. According to their sample size, the optimal confidence weight is $\sigma_k = 10000$ for Epinions, and around $\sigma_k = 5000$ for Ciao.

Now, we discuss the second essential parameter of TopRec, λ. Figure 5 shows how the social regularization parameter λ impacts the performance of TopRec-Net. We vary $\lambda \in \{10, 100, 1000, 1000\}$ where larger λ enhance penalty of the disagreement of interest distribution between social neighbors. When λ is small, the unified model TopRec-Net behaves like TopRec-M which does not consider social link. When λ increase, the social regularization term becomes more influential on the model and brings the network information into the community mining. For Epinions and Ciao, the performance reaches the peaks at $\lambda = 100$. Nevertheless, if λ becomes increasingly large, the social smoothness term would overwhelm the rating information which is responsible for community mining, as well as prior knowledge which is in charge of topic alignment. As a result, the performance drops dramatically on the two datasets.

Further Probing on Topic Mining. A key reason for the performance improvement of TopRec is to utilize semi-supervised topic model to mine meaningful topics and align



(a) Epinions



(b) Ciao

Figure 6: Topic mining results of TopRec-Net under expert guidance.

them with communities. Here we would like to probe further how the topic mining works.

In our framework, community topic mining is posed as an intermediate step and resulting topical communities are used in domain-specific recommendation. The ultimate goal of topic mining for communities is to discover interpretable topics for each user. Yet, it is rather difficult to get a good topical summary for community by a pure unsupervised topic model. Hence, we resort to semi-supervised topic modeling, in which expert guidance is the critical component deciding whether the model can derive desired user clustering results.

To analyze the performance of topic mining, the greatest challenge is we do not know the true topics interested by users. However, the topics in our model could be interpreted by item categories. That is to say, for us ers, the categories of preferred items in test set are capable to represent their true labels. After topic mining, a series of predicted topics are obtained for users. Then we could evaluate mining results by combining precision and recall with respect to topics. In Figure 6, F-measure on Epinions and Ciao of TopRec-Net is presented as a function of the number of expert. Particularly, the number of experts for each domain is chosen from the range $\{1, 2, 3, 5, 10\}$. In implementation, when there is more than one expert in each domain, their rating distributions over items are added into the vector $c(\bar{u}_k, v_j)$ of Eq.(2). By viewing the tendency of clustering accuracy along with the number of experts in each domain, we can see that, more expert guidance leads to better clustering results.

6. RELATED WORK

In this section, we review related works for recommender systems with collaborative filtering, especially for model-based CF approaches. CF can be classified into two different approaches: memory-based algorithms and model-based algorithms. Memory-based CF algorithm usually search for the similar users or items to produce a prediction or top-n recommendation [22, 9]. Although memory-based approaches are easy implemented and commonly used in reality, they are normally limited by highly sparse data [24], since the similarity cannot be estimated accurately in this case.

In model-based approaches, a compact model employed machine learning and statistical techniques is trained from the known ratings. There are many model-based algorithms proposed, such as latent factor models [11, 23], graphical models [13], clustering models [5, 8], and Bayesian model [2].One popular latent factor model is low-rank matrix factorization whose premise is that users' preferences are only influenced by a small number of factors, so it uses low-rank latent factors to approximate rating data [21, 14, 20]. From the view of matrix completion, the low-rank factorization models are competent in tasks with a large amount of missing entries. Many evidence have shown that matrix factorization models outperform other CF approaches.

Recently, several approaches resort to trust-aware collaborative filtering [25, 17, 12] where users are no longer treated as independent and identically distributed. All these methods based on common rationale that users are likely to have similar tastes with their trusted friends in social networks. Previous studies manifest that social relations as another form of user information could alleviate sparsity problem and improve recommendation accuracy. Most of the existing social recommendation methods focus on encoding social network in user profile learning. However, many other web mining tasks show that when it comes to mining topics and relationships among objects, network structure are helpful [18, 3]. Our work employ the network-regularized topic modeling into a novel use for user clustering in recommender systems.

Apart from social relations, other relations such as item-category could also been incorporated into recommender systems to make up the lack of rating information. In [32], an extension of the probabilistic matrix factorization to multi-domain case is proposed. Through learning a covariance matrix, rating knowledge is transferred across domains adaptively. Yang et al. [31] model category-specific trust circle from ratings and trust links and formulate multi-faceted trust network into social matrix factorization. Different with [32], they evaluate the model in each domain without given the overall performance.

Clearly, we could take advantage of external information such as user's trust network and item's category to compensate the sparse data. On the other hand, there are more dense patterns or groups in the original rating matrix which can be uncovered though clustering methods. Traditional clustering CF models cluster users based on the items they rated or cluster items based on the users that rated them [5, 24]. However, these models overlook the user-item similarity during clustering procedure. To avoid the shortcomings, co-clustering models are proposed to map user and item into clusters simultaneously, and thus each cluster becomes more dense than the entire rating matrix [8, 27].

In this study, the proposed TopRec model leverages the power of clustering method and external relations to obtain domains with explicit topics. In experiments, we have shown that this combination works well on highly sparse datasets.

7. CONCLUSION

Effectively modeling interest topics for users and accordingly recommending their preferred items are fundamental issues to all recommender systems. In this paper, we propose a novel framework, TopRec, jointly exploiting community topic mining and domain-specific collaborative filtering for top-n recommendation task. To construct domains, we integrate expert guidance with social network to establish a unified probabilistic topic model for community topic mining. Then we utilize the observed user-item ratings across different domains for collaborative filtering. Experimental results on two datasets from real-world consumer review websites have demonstrated that the proposed method produces more accurate recommendation than other competitors.

8. ACKNOWLEDGEMENT

This work was supported in part by the 973 Program under Project 2010CB327905, by the National Natural Science Foundation of China under Grant No. 61170127 and 61070104.

9. REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[2] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52, 1998.

[3] D. Cai, Q. Mei, J. Han, and C. Zhai. Modeling hidden topics on document manifold. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 911–920, 2008.

[4] A. Clauset, M. Newman, and C. Moore. Finding community structure in very large networks. *Physical review E*, 70(6), 2004.

[5] M. Connor and J. Herlocker. Clustering items for collaborative filtering. In *SIGIR 2001 Workshop on Recommender Systems*, 2001.

[6] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*, 2010.

[7] D. M. Fleder and K. Hosanagar. Recommender systems and their impact on sales diversity. In *Proceedings of the 8th ACM conference on Electronic commerce*, pages 192–199, 2007.

[8] T. George and S. Merugu. A scalable collaborative filtering framework based on co-clustering. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, 2005.

[9] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237, 1999.

[10] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, 1999.

[11] T. Hofmann. Latent semantic models for collaborative filtering. 22(1):89–115, 2004.

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

[13] R. Jin, L. Si, and C. Zhai. Preference-based graphic models for collaborative filtering. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, UAI'03, pages 329–336, 2003.

[14] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434, 2008.

[15] G. Linden, B. Smith, and J. York. Amazon.com recommendations: item-to-item collaborative filtering. *Internet Computing, IEEE*, 7:76 – 80, 2003.

[16] Y. Lu and C. Zhai. Opinion integration through semi-supervised topic modeling. In *Proceedings of the 17th international conference on World Wide Web*, pages 121–130, 2008.

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

[18] Q. Mei, D. Cai, D. Zhang, and C. Zhai. Topic modeling with network regularization. In *Proceedings of the 17th international conference on World Wide Web*, pages 101–110, 2008.

[19] K. Onuma, H. Tong, and C. Faloutsos. Tangent: a novel, 'surprise me', recommendation algorithm. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 657–666, 2009.

[20] J. D. M. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd international conference on Machine learning*, pages 713–719, 2005.

[21] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, 2008.

[22] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295, 2001.

[23] L. Si and R. Jin. Flexible mixture model for collaborative filtering. In *Proceedingins of the 20th international Conference on Machine Learning*, pages 704–711, 2003.

[24] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advance in Artificial Intelligence*, pages 4:2–4:2, 2009.

[25] J. Tang, H. Gao, and H. Liu. mtrust: discerning multi-faceted trust in a connected world. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 93–102, 2012.

[26] R. Wetzker, W. Umbrath, and A. Said. A hybrid approach to item recommendation in folksonomies. In *Proceedings of the WSDM '09 Workshop on Exploiting Semantic Annotations in Information Retrieval*, pages 25–29, 2009.

[27] B. Xu, J. Bu, C. Chen, and D. Cai. An exploration of improving collaborative recommender systems via user-item subgroups. In *Proceedings of the 21st international conference on World Wide Web*, pages 21–30, 2012.

[28] S.-H. Yang, B. Long, A. Smola, N. Sadagopan, Z. Zheng, and H. Zha. Like like alike: joint friendship and interest propagation in social networks. In *Proceedings of the 20th international conference on World wide web*, pages 537–546, 2011.

[29] S.-H. Yang, B. Long, A. J. Smola, H. Zha, and Z. Zheng. Collaborative competitive filtering: learning recommender using context of user choice. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 295–304, 2011.

[30] T. Yang, R. Jin, Y. Chi, and S. Zhu. Combining link and content for community detection: a discriminative approach. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 927–936, 2009.

[31] X. Yang, H. Steck, and Y. Liu. Circle-based recommendation in online social networks. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1267–1275, 2012.

[32] Y. Zhang, B. Cao, and D.-Y. Yeung. Multi-domain collaborative filtering. In *Proceedings of the Twenty-Sixth conference on Uncertainty in artificial intelligence*, pages 725–732, 2010.