Connecting Comments and Tags: Improved Modeling of Social Tagging Systems

Dawei Yin[†] Shengbo Guo[S] Boris Chidlovskii[S]
Brian D. Davison[†] Cedric Archambeau[S] Guillaume Bouchard[S]
[†]Dept. of Computer Science and Engineering, Lehigh University, Bethlehem, PA, USA
[S]Xerox Research Center Europe, Grenoble, France

ABSTRACT

Collaborative tagging systems are now deployed extensively to help users share and organize resources. Tag prediction and recommendation can simplify and streamline the user experience, and by modeling user preferences, predictive accuracy can be significantly improved. However, previous methods typically model user behavior based only on a log of prior tags, neglecting other behaviors and information in social tagging systems, e.g., commenting on items and connecting with other users. On the other hand, little is known about the connection and correlations among these behaviors and contexts in social tagging systems.

In this paper, we investigate improved modeling for predictive social tagging systems. Our explanatory analyses demonstrate three significant challenges: coupled high order interaction, data sparsity and cold start on items. We tackle these problems by using a generalized latent factor model and fully Bayesian treatment. To evaluate performance, we test on two real-world data sets from Flickr and Bibsonomy. Our experiments on these data sets show that to achieve best predictive performance, it is necessary to employ a fully Bayesian treatment in modeling high order relations in a social tagging system. Our methods noticeably outperform state-of-the-art approaches.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.2.8 [**Database Management**]: Database applications—*Data Mining*; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.1.2 [**Models and Principles**]: User/Machine Systems—*Human Information Processing*

Keywords

social tagging; tag recommendation; personalized tag prediction; personalized comment recommendation; co-factorization

WSDM'13, February 4–8, 2013, Rome, Italy.
Copyright 2013 ACM 978-1-4503-1869-3/13/02 ...$15.00.

1. INTRODUCTION

Collaborative tagging systems have become widely used for sharing and organizing resources in recent years. In social bookmarking, as one type of collaborative tagging system, users can add metadata in the form of descriptive terms, called tags, to describe web resources. Social bookmarking systems have been utilized successfully in many areas, such as web search [5], personalized search [36], web resource classification [39] and clustering [26]. However, the major bottleneck for large scale applications of social tagging systems is that only a small number of web resources have manually assigned tags, compared to the size of the Web. Therefore, systems that can automatically tag web resources are needed. On the other hand, from the user's perspective, a system that can provide suggestions and recommendations when users are about to assign tags to new resources can improve human-computer interactions and organization of the knowledge base as well.

Motivated by the needs described above, researchers have considered how to build systems to recommend or predict tag usage. Early work in this area, such as Hotho et al. [14] and Lipczak et al. [18], has demonstrated two basic approaches—content-based and graph-based methods—to tackle the problem. Recent work, including Rendle et al. [27] and Yin et al. [38, 37], show how personalized tag recommenders that take the user's previous tagging behaviors into account usually have better performance. However, most previous methods for social tagging system only model user behavior based on the log of prior tags [27] and item content [18, 38], neglecting the other behaviors and information found in real-world social tagging systems. In such social tagging systems, there exist many other artifacts of user activities, such as comments on items [4] and social connections with other users. Unlike tags which are usually personalized descriptions of the item, users' comments on the items are user-generated opinionated texts. Although they share some similar patterns of behaviors (both are user-generated texts for specific items), they have different properties and aims. Personalized tag prediction has been studied for several years, but comment prediction, which is quite different from traditional opinion mining, is rarely investigated. Recently, Agarwal et al. [4] develop personalized comment recommendation via factor models but they do not predict the content (e.g., term frequency) of personalized comments which could potentially help in the interpretation of comments and be applied to improve sentiment analysis of comments. In addition, the effects of mutual reinforcement across contexts such as user-tag-item and user-comment-item are still unknown.

In addition to user-generated tags and comments, users are often also able to denote friendship (via links) with other users. All of these activities provide potential hints for tag prediction, comment prediction and prediction of other user behaviors. By analyzing all of these activities, we can better capture users' preferences and make more accurate recommendations, but many of these activities are coupled and their effects on each other are not easily modeled.

In this paper, we systematically investigate the coupled activities of users and their mutual effects in a social tagging system. Our explanatory analyses show that the main challenges in modeling tagging systems are from three points: coupled high order interaction, data sparsity, and cold start on items. We tackle these problems by proposing a generalized Bayesian probabilistic latent factor model which can be tailored to fit the tagging system. We conduct empirical evaluations on two public data sets—Flickr and Bibsonomy. The experiments show that in social tagging systems, a user's commenting and tagging behaviors are highly correlated and can be mutually inferred, which has not been explored previously. The contributions of this paper are summarized as follows:

1. To better model social tagging systems, we propose a novel generalized latent factor model which is based on a Bayesian approach.
2. We find that connecting comments and tags within the same model permits mutual reinforcement and improves overall performance.
3. Experiments on real-world data show that our Bayesian methods can achieve much better performance than the probabilistic version of our model, due to the sparsity of the high-order relational data in social media. Our model significantly outperforms state-of-the-art methods.

The paper is organized as follows: Section 2 discusses related work. We present preliminary experiments in Section 3. Section 4 presents the proposed model, followed by describing an efficient and scalable approach developed for estimating the model parameters in Section 5. Section 6 presents the empirical evaluations of the proposed approach on both data sets. Section 7 concludes the paper.

2. RELATED WORK

Personalized tag recommendation, as a special case of collaborative filtering, is a recent topic in recommender systems. Two main directions for these systems are content-based approaches and graph-based approaches.

Content-based methods, which usually encode users' preferences from textual information within items (e.g., web pages, academic papers, tags), can predict tags for new users and new items. State-of-the-art content-based tag recommendation systems [18, 37] utilize several tag sources including item content and the user's previous history to build profiles for both users and tags. New tags are checked against user profiles, which are a rich, but imprecise source of information about user interests. The result is a set of tags related both to the resource and user.

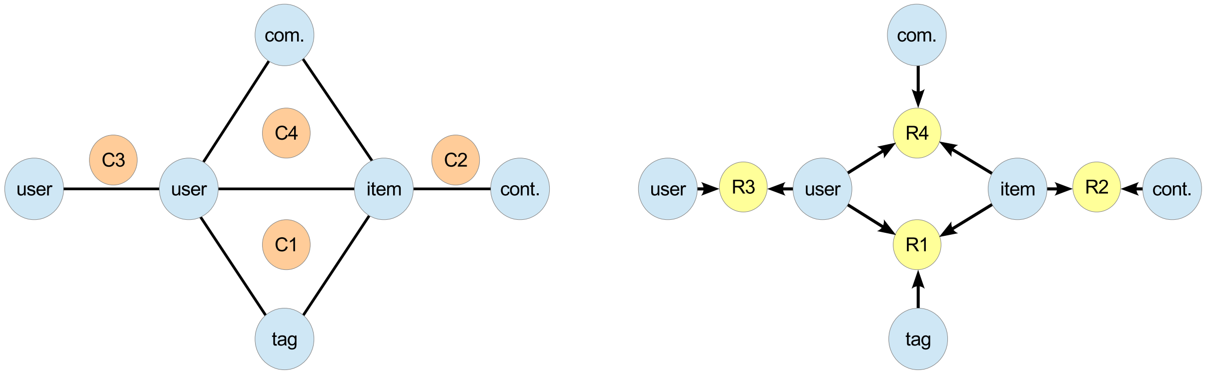
Graph-based approaches, which usually have stronger assumptions than content-based ones (e.g., every user, every item and every tag has to occur in at least p posts), can often provide better performance. An example of early work is FolkRank, introduced by Hotho et al. [14], which is an adaptation of PageRank that can generate high quality recommendations and is shown to be empirically better than previously proposed collaborative filtering models [16]. Guan et al. [10] propose a framework based on the graph Laplacian to model interrelated multi-type objects involved in a tagging system. More recently, factorization models (also considered as graph-based approaches) have shown success on personalized tag recommendation problems. Symeonidis et al. [33] propose a method based on Higher-Order-Singular-Value-Decomposition (HOSVD), which corresponds to a Tucker Decomposition (TD) model optimized for square-loss where all not observed values are learned as 0s. Rendle et al. [27, 28] present a better learning approach for TD models, which is to optimize the model parameters for the AUC (area under the ROC-curve) ranking statistic. The main problem of the graph-based methods is that they can only predict tags for certain groups of existing users and web resources. Thus, a better tag recommender should be able to recommend tags for new users or new web resources, and still have reasonably good performance.

Non-personalized tag recommenders—i.e., for a given item they recommend to all users the same tags—have also attracted attention (e.g., [12, 32]). However, Rendel et al. [27] shows that personalized tag recommendation systems empirically outperform the theoretical upper bound for any non-personalized tag recommender.

Existing opinion mining or sentiment analysis studies [8, 15, 19] focus on summarizing and classifying comments, and discard higher-order relations for user-comment-item. In contrast, we focus on predicting users' opinions (i.e., the terms used) for an item instead of simply classifying the comment contents. Recently, Agarwal et al. [4] develop personalized comment recommendation via factor models but they do not predict the content (e.g., term frequency) of personalized comments.

There exist a number of latent factor methods. One related method is collective matrix factorization from Singh and Gordon [31] which provides a general framework to model multi-relational data, extending many previous approaches for matrix factorization in the presence of additional features. These extensions of matrix factorization/factor analysis tend to be limited to two or three relations to take into account contextual information (such as user-specific and movie-specific features) in a recommender system [42, 1, 2, 20]. For instance, Zhu et al. [42] propose to make use of links and content for web page classification. More recently, Agarwal and Chen [3] incorporate explicit features of users and items into latent factor models, and Ma et al. [20] propose to improve recommendation quality based on social regularization. However, these methods only model two factor data and cannot be directly used in a social tagging system which is naturally a higher-order system.

More generally, co-factorization models [11, 40, 41] make recommendations across multiple contexts or domains. While the framework proposed by Singh and Gordon [31] is fairly general, the key weakness is that it does not enable the handling of high-order relations, such as that needed in a social tagging system and it does not use Bayesian estimation to tackle the problem of data sparsity. Modeling higher or-



(a) Clique relations among the entity types serve as contexts

(b) Bipartite graph between relations and entity types

Figure 1: An example of four relations on five entity types in a social tagging system. R_1 is the tag post context (user-tag-item), R_2 is the item-content context (item-content feature), R_3 is the social network context (user-user) and R_4 is the comment context (user-comment-item).

der data in social media is often neglected in existing factor models.

3. PRELIMINARY EXPERIMENTS

In this section, we review characteristics of social tagging systems and describe the challenges and problems in modeling user behaviors such as tagging and commenting.

Unlike traditional collaborative filtering and recommendation tasks, in social tagging systems, a user's tagging and commenting activities generate relations involving more than two types of entities. In contrast, most traditional work focuses on second order relations that involve just two types of entities (e.g., user-item). In social tagging systems, the posts (that is, each tag produced by a user for an item) are by nature third order data [28, 27, 38, 37] that we consider as a triple (user-tag-item). Figure 1(a) shows that in a tagging system, users, tags and items pairwise interact and compose a clique. Similarly, users, tags and comments also interact pairwise. For the tag and comment prediction tasks, we cannot drop any one of user, tag/comments, or item. By involving the temporal factor, it even becomes fourth order data [37]. However, these types of higher order relations have rarely been studied due to the complexity and difficulty in modeling and inference.

On the other hand, the relational data from different contexts are coupled together. In Figure 1(a), we can see the social tagging entity relations: there exist four cliques in this social tagging system (user-tag-item, user-comment-item, user-user, item-content). Within these cliques, all involved entities interact with all others. Activities within these cliques are also strongly correlated with each other: for instance, activities where users comment on items or where users rate items share two of the same types of entity—user and item. With Figure 1(a), after recognizing the cliques, we can define them as contexts. Each context can be considered a type of observation individually and generated by the associated entities. In Figure 1(b), we see the directed bipartite graph, which describes which entities contribute to the process of generating each context. These contexts are frequently coupled together by sharing the same entities, increasing the difficulty of modeling.

3.1 Data sets

In this section, we conduct some simple analysis on two data sets: Flickr and Bibsonomy. The main data set is from Flickr. We crawl the data from **Flickr** by using the social-network connectors (Flickr API).[1] This data set includes 2,866 users, 60,339 tags, 32,752 comment terms and 46,733 items (e.g., images), leading to the four relations shown in Figure 1(b). The remaining dataset is a public dataset. The **Bibsonomy** dataset is from the ECML PKDD 09 Discovery Challenge Workshop[2] which includes two relations: user-tag-item and item-content.

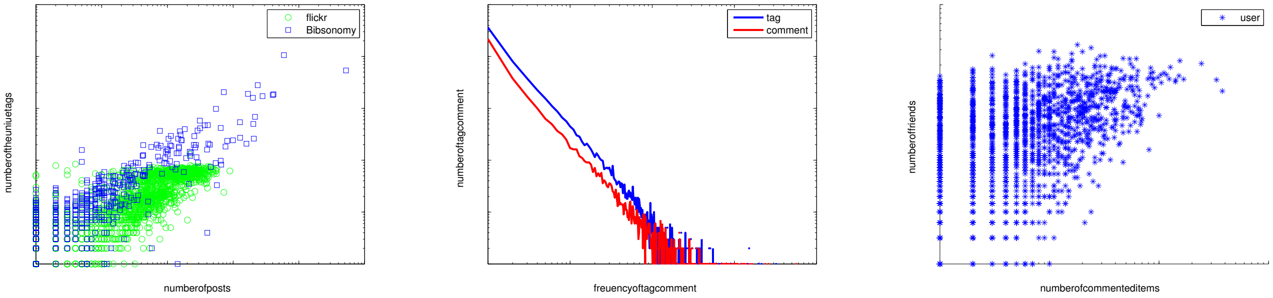
In order to observe the variety of user interests in both datasets, for each user, we calculate and plot the total number of tags, and the total number of posts. In Figure 2(a), we can see that the two datasets have different properties and users form two clusters. In Bibsonomy, users typically apply a larger variety of tags across fewer posts, suggesting that their interests are more varied. In contrast, the users in Flickr use fewer tags and their interests are more focused, reusing their tags many times. This suggests that it may be easier to track user interests in Flickr.

3.2 Coupled higher-order system

We now conduct some simple analyses for different relations. At first, we check the distribution of tags and terms in the comments. Figure 2(b) shows a linear relationship between the of number of tags/terms and the frequency of tags/terms in log scale. We can see that the distributions over both tags and comment terms are very similar and show two straight lines with the same slope in the log-log plot. In the (user, comment, item) relation, among the 21,881 records where a user comments on an item, 8,307 records show that the user also tags the same item, meaning if a user comments on an item there will be around 1/3 chance that the user will also contribute a tag on that item. This

[1] http://www.flickr.com/services/api/

[2] http://www.kde.cs.uni-kassel.de/ws/dc09/



(a) The number of unique tags as a function of the number of posts for each user across datasets

(b) Distribution of tag/comments frequency in Flickr

(c) For each user, number of friends as a function of the number of commented items in Flickr

Figure 2: Representative analyses for different activities.

Table 1: Fractions of new users, items, or tags in samples from each data set.

	Bibsonomy	Flickr
New/Total Users	41/668	23/1000
New/Total Items	602/668	1000/1000
New/Total Tags	321/2207	175/4123

evidence shows the strong connection between the relation (user, comment, item) and the relation (user, tag, item).

Figure 2(c) shows the coupling between user-user interaction and commenting activity. From the figure, we can see that most users are located on the upper left portion of the Figure. Some users with many friends may NOT comment at all (or very little) but users who frequently comment items usually have many friends. We also note that the inverse does not hold.

3.3 Cold start

As in our earlier work in tag prediction [38, 37], we employ online evaluation[3] in which only training posts which have earlier timestamps than those of the test posts are used. Note that this implies that the available training data is different for each test post and, for items tagged earlier in the timeline, fewer training data are available. While the online evaluation approach naturally fits the real-world case in which every post is used for testing a model trained on all prior posts, its feasibility depends highly on the efficiency of the training method as a new model may be necessary for each post. Instead, we can estimate the performance of the complete system by performing evaluation on only a sample of test posts, and largely avoid model-building efficiency concerns for the purpose of evaluation of effectiveness.

We utilize the online evaluation model and conduct time-sensitive sampling experiments on two data sets. For the Bibsonomy dataset, we use the same sampling dataset as in Yin et al. [38] which includes 668 test posts. For Flickr, we randomly choose 1000 posts. In all cases we effectively simulate a system running—the tagging system operates in an incremental mode. The data set statistics (shown in Table 1) demonstrate that in Bibsonomy data, we face a new user (a user which is not in any prior data) in 6.1% of the cases, and in 90.1% of the time users are trying to bookmark a "new item" not previously seen by the system. In addition, there is 13.9% chance that users would use new tags (which do not appear in the system before).

This shows that most of the time (i.e., 86.1% of posts) it is feasible to predict tags based only on past tags. The other dataset also shows similar distributions. Thus, in the real world, the principal difficulty is to handle cases in which existing users try to tag new items and therefore strictly graph-based recommenders (e.g., [27, 28]) will not be able to make recommendations most of the time. This also suggests that incorporating external information, such as item or comment content into the model might help process these cold start cases.

3.4 Data sparsity

Another notorious problem in most social media systems is data sparsity. Here, we define the number of observations over the total number of entries in the relations as the density of the data. For comparison, in one MovieLens data set[4], there are 1,000,000 ratings for 6,000 users and 4,000 movies, so the data density is just 4.17%, rendering it 95.83The sparsity of data is even more serious when the relation is higher-order and coupled in a social tagging system: in our Flickr data, there are 373,125 records of the user-tag-item relation, so the density of the context user-tag-item is 4.6170×10^{-8} $(373125/(2866 \times 60339 \times 46733))$, and for the context user-comments-item, there are 218161 records, so its density is 3.8518×10^{-8} $(218161/(2866 \times 60339 \times 32752))$. Similarly in our Bibsonomy data, the data density is 3.52×10^{-8}. Thus, data sparsity is dauntingly higher in social tag data than the traditional two-dimensional recommendation problem. The serious problem of sparsity in higher order relations strongly suggests Bayesian treatment. Previous work has already shown the significant advantage of the Bayesian approach in processing sparse data, such as in the comparison of LDA [6] to PLSA [13] and BPMF [29] to PMF [30].

4. MULTI-RELATIONAL DATA MODEL

To address the problems described above in Section 3, here we propose a novel latent factor model to handle coupled

[3] In this paper, online mode means an incremental mode of a real tagging system rather than real-time tag prediction.

[4] http://www.grouplens.org/node/73

K	Number of entity types.
N_k	Number of entities of type k.
D	Latent feature dimension.
V	Number of relations.
Θ_k	Latent features for entities of type k.
R_v	Set of relation v observations.
M_v	Total number of observations of relation v.
S_v	List of indices identifying the types of relation v.
α_v^{-1}	Variance of the observations of relation v.

Table 2: Summary of the notation used.

higher-order data in social tagging systems. We describe a Bayesian treatment to learn the parameters in the model.

An activity performed by a user in a specific social tag context induces a relation; for instance, the activity consisting of the triple (user-comment-item) is a 3-order relation with three types of entities. Let us consider a coupled higher order relational dataset with K types of entities. There are V possible relations among the entities and, for each entity type $k \in \{1, \dots, K\}$, there are N_k possible entities. Each relation $v \in \{1, \dots, V\}$ is associated with the list S_v of the entity types involved in relation v, that is $S_v = (S_{v1}, \dots, S_{v|S_v|})$ with $S_{vj} \in \{1, \dots, K\}$. Relations are then encoded by multi-dimensional arrays, where dimension j is indexed by entity type S_{vj}. The data associated with relation v are the observed triplets $\mathcal{D} = (v_m, \mathbf{i}_m, r_m)_{m=1}^{M}$ where for the m^{th} observation, $v_m \in \{1, \dots, V\}$ is the index of the relation and $\mathbf{i}_m = (i_{m1}, \dots, i_{m|S_{v_m}|})$ is a list of entity indices identifying the observation with value $r_m \in \mathbb{R}$.

Our probabilistic multi-relational data model assumes that each entity can be represented by a latent (i.e., unobserved) continuous feature vector in $\mathbb{R}^D$, where D is typically small (e.g., of the order of 10 or 100). The low-dimensional latent features are denoted by $\boldsymbol{\Theta} = (\Theta_1, \dots, \Theta_K)$, where $\Theta_k = (\boldsymbol{\theta}_{k1}, \dots, \boldsymbol{\theta}_{kN_k})^T \in \mathbb{R}^{N_k \times D}$ contains the feature vectors associated to entity type k.

A summary of notation is shown in Table 2. To facilitate understanding of the notation, we consider the example described in Figure 3 where there are four relations and five entity types: u for users, i for items, f for item features, t for tags and c for comment terms. The four relations are coupled together by linking the same types of entities. Two of these four relations linking different entity types forms a 3-dimensional array, while the other two relations are encoded as two 2-dimensional arrays. To this end, we can define S as $\{S_1, S_2, S_3, S_4\}$, where $S_1 = \{u, i, t\}$, $S_2 = \{i, f\}$, $S_3 = \{u, u\}$ and $S_4 = \{u, c, i\}$.

Figure 4 shows the graphical model for multi-relational data factorization. The model assumes multilinear links in order to predict the mean of the observations given the latent features of the corresponding entities. Formally, this means that for an observation of relation v with indices $\mathbf{i} = (i_1, \dots, i_{|S_v|})$, the mean of the observation r is a multi-dimensional dot-product $\langle ., \cdots, . \rangle$ defined as

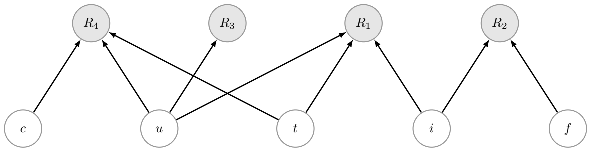


Figure 3: A bipartite graph representation of tagging system shown in Fig 1(b)

$$\langle \boldsymbol{\Theta}_{\mathbf{i}} \rangle = \langle \theta_{i_1}, \cdots, \theta_{i|S_v|} \rangle = \sum_{d=1}^{D} \prod_{k \in S_v} \theta_{k i_k d} \enspace .$$

Note that for binary relations, this is equivalent to a standard vectorial dot-product. In this paper, the distribution of the observations is assumed to be Gaussian with relation-dependent variances α_v^{-1}. This assumption can be relaxed easily to model other types of generalized linear models such as Poisson or Bernoulli distributions. Assuming independent observations, the likelihood of total observations $\mathcal{D}$ is given by

$$\begin{aligned} p(\mathcal{D}|\boldsymbol{\Theta}) &= \prod_{(v,\mathbf{i},r)\in\mathcal{D}} p(r|\theta_{S_{v1}i_1}, \dots, \theta_{S_{v|S_v|}i_{|S_v|}}, \alpha_v) \\ &= \prod_{(v,\mathbf{i},r)\in\mathcal{D}} \mathcal{N}(r|\sum_{d=1}^{D} \prod_{k\in S_v} \theta_{k i_k d}, \alpha_v^{-1}) \\ &= \prod_{(v,\mathbf{i},r)\in\mathcal{D}} e^{-\ell(\sum_{d=1}^{D} \prod_{k\in S_v} \theta_{k i_k d}, r; \alpha_v)}, \end{aligned}$$

where $\ell(\bar{r}, r; \alpha) = \frac{\alpha}{2}(r - \bar{r})^2 - \frac{1}{2}\log\frac{\alpha}{2\pi}$ is the quadratic loss.

We also assume that the prior distributions over $\Theta_1, \dots, \Theta_K$ are independent isotropic Gaussian distributions with type-dependent variances $\sigma_1^2, \dots, \sigma_K^2$:

$$p(\Theta_k|\sigma_k^2) = \prod_{j=1}^{N_k} \mathcal{N}(\theta_{kj}|0, \sigma_k^2 \mathbf{I}).$$

Now that we have presented the model, the remaining problem is to infer the latent variables $\boldsymbol{\Theta}$ given the observations. We consider the *Maximum a Posteriori* (MAP) estimator of $\boldsymbol{\Theta}$. The problem is therefore a simple minimization problem of a smooth and differentiable objective function equal to the negative log-likelihood:

$$\min_{\boldsymbol{\Theta}} \mathcal{O}, \quad \text{where} \quad \mathcal{O} := -\log p(\mathcal{D}|\boldsymbol{\Theta}, \boldsymbol{\alpha}) - \log p(\boldsymbol{\Theta}|\boldsymbol{\sigma}) \qquad (1)$$

and $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_V)$ and $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_K)$.

Two approaches to solve the optimization problem are *stochastic gradient descent* (SGD) and *alternating least squares* (ALS). ALS is a block-coordinate descent algorithm which minimizes Equation (1) with respect to one of the types, say Θ_k by fixing all others and repeats the same procedure for each Θ_k sequentially, ensuring that each step decreases the objective function. The procedure is repeated until convergence. The inner optimization problems are ordinary least squares which can be solved optimally. However, there is evidence from the tensor factorization literature that this procedure is not always effective because there are often strong dependencies between the feature values of the different types [25]. In addition, our method targets very large data sets for which even one pass through the data can be slow. This setting favors SGD-type algorithms since every gradient estimation is much cheaper than their batch counterpart (i.e., using standard unconstrained optimization tools such as L-BFGS [23]). This type of first-order optimization technique can be formally justified by a bias-variance argument, remarking that the ultimate goal of the estimation procedure is not the minimization of the objective (1), but the minimization of its expectation $\mathbb{E}\left[\mathcal{O}\right]$ under the sample distribution [7].

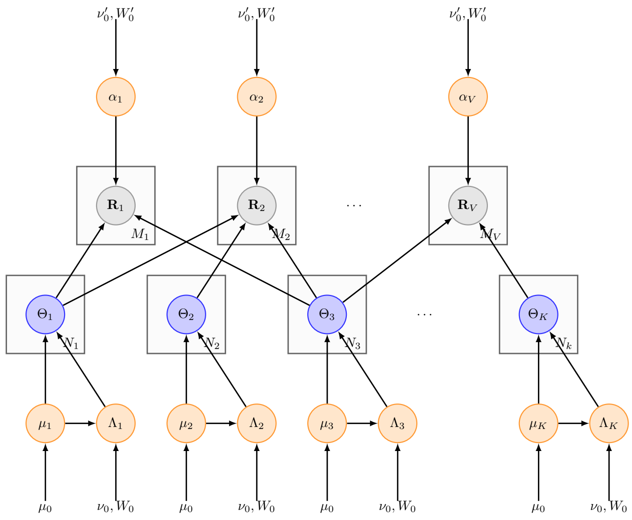


Figure 4: Sample Bayesian probabilistic multi-relational data factorization graphical model. $R_1, \ldots, R_V$ are the observed relations. $\Theta_1, \ldots, \Theta_K$ are the latent features associated to the K entity types. $\alpha_1, \ldots, \alpha_V$ are the unobserved precisions (inverse variances) associated with the observed relations, and similarly $\mu_1, \mu_2, \ldots, \mu_k$ and $\Lambda_1, \Lambda_2, \ldots, \Lambda_k$ are the unobserved mean and variances associated with latent features.

4.1 Bayesian Treatment

The performance of the probabilistic model is tied to the careful tuning of the hyper-parameter when model parameters Θ are estimated by maximum a posteriori probability (MAP) [29]. When the hyper-parameters are not properly tuned, such a point estimation—MAP—is often vulnerable to overfitting, especially when the data is large and sparse.

Instead of using MAP, an alternative estimation scheme that may avoid these problems is a fully Bayesian treatment, which integrates out all model parameters and hyper-parameters, arriving at a predictive distribution of future observations given observed data. Because this predictive distribution is obtained by averaging all models in the model space specified by the priors, it is less likely to over-fit a given set of observations.

A graphical overview of our entire model is in Figure 4, and each component is described below.

In this paper, we assume the observations follow a Gaussian distribution. As in the probabilistic model, this assumption can also be relaxed easily to model other types of generalized linear models such as Poisson or Bernoulli distributions. For each observation $(v, \mathbf{i}, r) \in \mathcal{D}$, we have

$$r|\Theta_{\mathbf{i}} \sim \mathcal{N}(\langle \mathbf{\Theta_i} \rangle, \alpha_v), \text{ where } (v, \mathbf{i}, r) \in \mathcal{D}$$

The prior distribution for hidden feature Θ is assumed to be Gaussian too, but the mean and the precision matrix (inverse of the covariance matrix) may take arbitrary value:

$$\theta_{kj} \sim \mathcal{N}(\mu_k, \Lambda_k^{-1}), \quad j = 1 \ldots N_k$$

The key ingredient of our fully Bayesian treatment is to view the hyper-parameter $\Phi_k \equiv \{\mu_k, \Lambda_k\}$ also as a random variable, leading to a predictive distribution for an unobserved rating $(v, \mathbf{i}, \hat{r})$

$$p(\hat{r}|\mathcal{D}) = \int\int p(\hat{r}|\Theta_{\mathbf{i}}, \alpha_v) p(\Theta_{\mathbf{i}}, \alpha, \Phi_{\mathbf{i}}|\mathcal{D}) d\{\Theta_{\mathbf{i}}, \alpha_v\}, d\{\Phi_{\mathbf{i}}\}$$

For convenience, we also define $\Phi_{\mathbf{i}} = \{\Phi_{i_1}, \ldots \Phi_{i_{|S_v|}}\}$. We then need to choose a prior distribution for the hyper-parameters. For the Gaussian parameter, we choose the conjugate distribution as priors that facilitate subsequent computation:

$$p(\alpha_v) = \mathcal{W}(\alpha_v|W_0', v_0')$$
$$p(\Phi_k) = p(\mu_k|\Lambda_k)p(\Lambda_k) = \mathcal{N}(\mu_0, (\beta_0\Lambda_k)^{-1})\mathcal{W}(\Lambda_k|W_0, v_0)$$

Here $\mathcal{W}$ is the Wishart distribution of a $D \times D$ random matrix Λ with v_0 degrees of freedom and a $D \times D$ scale W_0:

$$\mathcal{W}(\Lambda|W_0, v_0) = \frac{|\Lambda|^{(v_0 - D - 1)/2}}{C} \exp(-\frac{\mathrm{Tr}(W_0^{-1}\Lambda)}{2})$$

where C is a normalizing constant. There are several parameters in the hyper-priors: $\mu_0, \rho_0, \beta_0, W_0, v_0, W_0', v_0'$, which reflect our prior knowledge about the specific problem and can be treated as constants during training. In fact, Bayesian learning is able to adjust them according to the training data, and varying their values (within in a reasonably large range) has little impact on the final prediction, as is often observed in Bayesian estimation procedures [35].

5. INFERENCE

One can represent the predictive distribution of the relation value r given observation $(v, \mathbf{i}, r) \in \mathcal{D}$ by marginalizing over model parameters:

$$p(\hat{r}|\mathcal{D}) = \int\int p(\hat{r}|\Theta_{\mathbf{i}}, \alpha_v) p(\Theta_{\mathbf{i}}, \alpha, \Phi_{\mathbf{i}}|\mathcal{D}) d\{\Theta_{\mathbf{i}}, \alpha_v\}, d\{\Phi_{\mathbf{i}}\}$$

Often the exact predictive disribution is intractable; thus one relies on approximate inference such as sampling based on Markov chain Monte Carlo (MCMC) [21, 22]. For instance, MCMC can be used to approximate the predictive distribution of Eq. 2:

$$p(\hat{r}|\mathcal{D}) = \frac{1}{L}\sum_{l=1}^{L} p(\hat{r}|\Theta_{\mathbf{i}}^{(l)})$$

where the sample $\Theta_{\mathbf{i}}^{(l)}$ is generated by running a Markov chain whose stationary distribution is the posterior distribution over the model parameters and hyper-parameter Θ, Φ.

One of the simplest MCMC algorithms is Gibbs sampling [9], which cycles through the latent variables, sampling each one from the conditional distribution given the current values of all other variables. Gibbs sampling is typically used when these conditional distributions can be sampled from easily. In this section we give detailed derivation for the conditional distributions of model parameters and hyper-parameters which are required for implementing Gibbs sampling. Note that with our model assumptions, the joint posterior distribution can be factorized as

$$p(\Theta, \alpha, \Phi|\mathcal{D}) \propto \prod_{(v,\mathbf{i},r)\in\mathcal{D}} p(r|\theta_{S_{v1}i_1}, \ldots, \theta_{S_{v|S_v|}i_{|S_v|}}, \alpha_v)$$
$$\prod_k \left[p(\Theta_k|\Phi_k)p(\Phi_k)\right] \prod_v p(\alpha_v) \quad (2)$$

5.1 Hyper-parameters

We start with the derivation of the conditional distributions of the model hyper-parameters. For each v, α_v follows the Wishart distribution. By using the conjugate prior to α_v, we have the conditional distribution of α_v given R_v, Θ following the Wishart distribution:

$$p(\alpha_v|\mathcal{D}_v, \Theta) = \mathcal{W}(\alpha_v|W_0^*, v_0^*) \tag{3}$$

where

$$\begin{aligned} v_0^* &= v_0' + |\mathcal{D}_v|, \\ (W_0^*)^{-1} &= W_0'^{-1} + \sum_{(v,\mathbf{i},r)\in\mathcal{D}_v} (r - \langle\mathbf{\Theta_i}\rangle)^2. \end{aligned}$$

Next, we derive the conditional probability for Φ_k. Our graphical model (Fig. 4) assumption suggests that it is conditionally independent of all the other parameters given Θ_k. We thus integrate out all the random variables in Eq. 2 except Θ_k, and obtain the Gaussian-Wishart distribution:

$$p(\Phi_k|\Theta_k) = N(\mu_k|\mu_0^*, (\beta_0^*\Lambda_k)^{-1})\mathcal{W}(\Lambda_k|W_0^*, v_0^*), \tag{4}$$

where

$$\mu_0^* = \frac{\beta_0\mu_0 + N_k\bar{\theta}_k}{\beta_0 + N_k}, \; \beta_0^* = \beta_0 + N_k, \; v_0^* = v_0 + N_k;$$

$$(W_0^*)^{-1} = W_0^{-1} + N_k\bar{S} + \frac{\beta_0 N_k}{\beta_0 + N_k}(\mu_0 - \bar{\theta}_k)(\mu_0 - \bar{\theta}_k)^T,$$

$$\bar{\theta}_k = \frac{1}{N_k}\sum_{j=1}^{N_k}\theta_{kj}, \; \bar{S} = \frac{1}{N_k}\sum_{j=1}^{N_k}(\theta_{kj} - \bar{\theta}_k)(\theta_{kj} - \bar{\theta}_k)^T.$$

5.2 Model-parameters

The remaining conditional distributions are for model parameters Θ_k, and we describe the derivation of these distributions in this section. According to the graphical model (Fig. 4), its conditional distribution factorizes with respect to the individual entities:

$$p(\Theta_k|\mathcal{D}, \Theta_{-k}, \alpha, \Phi_k) = \prod_{j=1}^{N_k} p(\theta_{kj}|\mathcal{D}, \Theta_{-k}, \alpha, \Phi_k)$$

$$p(\theta_{kj}|\mathcal{D}, \Theta_{-k}, \alpha, \Phi_k) = \mathcal{N}(\theta_{kj}|\mu_{kj}^*, (\Lambda_{kj}^*)^{-1}) \tag{5}$$

where

$$\begin{aligned} \mu_{kj}^* &= (\Lambda_{kj}^*)^{-1}(\Lambda_k\mu_k + \sum_{v\in\{v'|k\in S_{v'}\}} \alpha_v \sum_{(v,\mathbf{i},r)\in\mathcal{D}_v, kj\in\mathbf{i}} rQ_{(v,\mathbf{i},r)}) \\ \Lambda_{kj}^* &= \Lambda_k + \sum_{v\in\{v'|k\in S_{v'}\}} \alpha_v \sum_{(v,\mathbf{i},r)\in\mathcal{D}_v, kj\in\mathbf{i}} Q_{(v,\mathbf{i},r)}Q_{(v,\mathbf{i},r)}^T \\ Q_{(v,\mathbf{i},r)} &= \frac{\prod_{n=1}^{|S_v|}\theta_{S_{v,n},i_n}}{\theta_{kj}} \end{aligned}$$

6. EXPERIMENTS

We conduct systematic experiments to evaluate the two versions of our proposed model, named PRA (Probabilistic Relational-data Analysis) and BPRA (Bayesian Probabilistic Relational-data Analysis) on two data sets: Flickr and Bibsonomy.[5]

Algorithm 1 Gibbs sampler for Relational Data Analysis

```
INPUT: hyper-prior parameters $\{\mu_0, \rho_0, \beta_0, W_0, v_0, W_0', v_0'\}$
OUTPUT: model parameters $\{\Theta\}$
Initialize model parameters $\{\Theta^{(1)}\}$
for $l = 1, ..., L,$ do
    Sample the hyper-parameters according to Eq. {3, 4 }, respectively:
        $\alpha_v^{(l)} \sim p(\alpha_v|\mathcal{D}, \Theta^{(1)})$ where $v = 1, \ldots, V$
        $\Phi_k^{(l)} \sim p(\Phi_k|\mathcal{D}, \Theta_k^{(1)})$ where $k = 1, \ldots, K$
    Sample the model parameters in parallel
    according to Eq. {5}:
    for $k = 1, \ldots, K$ do
        for each latent factor
            $\theta_{kj}^{(l+1)} \sim p(\theta_{kj}|\mathcal{D}, \Theta_{1:k-1}^{(l+1)}, \Theta_{k+1:K}^{(l)}, \alpha^{(l)}, \Phi^{(l)})$
                where $j = 1, \ldots, N_k$and $k \neq t$
    end for
end for
```

6.1 Evaluation and Comparison Methods

As there are different kinds of responses (such as binary, term frequency and real value) in our recommendation tasks across multi-contexts, we employ Root Mean Square Error (RMSE) as our primary measurement for all contexts. In our Bayesian probabilistic relational-data model, we simply set $\mu_0, \rho_0, \beta_0, W_0, v_0, W_0', v_0'$ all equal to one or identity vector and $D = 20$ for the dimension of latent factors, on all three data sets. Our experiments also show that the performance is fairly robust to changes to the hyper-prior.

In the following experiments, we compare our methods with four state-of-the-art latent factor methods:

- Salakhutdinov's Probabilistic Matrix Factorization (PMF) [30]: collaborative filtering using probabilistic matrix factorization which treats activities as independent.

- Bayesian Probabilistic Matrix Factorization (BPMF) proposed by Salakhutdinov et al. [29]: the Bayesian version of PMF.

- Rendle's Tensor Factorization (TF) [27, 28] which handles high-order relational data for tag prediction and showed prior success in the graph-based tag recommendation task.

- Bayesian Probabilistic Tensor Factorization (BPTF) proposed by Xiong et al. [35] which models temporal collaborative filtering, and whose extension is straightforward to model higher order relational data such as user-tag-comments.

6.2 Flickr Experiments

6.2.1 *Data set*

The Flickr data has been briefly described in Section 3. This data set includes 2,866 users, 60,339 tags, 32,752 comment terms and 46,733 items (e.g., images), leading to four relations. The relation $S_1 = (u, t, i)$ indicates that user u tags item i with tag t. The relation $S_2 = (i, f)$ characterizes item i with a 1024-dimension feature vector f extracted

[5] To facilitate replication of experiments, sourcecode and datasets are available upon request.

according to [24], which are of real numbers. The relation $S_3 = (u_1, u_2)$ encodes a partially observed adjacency matrix representing the explicitly expressed friendship relations among users. For instance, if user u_1 and u_2 are friends, then the value at (u_1, u_2) and (u_2, u_1) are both equal to 1, 0 otherwise. The relation $S_4 = (u, c, i)$ indicates that user u comments on item i using word c, and this relation can be described by term frequency (positive integers).

In the first relation, the problem of interest is tag prediction, that is, to predict tags that users will assign to items. We need to model relation S_1, for which the Flickr data set has a total of 373,125 records with time stamps. The data is partitioned into training and test sets based on the time stamp of April 1st 2010. In total, there are 2,613,388 observations for training and 205,880 observations for testing. Note that there are only positive samples of tags available for the Flickr data set, so we sample 50 tags at random as negative examples for training. For the relation user-comment-item, where users could make some comments on a specific item, we try to predict the term frequency in the comments and the data also are split into training and test data set similarly, resulting in 1,366,068 training observations and 341,043 testing observations.

As mentioned above, we also have two more contexts: for image content, we characterize image i by a feature vector f of 1024-dimensional visual features according to Perronnin and Dance [24]; the social context is also comprised of binary typed observations, which contain 1,377,548 training observations and 342,576 test observations.

6.2.2 *Analysis of relations and their co-effects*

Some explanatory analysis has been presented in Section 3. A social tagging system is a coupled higher-order data system and multiple contexts are coupled together. Here, we will show that by using our methods together with Bayesian treatment, predictive accuracy can be mutually improved.

We first conduct two versions of our model: PRA for MAP version and BPRA for Bayesian version. In Table 3, it can be seen that the Bayesian method clearly outperforms the MAP version (in all scenarios), due to the high data sparsity. In Figure 5(a), we show the convergence of our Bayesian model BPRA which starts sampling with parameters based on the results of PRA. We can see that after around 50 epochs, the performance on two relations converge. In the following sections, we will use Bayesian version for analysis and comparison.

Another interesting question is: do coupled relations lead to mutually improved prediction performance? We conduct experiments on modeling different relations with several combinations to study this question. The tasks are described in Section 6.2.1 for different relations and the results are shown in Table 3. The first four rows of the table indicate that best performances are achieved for all four relations when modeling them together. The following three rows (rows 5-7) of the table indicate the performance of modeling three relations (C1, C2, C4). Similarly, the results of modeling (C1, C3, C4) and (C1, C4) are shown in the remaining rows. Taking the prediction of Context 1 (C1: user-tag-item) relation as an example: the best performance is 0.3073 in modeling all four relations, 0.3177 in modeling the three relations (C1, C3, C4), and degrades to 0.3465 when only modeling the relation (C1, C4) together. Comparable results for comment prediction are also shown in Figure 5(c).

6.2.3 *Comparison with existing methods*

We report the evaluation of our models together with comparisons to state-of-the-art approaches introduced earlier. Bayesian Probabilistic Matrix Factorization and its Bayesian treatment are popular methods and have shown success in traditional collaborative filtering. In our experiments with binary contexts, we compare our methods with PMF and BPMF. Since TF and BPTF can model the tag prediction and comment prediction tasks, we compare our methods with them in such higher-order contexts.

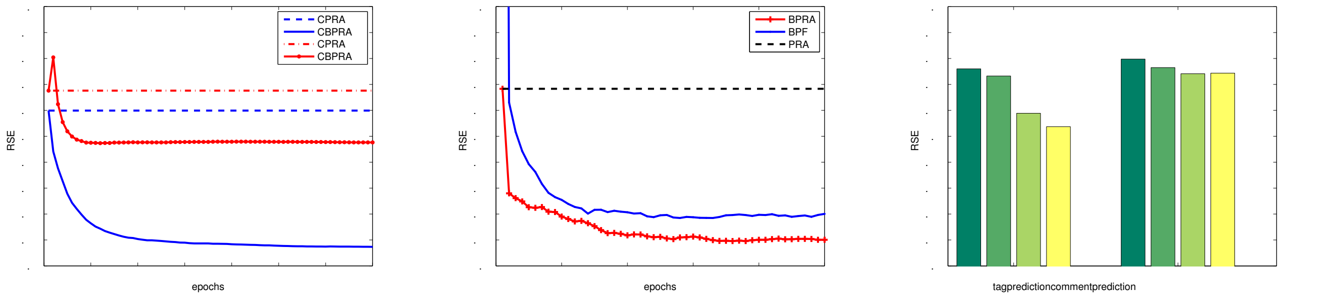
We summarize the results in Table 3. While Section 3 showed that over 90% of real-world cases are cold start problems and the graph-based methods (such as Hotho's Folkrank and Rendle's tensor factorization) will not work on such cases, we still compare to the state of the art method—tensor factorization. The results show that Rendle's TF performs the worst in tag prediction, because it only models a single relation without encoding external information of items. Intuitively the external information of items (e.g., comments, features) is more critical to the tag prediction task. This result agrees with [38]. For the cold start problem, the external information of items is essential for tag prediction because most items do not exist in the training data.

In the comment prediction context, we see similar results; tensor factorization performs the worst because of the lack of external information and data sparsity. Xiong's method—Bayesian Tensor Factorization—is much better, but our methods still achieve the best performance. In both tag and comment prediction, the experiments show that in such a real-world case, tensor factorization is insufficient and Bayesian treatment on tensor factorization can improve performance significantly because of the data sparsity. We also note that with more information in the model, the performance of our approach improves, e.g., with social relation information (C3), we can see that both tag and comment prediction improves.

Overall, we can see that for all methods, the Bayesian

Table 3: RMSE of 4 relations for Flickr data set. Context 1 for users tagging items (user-tag-item), Context 2 for item content (item-feature), Context 3 for social interaction (user-user) and Context 4 for users' comments on item (user-item-comments)

	BPRA	PRA	PMF	BPMF	TF	BPTF
C1	**0.3073**	0.3599	N/A	N/A	0.8226	0.3520
C2	0.9215	0.9627	0.9913	**0.9004**	N/A	N/A
C3	**0.1828**	0.2053	0.1841	0.1878	N/A	N/A
C4	**0.3478**	0.3676	N/A	N/A	0.4185	0.3593
C1	0.3449	0.4450	N/A	N/A	0.8226	0.3520
C2	0.9198	0.9630	0.9913	0.9004	N/A	N/A
C4	0.3516	0.3681	N/A	N/A	0.4185	0.3593
C1	0.3177	0.3984	N/A	N/A	0.8226	0.3520
C3	0.1858	0.2298	0.1841	0.1878	N/A	N/A
C4	0.3482	0.4241	N/A	N/A	0.4185	0.3593
C1	0.3465	0.7843	N/A	N/A	0.8226	0.3520
C4	0.3530	0.3656	N/A	N/A	0.4185	0.3593



(a) BPRA vs. PRA on Flickr

(b) Experiments on Bibsonomy

(c) RMSE decreasing with more contexts on Flickr

Figure 5: Experimental results for different activities exhibited in the two data sets.

versions always outperform the MAP version respectively, due to the sparsity of the data. Our model outperforms all four recent nontrivial methods—PMF, TF, BPMF, BPTF in the comments context, social network context and tag context. We also notice that in the item feature relation, our model is slightly worse than BPMF. That is because our model tries to maximize the total likelihood for all relations.

Table 4: RMSE on the Bibsonomy data set. Context 1 for users tagging items (user-tag-item) and Context 2 for item content (item-feature)

	BPRA	PRA	PMF	BPMF	TF	BPTF
C1	**.3097**	0.3484	N/A	N/A	1.0409	0.3455
C2	**1.0088**	1.0118	1.7387	1.1025	N/A	N/A

6.3 Bibsonomy Experiments

The second data set used to evaluate our model is Bibsonomy—the bookmark data set of the ECML-PKDD'09 Discovery Challenge. This data set involves 2,679 users, 263,004 items, 56,424 tags, 262,336 posts and 1,401,104 records. Clearly, this is also a very sparse data set, whose density is only 3.52×10^{-8}. Each post is associated with a time stamp, and each item contains textual content. In this experiment, we show that the single graph-based model cannot work in the real world (where the data set is split by time stamp). By incorporating content into the model, prediction accuracy can be significantly improved. To generate a descriptor for the items, we first use the bag-of-words language model and then use Latent Dirichlet Allocation [6] to produce a latent factor for each item. There are only two relations for this data set: $S_1 = (u, t, i)$, where user u tag item i with tag t, and $S_2 = (i, f)$, where each item i is described by a 100-dimensional feature f. To model S_1, we use a time stamp of August 1st 2008 to distinguish training and testing sets with 7,214,426 and 1,585,179 observations respectively.

We show the results for Bibsonomy in Table 4. At first, we compare the two versions of our model: BPRA is still clearly much better than PRA, benefiting from handling sparse data well. Similarly, in Figure 5(b), we show the convergence of our Bayesian model BPRA which starts sampling with parameters based on the results of PRA. We can see that after around 50 epochs, performance converges. The convergence in Bibsonomy experiments is consistent with our Flickr experiments. We also compare our methods with the baselines. Similarly, BPMF and BPTF outperform PMF and TF respectively. The experiments on this data set also verify the necessity of employing Bayesian treatment in social relational data.

TF almost fails to solve the task specified by (user-tag-item) relation without item external information, because as we have shown in Section 3, most items in a tagging log are new items. The results of our model are consistent with the Flickr data: our model noticeably decreases the RMSE for the tag prediction task. The performance for both relations can lead to significant improvements: 0.3097 in the (user-tag-item) relation and 1.0088 in the (item-feature) relation respectively. This also confirms that the two contexts can mutually reinforce the performance of the model. Overall, like in Flickr experiments, our Bayesian model noticeably outperforms all other methods in the Bibsonomy data set.

7. CONCLUSION AND FUTURE WORK

In this paper, we examined how to model predictive social tagging systems. We found that user activity modeling in social tagging systems suffers from coupled high order interaction, data sparsity, and the cold start problem. We tackled these problems with a novel generalized latent factor model and Bayesian treatment. We found that in social tagging systems, the user-comment-item and user-tag-item relations can be mutually inferred based on common latent factors and thus improve prediction performance, which has not been explored previously.

Our novel latent factor model can handle multiple activities, such as commenting within tagging systems and can do so simultaneously and demonstrate predictive superiority over state-of-the-art methods. Our experiments on two real-world data sets also show the advantage of employing a fully Bayesian treatment to boost the performance of point estimation when modeling high order relations.

There are many possible extensions to the current approach, either in terms of scalability or in terms of modeling. A first direction is to investigate how to incorporate temporal factors into the model. Temporal factors have been shown to be important by a number of previous efforts[37, 17, 35]. A significant improvement is expected through incorporating temporal factors.

Secondly, in our experiments, we found that different contexts, e.g., tag context and comments context may have different convergence speeds. One possible solution is that one could add a *core tensor* in each of the factored matrices and tensors, as is done in Tucker decomposition for tensors [34]. It enables a more flexible parameterization of the problem thanks to the use of relation-specific core tensors.[6] It will also enable entity specific latent dimensions $D_1, \cdots, D_k$ instead of the constant dimension D used for all the entities.

While the proposed algorithm can scale to hundreds of thousands of observations, it requires several hours to converge. A possible solution is to utilize deterministic approximate inference techniques such as variational Bayes to further improve the convergence speed and enable the possibility of using gradient descent algorithms instead of Gibbs sampling.

Another potentially interesting follow-up of this work is the study of very large data sets, for which distributed algorithms can be designed thanks to the decomposable formulation of the loss. The on-the-fly computation of predictions might be also of interest in order to obtain near real-time responses.

8. REFERENCES

[1] E. Acar, T. G. Kolda, and D. M. Dunlavy. All-at-once optimization for coupled matrix and tensor factorizations. In *MLG*, 2011.

[2] R. Adams, G. Dahl, and I. Murray. Incorporating side information in probabilistic matrix factorization with gaussian processes. In *UAI*, pages 1–9, 2010.

[3] D. Agarwal and B.-C. Chen. Regression-based latent factor models. In *SIGKDD*, pages 19–28, 2009.

[4] D. Agarwal, B.-C. Chen, and B. Pang. Personalized recommendation of user comments via factor models. In *EMNLP*, pages 571–582, 2011.

[5] S. Bao, G. Xue, X. Wu, Y. Yu, B. Fei, and Z. Su. Optimizing web search using social annotations. In *WWW*, 2007.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[7] L. Bottou and O. Bousquet. The trade-offs of large scale learning. In *NIPS*, volume 20, pages 1161–168, 2008.

[8] X. Ding, B. Liu, and P. S. Yu. A holistic lexicon-based approach to opinion mining. In *WSDM*, pages 231–240, 2008.

[9] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE TPAMI*, 6(6):721–741, November 1984.

[10] Z. Guan, J. Bu, Q. Mei, C. Chen, and C. Wang. Personalized tag recommendation using graph-based ranking on multi-type interrelated objects. In *SIGIR*, pages 540–547, 2009.

[11] S. K. Gupta, D. Phung, B. Adams, T. Tran, and S. Venkatesh. Nonnegative shared subspace learning and its application to social media retrieval. In *SIGKDD*, 2010.

[12] P. Heymann, D. Ramage, and H. Garcia-Molina. Social tag prediction. In *SIGIR*, pages 531–538, 2008.

[13] T. Hofmann. Probabilistic latent semantic analysis. In *UAI*, pages 289–296, 1999.

[14] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. In *The Semantic Web: Research and Applications*, Lecture Notes in Computer Science, chapter 31, pages 411–426. 2006.

[15] M. Hu and B. Liu. Mining and summarizing customer reviews. In *SIGKDD*, pages 168–177, 2004.

[16] R. Jäschke, L. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme. Tag recommendations in folksonomies. In *PKDD*, pages 506–514, Berlin, Heidelberg, 2007.

[17] Y. Koren. Collaborative filtering with temporal dynamics. In *KDD*, pages 447–456, 2009.

[18] M. Lipczak, Y. Hu, Y. Kollet, and E. Milios. Tag sources for recommendation in collaborative tagging systems. In *ECML/PKDD Discovery Challenge Workshop (DC09)*, 2009.

[19] B. Liu, M. Hu, and J. Cheng. Opinion observer: analyzing and comparing opinions on the web. In *WWW*, 2005.

[20] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *WSDM*, 2011.

[21] N. Metropolis and S. Ulam. The Monte Carlo methods. *JASA*, 44(247):335–341, 1949.

[22] R. M. Neal. Probabilistic inference using Markov Chain Monte Carlo methods, 1993.

[23] J. Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of Computation*, 35, 1980.

[24] F. Perronnin and C. R. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, 2007.

[25] M. Rajih, P. Comon, and R. A. Harshman. Enhanced line search: A novel method to accelerate PARAFAC. *SIAM J. Matrix Anal. Appl.*, 30:1128–1147, 2008.

[26] D. Ramage, P. Heymann, C. D. Manning, and H. Garcia-Molina. Clustering the tagged web. In *WSDM*, 2009.

[27] S. Rendle, L. Balby Marinho, A. Nanopoulos, and L. Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In *SIGKDD*, 2009.

[28] S. Rendle and L. Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *WSDM*, pages 81–90, 2010.

[29] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *ICML*, pages 880–887, 2008.

[30] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *NIPS*, pages 1257–1264, 2008.

[31] A. P. Singh and G. J. Gordon. Relational learning via collective matrix factorization. In *SIGKDD*, pages 650–658, 2008.

[32] Y. Song, L. Zhang, and C. L. Giles. A sparse gaussian processes classification framework for fast tag suggestions. In *CIKM*, pages 93–102. ACM Press, 2008.

[33] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos. Tag recommendations based on tensor dimensionality reduction. In *RecSys*, pages 43–50. ACM Press, 2008.

[34] L. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31:279–311, 1966.

[35] L. Xiong, X. Chen, T.-K. Huang, J. Schneider, and J. G. Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *SDM*, 2010.

[36] S. Xu, S. Bao, B. Fei, Z. Su, and Y. Yu. Exploring folksonomy for personalized search. In *SIGIR*, pages 155–162, 2008.

[37] D. Yin, L. Hong, Z. Xue, and B. D. Davison. Temporal dynamics of user interests in tagging systems. In *AAAI*, pages 1279–1285, 2011.

[38] D. Yin, Z. Xue, L. Hong, and B. D. Davison. A probabilistic model for personalized tag prediction. In *KDD 2010*.

[39] Z. Yin, R. Li, Q. Mei, and J. Han. Exploring social tagging graph for web object classification. In *SIGKDD*, 2009.

[40] J. Yoo and S. Choi. Bayesian matrix co-factorization: Variational algorithm and cramer-rao bound. In *ECML/PKDD*, 2011.

[41] Y. Zhang, B. Cao, and D.-Y. Yeung. Multi-domain collaborative filtering. In *UAI*, 2010.

[42] S. Zhu, K. Yu, Y. Chi, and Y. Gong. Combining content and link for classification using matrix factorization. In *SIGIR*, pages 487–494, 2007.

[6] Tucker decomposition is a strict generalization of the Parafac decomposition