Transfer Learning for Collective Link Prediction in Multiple Heterogenous Domains

Bin Cao CAOBIN@CSE.UST.HK
Nathan Nan Liu NLIU@CSE.UST.HK
Qiang Yang QYANG@CSE.UST.HK
Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong

Abstract

Link prediction is a key technique in many applications such as recommender systems, where potential links between users and items need to be predicted. A challenge in link prediction is the data sparsity problem. In this paper, we address this problem by jointly considering multiple heterogeneous link prediction tasks such as predicting links between users and different types of items including books, movies and songs, which we refer to as the *collective link prediction* (CLP) problem. We propose a nonparametric Bayesian framework for solving the CLP problem, which allows knowledge to be adaptively transferred across heterogeneous tasks while taking into account the similarities between tasks. We learn the inter-task similarity automatically. We also introduce link functions for different tasks to correct their biases and skewness of distributions in their link data. We conduct experiments on several real world datasets and demonstrate significant improvements over several existing state-of-the-art methods.

1. Introduction

Relational data modeling has been attracting growing interests in recent years and has found successful applications in many areas such as social network analysis, computational biology and recommender systems. In this paper, we focus on a particular task for relational data modeling: link prediction, which is concerned with predicting whether two entities have certain relations. Many important applications can be cast as link prediction problems. For example, personalized recommendation and targeted advertising involve predicting the potential links between users and products/advertisements based on observed links in the form of users' past purchases or clicks. A major difficulty faced by many real-world link prediction tasks is the data sparsity problem, which happens when many items do not have links between them. For example, in recommender system applications, a majority of users only rate very a few items. As a result, there exists a large number of items in the long tail that are only rated a few times. This sparsity problem can be even more severe for new users and new items, creating a problem that is also known as the cold-start problem. In this paper, we focus on how to solve the data-sparsity problem by considering a *collective link prediction* (CLP) formulation, which jointly models a collection of link prediction tasks arising from multiple heterogeneous domains. CLP is particularly suitable for large-scale e-commerce and social networking services, which often provide a diverse range of products or services, and different product/service categories such as books, clothes, electronics naturally constitute different domains. By exploring the correlation between link prediction tasks in different domains, we can transfer the shared knowledge among similar tasks to alleviate the data sparsity problem associated with individual tasks and therefore improving the performances of all tasks.

We propose a nonparametric Bayesian framework for collective link prediction by developing a multi-task extension of the Gaussian-process latent-variable model (Lawrence, 2003). Our CLP model addresses two major challenges that are not considered by previous work on link prediction. The first challenge is the different degrees of relatedness between heteroge-

Appearing in *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 2010. Copyright 2010 by the author(s)/owner(s).

neous task domains, from which we wish to transfer the knowledge to the target learning domains. For example, the task of predicting user preferences on books should be more related to predicting user preferences on movies than user preferences on food. Since not all tasks are equally correlated, we should consider the task similarities in their joint model. Towards this end, we incorporate a task similarity kernel in the model, which is automatically learned from the data to adaptively transfer knowledge between heterogeneous tasks. The second challenge is caused by the skewed distributions of most real-world link data. For example, when the link data consist of user ratings on items, the amount of positive ratings often significantly exceeds the amount of negative ratings, since users are reluctant to rate items that they do not like. For Gaussian process models, such imbalanced distributions violate the assumption of data distribution. To correct the bias and skewness of the distributions of link data and handle heterogeneous link types, we further introduce a specific family of link functions for the collective link prediction. We conduct experiments on several representative real-world datasets from multiple heterogeneous domains. We demonstrate the effectiveness of our proposed mode in several experiments

2. Related Work

Link prediction is an important task of relational data modeling (Getoor & Diehl, 2005). Early link prediction models are entirely based on structural properties of the observed network. (Liben-Nowell & Kleinberg, 2003) compares many predictors based on different graph proximity measures. Within the machine learning community, the link prediction problem has received more and more attention in recent years (Yu & Chu, 2007; Salakhutdinov & Mnih, 2007) and nonparametric Bayesian models are becoming a popular tool for such problems. (Yu et al., 2006) proposed the stochastic relational models for the link prediction problem, which are essentially the Gaussian process models. They further extend this work to more general problems in (Yu & Chu, 2007; Yu et al., 2009). (Salakhutdinov & Mnih, 2007; 2008) extend the matrix factorization model to the probabilistic framework for collaborative filtering tasks. (Lawrence & Urtasun, 2009) discussed the relation between the probabilistic matrix factorization (PMF) model and the Gaussian process models and proposed a nonlinear extension to the PMF model, which is equivalent to the GP-LVM model (Lawrence, 2003). However, all the related work above in link prediction literature focus only on the single link prediction task.

Our work is related to the multi-relational learning problem, where several relations are jointly modeled. Most methods view the given relations as a collection of matrices, where an entity may correspond to a row or column in multiple matrices. Different strategies have been developed to enable parameter sharing when jointly factorize a collection of related matrices so that knowledge can be transferred across different tasks. For example, in (Singh & Gordon, 2008; Zhu et al., 2007), an entity is required to be represented by the same latent features in different matrices. Xu et al. in (Xu et al., 2009) extended such collective matrix factorization models to a nonparametric Bayesian framework based on Gaussian processes. The CLP problem we consider in this paper can be solved via these multi-relational learning models by simply treating each domain as a different type of relation. However, a crucial difference between our approach and previous multi-relational learning work is that we place special care in accommodating different degrees of relatedness between different domains whereas previous models all assume the behaviors of the same entities in different domains are always consistent. Although the problem is also proposed in (Berkovsky et al., 2007), no effective solution has been proposed.

3. Collective Link Prediction

In this section, we discuss our collective link prediction model in detail. We first introduce the probabilistic nonlinear matrix factorization model for single link prediction task. Then we extend it to collective link prediction tasks.

3.1. Notations and Problem Definition

We first define the notations used in the paper. We use $\mathbb{R}^{m,n}$ to denote the space of matrices of m rows and n columns. The ij-element of a matrix $\mathbf{X}$ is denoted by $\mathbf{x}_{i,j}$. The i^{th} row of the matrix $\mathbf{X}$ is denoted by $\mathbf{x}_{i,:}$ and the j^{th} column of the matrix $\mathbf{X}$ is denoted by $\mathbf{x}_{:,j}$. $\mathbf{X}^T$ is the transpose of the matrix $\mathbf{X}$. For a vector $\mathbf{x}$, the notation of $\frac{1}{\mathbf{x}}$ means the entry-wise divide and still represents a vector.

For a link prediction task, a set of observed link data are given, which form a sparse matrix $\mathbf{X}$ with missing entries. We aim at predicting the missing entries in the matrix. For the collective link prediction tasks, we are given a collection of matrices $\{\mathbf{X}^{(t)}\}, t = 1 \cdots, T$, where T is the number of tasks. Our objective is to predict the missing values for all the matrices by considering all the data available.

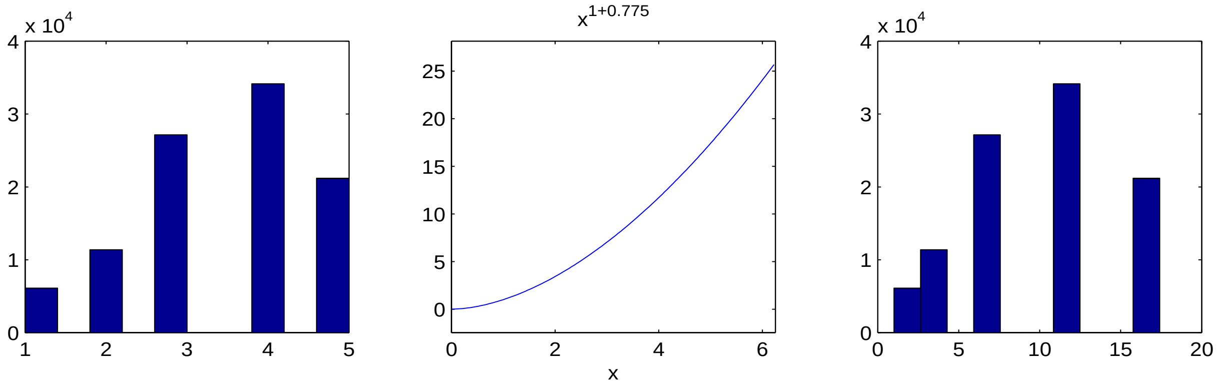

Figure 1. A skewness example on MovieLens. The skewness of the distribution of original data is -0.51. The skewness of the distribution of transformed data is -1.8×10^{-4}. $\alpha = 0.775$ for this dataset.

3.2. Link Modeling via Nonlinear Matrix Factorization

Considering a link matrix $\mathbf{X}$, its ij-element can be modeled by,

$$\mathbf{x}_{i,j} \sim f(\mathbf{u}_i, \mathbf{v}_j, \varepsilon) \tag{1}$$

where f is a link function, $\mathbf{u}_i$ and $\mathbf{v}_j$ are the latent feature representations of two entities involved and ε is a noise term. A linear model in a matrix form would introduce the generalized matrix approximation model

$$\mathbf{X} \sim f(\mathbf{U}\mathbf{V}^{\mathrm{T}} + \mathbf{E}). \tag{2}$$

The matrices $\mathbf{U}$ and $\mathbf{V}$ are the low dimensional feature representations of the two types of entities involved and $\mathbf{E}$ is the matrix form of noise term. By assuming the noise be Gaussian, the latent variable $f^{-1}(\mathbf{X})$, which we denote by $\mathbf{Y}$, would follow a multivariate Gaussian distribution

$$p(\mathbf{Y}|\mathbf{U}, \mathbf{V}, \sigma^2) = \prod_i \mathcal{N}(\mathbf{y}_{i,:}|\mathbf{u}_{i,:}\mathbf{V}^{\mathrm{T}}, \sigma^2\mathbf{I}). \tag{3}$$

We can further model $\mathbf{U}$ with a Gaussian prior,

$$p(\mathbf{U}) = \prod_i \prod_j \mathcal{N}(\mathbf{u}_{i,j}|0, \beta_u^{-1}) \tag{4}$$

and marginalize it leads to

$$p(\mathbf{Y}|\mathbf{V}, \sigma^2, \beta_u) = \prod_i \mathcal{N}(y_{i,:}|\mathbf{0}, \beta_u^{-1}\mathbf{V}^{\mathrm{T}}\mathbf{V} + \sigma^2\mathbf{I}), \tag{5}$$

which is similar to the probabilistic PCA model (Tipping & Bishop, 1999), despite that the observation are transformed by f^{-1}. The covariance matrix $\mathbf{V}^{\mathrm{T}}\mathbf{V}$ can be seen as a kernel function. Therefore, using the kernel trick, we can obtain the nonlinear matrix factorization model (Lawrence & Urtasun, 2009) for link prediction,

$$p(\mathbf{Y}|\mathbf{V}, \sigma^2, \beta_u) = \prod_i \mathcal{N}(\mathbf{y}_{i,:}|\mathbf{0}, \mathbf{K} + \sigma^2\mathbf{I}) \tag{6}$$

where $\mathbf{K}_{i,j} = k(\mathbf{v}_{:,i}, \mathbf{v}_{:,j})$.

With the link function, we further consider the distribution with respect to the observed link data $\mathbf{X}$, which can be formulated in a warped GP model (Snelson et al., 2003). If we choose the link function to be investable and differentiable, then we can obtain the distribution for $\mathbf{X}$,

$$p(\mathbf{X}|\mathbf{V}, \sigma^2, \beta_u) = \prod_i \mathcal{N}(g(\mathbf{x}_{i,:})|\mathbf{0}, \mathbf{K} + \sigma^2\mathbf{I}) \cdot |g'(\mathbf{x}_{i,:})| \tag{7}$$

where we let $g = f^{-1}$ and $g'(\mathbf{x}_{i,:}) = \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}}|_{\mathbf{x}_{i,:}}$.

The nonlinear matrix factorization model (Lawrence & Urtasun, 2009) does not consider the link function. However, the introduction of link function is important in link prediction tasks. We will explore the link function in the next section.

3.2.1. Link Function

Gaussian process models assume that the observation data follow Gaussian distributions. However, real world data may not satisfy this assumption. For example, in recommender systems, users are reluctant to rate items they do not like. In fact, the distribution of ratings is negative skewed, as shown in Figure 1, where the skewness of a distribution for a random variable Z is defined by

$$\gamma = \frac{\mathrm{E}[(Z-\mu)^3]}{\mathrm{E}[(Z-\mu)^2]^{3/2}} \tag{8}$$

where μ is the expectation of Z. The Gaussian distribution should have zero skewness. However, the distribution of ratings on the MovieLens dataset has skewness -0.51 and therefore does not fit in a normal distribution well, which violates the underlying assumption made by GP models. Introducing of the link function can help adjust the distributions to be more suitable for GP models.

We directly consider the inverse link function g instead of link function f in order to simplify the model learning. Since distributions of the rating data are negative skewed, we consider the following form of link function to correct the skewness,

$$g(x) = x^{1+\alpha}, \quad \alpha > 0 \tag{9}$$

Figure 1 also shows the link function fitted in the MovieLens dataset[1] and the rating distribution after correction. The skewness can drop from -0.51 to -1.8×10^{-4}. Therefore, we can adjust the skewness as preprocessing. However, a more principle method is to learn the parameters in the link function together with the link prediction model as we show later.

3.3. Collective Link Modeling

In this section, we present a solution to the problem of collective link prediction tasks, where multiple entities and relationships are involved. In particular, we consider the collective link prediction problem where one entity type $\mathbf{U}$ can be used together with several other entity types $\mathbf{V}^{(t)}(t = 1 \cdots T)$. As a result, we can learn for T link-prediction tasks in total.

For each link prediction task t, we obtain a GP model,

$$p(\mathbf{Y}^{(t)}|\mathbf{V}^{(t)}, \sigma^2, \beta_u) = \prod_i \mathcal{N}(\mathbf{y}_{i,:}|\mathbf{0}, \mathbf{K}^{(t)} + \sigma^2\mathbf{I}^{(t)}). \tag{10}$$

These link prediction tasks may not be independent with each other. Therefore, we consider a joint prediction model where we have

$$p(\{\mathbf{Y}^{(t)}\}|\mathbf{V}, \sigma^2, \beta_u) = \prod_i \mathcal{N}(\{\mathbf{y}_{i,:}\}|\mathbf{0}, \mathbf{C}) \tag{11}$$

where $\{\mathbf{Y}^{(t)}\} = (\mathbf{Y}^{(1)}, \mathbf{Y}^{(2)}, \ldots, \mathbf{Y}^{(T)})$ is the joint vector of link values and $\mathbf{C}$ is the joint task kernel to model the cross task-link similarities.

The joint task kernel can be defined by,

$$\mathbf{C} = \mathbf{T} \otimes \mathbf{K} + \sigma^2\mathbf{I}, \tag{12}$$

where $\mathbf{T}$ is a positive semi-definite (PSD) matrix that specifies the inter-task similarities and $\mathbf{K}$ is the kernel matrix for link modeling using kernel function k. For such a joint task kernel, the inner product between two entities $\mathbf{v}$ from task s and $\mathbf{v}'$ from task t is,

$$< \mathbf{v}, \mathbf{v}' >= \mathbf{T}_{s,t} k(\mathbf{v}, \mathbf{v}') \tag{13}$$

where $\mathbf{T}_{s,t}$ is the similarity between tasks t and s.

3.3.1. Task Similarity Kernel

The task similarity kernel $\mathbf{T}$ plays an important role in the collective link prediction problem. Although we can define heuristic similarity functions by hand, it is better to learn the similarity automatically from the data. In our proposed approach, we assume the task similarity kernel to have a free form. Therefore, it would be a PSD matrix of $T \times T$. We can parameterize $\mathbf{T}$ by the Cholesky decomposition $\mathbf{T} = L^{\mathrm{T}}L$, which can guarantee the PSD constraint for $\mathbf{T}$.

3.3.2. Link Functions

We consider link functions in different link prediction tasks to have the same form but different parameters. The difference between different link prediction tasks should be handled such as the scale difference and bias difference. We also need to adjust the skewness of the rating distribution. Therefore, we generalize the link function in Eq. 9 and consider the following form of link function,

$$g^{(t)}(x) = c^{(t)} x^{1+\alpha^{(t)}} + b^{(t)}, \quad c^{(t)} > 0, \alpha^{(t)} > 0 \tag{14}$$

for task t. Our observation is that most of the rating information in social networks are negative skewed. However, we can relax the constraint of $\alpha > 0$ to make the representation power of the link function stronger for more general cases when we do not have prior on the skewness of the distributions of link data.

3.4. Collective Link Prediction

The inference process of the model is similar to that in standard GP models. For a user with observed ratings $\mathbf{x}$, we need to predict the link between him and an item with latent low rank representation $\mathbf{v}$ in the task t. We first consider prediction for the transformed value $y = g(x)$. The mean and variance of the predictive distribution of y are given by

$$\begin{aligned} m(y) &= \mathbf{k}_{\mathbf{y}}^{\mathrm{T}}(\mathbf{T} \otimes \mathbf{K} + \sigma^{\mathbf{2}}\mathbf{I})_{\mathbb{O}}^{\mathbf{-1}}\mathbf{y}, \\ \sigma^2(y) &= c - \mathbf{k}_{\mathbf{y}}{}^{\mathrm{T}}(\mathbf{T} \otimes \mathbf{K} + \sigma^2\mathbf{I})_{\mathbb{O}}^{-1}\mathbf{k}_{\mathbf{y}} \end{aligned} \tag{15}$$

where $c = \mathrm{T}_{t,t}k(\mathbf{v}, \mathbf{v}) + \sigma^2$ and $(\mathbf{k}_{\mathbf{y}})_j = \mathrm{T}_{t,s}k(\mathbf{v}, \mathbf{v}_j)$ for $\mathbf{v}_j$ in task s. We use the notation $_{\mathbb{O}}$ to denote the submatrix corresponding to the observed entries for the user. The above equation for $m(y)$ indicates that the predicted rating of a user is the weighted combination of other observed ratings of the user. It is similar to the idea of memory-based methods. However, a difference is that here the weights are learned from data. The

[1] http://www.grouplens.org/

mean $m(y)$ can be further decomposed as follows

$$m(y) = \mathbf{T}_{t,t} \sum_{x_j \in \mathbf{X}^{(t)}} w_j k(\mathbf{v}, \mathbf{v}_j) + \sum_s \mathbf{T}_{s,t} \sum_{x_i \in \mathbf{X}^{(s)}} w_i k(\mathbf{v}, \mathbf{v}_i) \tag{16}$$

where $w_i = (\mathbf{C}_{\mathbb{O}}^{-1}\mathbf{k_y})_i$. Therefore, the similarities between the link prediction tasks are also addressed. The first term in the above formula represents the correlation between the test data point and the data in the same task. The second term represents the correlation between the test data point and the data from other tasks where a factor is introduced based on the similarity between tasks. If two tasks are more similar, then the data have stronger influence on the predicted rating in general.

We can further consider the distribution with respect to the value x before transformation of the link function, as follows

$$p(x|y, \mathbf{X}, \theta) = \frac{g'(x)}{\sqrt{2\pi\sigma^2}} \exp(-\frac{1}{2}(\frac{g(x) - y}{\sigma})), \tag{17}$$

and we can predict the median of x by

$$\text{median}(x) = g^{-1}(y) = (\frac{y - b^{(t)}}{c^{(t)}})^{1/(1+\alpha^{(t)})}.$$

4. Parameter Learning with Stochastic Optimization

4.1. The Objective Function

The complete negative log-likelihood of the link data is given as follows,

$$\ln p(\mathbf{Y}|\boldsymbol{\theta}) = -\sum_i (\frac{N_i}{2} \log |\mathbf{C}_{\mathbb{O}}| + \frac{1}{2}(\mathbf{y}_{i,:}^{\mathrm{T}} \mathbf{C}_{\mathbb{O}}^{-1} \mathbf{x}_{i,:})) + \text{const.} \tag{18}$$

where N_i is the number of ratings for user i.

If we directly consider the distribution with respect to the latent variable $\mathbf{X}$, then for the i^{th} user, we can obtain the loglikelihood,

$$\ln p(g(\mathbf{x}_{i,:})|\boldsymbol{\theta}) = -\frac{N_i}{2} \log |\mathbf{C}_{\mathbb{O}}| - \frac{1}{2}(g(\mathbf{x}_{i,:})^{\mathrm{T}} \mathbf{C}_{\mathbb{O}}^{-1} g(\mathbf{x}_{i,:})) + (\log g'(\mathbf{x}_{i,:}))^{\mathrm{T}} \mathbf{1} + \text{const.} \tag{19}$$

The parameters $\boldsymbol{\theta}$ include the task similarity kernel and kernel parameters $\{\theta_i\}$, the parameters in the link function $\{\beta_i\}$ and the low rank representation of the users $\{\mathbf{v}_i\}$. In the next section, we propose an efficient learning algorithm based on stochastic optimization.

4.2. Gradient of Parameters

To make the joint task kernel valid, we need to impose the PSD constraint on the task correlation matrix. We can parameterize $\mathbf{T}$ by the Cholesky decomposition $\mathbf{T} = L^{\mathrm{T}} L$. Since the problem is not convex, the initialization has strong influence on the parameters learned. A good initialization of task correlation is also important. Domain knowledge should be utilized to give a good initialization for task correlation. Another approach is to provide a prior for the task correlation kernel and obtain a MAP estimation.

We can compute the derivative of the log-likelihood with respect to the parameters of the kernel by,

$$\frac{\partial}{\partial \theta_i} \ln p(g(\mathbf{x}_{i,:})|\boldsymbol{\theta}) = -\frac{1}{2}\mathrm{Tr}(\mathbf{C}_{\mathbb{O}}^{-1} \frac{\partial \mathbf{C}_{\mathbb{O}}}{\partial \theta_i}) + \frac{1}{2} \mathbf{y}_{i,:}^{\mathrm{T}} \mathbf{C}_{\mathbb{O}}^{-1} \frac{\partial \mathbf{C}_{\mathbb{O}}}{\partial \theta_i} \mathbf{C}_{\mathbb{O}}^{-1} \mathbf{y}_{i,:}$$

The gradient to the parameters in the link function is,

$$\frac{\partial}{\partial \beta_i} \ln p(g(\mathbf{x}_{i,:})|\boldsymbol{\theta}) = -g(\mathbf{x}_{i,:})^{\mathrm{T}} \mathbf{C}_{\mathbb{O}}^{-1} \frac{\partial g(\mathbf{x}_{i,:})}{\partial \beta_i} + \frac{1}{g'(\mathbf{x}_{i,:})} \frac{\partial g'(\mathbf{x}_{i,:})}{\partial \beta_i}$$

Different from classical GP, we also need to learn the latent feature representation $\mathbf{v}$. Therefore, the gradient to the low dimensional representation of data $\mathbf{v}_i$ is,

$$\frac{\partial}{\partial \mathbf{v}_i} \ln p(g(\mathbf{x}_{i,:})|\boldsymbol{\theta}) =(-(\mathbf{C}_{\mathbb{O}}^{-1}) + \mathbf{C}_{\mathbb{O}}^{-1} \mathbf{y}_i \mathbf{y}_i^{\mathrm{T}} \mathbf{C}_{\mathbb{O}}^{-1}) \mathbf{v}_i$$

After obtaining the gradients to the parameters, we can employ a stochastic gradient descendent algorithm for optimization.

5. Discussion

The relation between multi-task learning and collaborative filtering has been addressed before in (Lawrence, 2003; Abernethy et al., 2009). If we consider the link prediction for each user as a single task, then a link prediction task for users would become a multi-task problem. In our case, we are considering a collection of link prediction tasks. Therefore, our problem of collective link prediction can be regarded as a collection of multi-task learning problems from the multi-task learning point of view. We can see this characteristic by looking at the kernel. Under the GP framework, for a link prediction task, the kernel for link data can be expanded as,

$$\widetilde{\mathbf{K}}^{(t)} = \mathbf{U} \otimes \mathbf{V}^{(t)}$$

For a collective link prediction tasks, the kernel can be expanded as,

$$\widetilde{\mathbf{K}} = \mathbf{T} \otimes \mathbf{U} \otimes \mathbf{V}$$

where the kernel becomes a multi-dimensional kernel in our problem.

6. Experiments

In this section, we show the experiments on the several real world datasets, along with our analysis on the results.

6.1. Experimental Settings

We test the proposed method on several public recommendation datasets, where the items come from different domains or sub-domains. We formulate the recommendation problem in each domain/sub-domain as a link prediction task and thereby construct a collective link prediction task when considering these domains together.

6.1.1. Datasets

We use three representative datasets in our experiments including one of movie ratings dataset, one of book ratings and one from a popular social networking services. In all these datasets, the items can be divided into multiple heterogeneous domains.

- MovieLens is a widely used movie recommendation dataset. It contains 100,000 ratings with scale 1-5. The ratings are given by 943 users on 1,682 movies. The public dataset only contains users who have at least 20 ratings. Besides rating information, the genre information about movies is also available.
- Book-Crossing[2] is a public book ratings dataset. We use a subset of the data consisting of the ratings on books whose category information are available on Amazon, which contains 56,148 ratings with scale 1-10. The ratings are given by 28,503 users on 9,009 books.
- Douban[3] is a social network based recommendation service focused on three types of items including movie, book and music. We crawled the rating information from the Web site and collect a dataset of 10,000 users and 200,000 items including all of movies, books and music.

The Douban dataset crosses three very distinct domains, i.e. book, movie and music, whereas for other two datasets, the items all belong to the same general category (book or movie) but could be further divided into different specific subcategories. For the MovieLens dataset, we chose the 5 most popular genres to form the different domains. For the Book-Crossing dataset, the 4 most general book categories are used to define the domains.

6.1.2. Evaluation

In this paper, we use Mean Absolute Error (MAE) for performance evaluation.

$$\text{MAE} = \frac{\sum_{u,m} |r_{um} - \widehat{r}_{um}|}{N}$$

where r_{um} denotes the ground truth rating of the user u for the item m, and $\widehat{r}_{um}$ denotes the predicted rating. The denominator N is the number of tested ratings. Smaller MAE score corresponds with better performance.

6.1.3. Baselines

We compare our proposed models with the following baselines

- Independent Link Prediction using nonlinear matrix factorization via GP (I-GP) (Lawrence & Urtasun, 2009), which treats different link prediction tasks independently.
- Collective Matrix Factorization (CMF) models (Singh & Gordon, 2008), which is a non-Bayesian model to handle problems involve multiple matrix factorization.
- Joint Link Prediction using multi-relational GP (M-GP) (Xu et al., 2009), which considers multi-relation of the data but neglects the difference between different tasks.

In the following, we refer to our proposed Gaussian processes based collective link prediction method as CLP-GP.

6.2. Experimental Results

6.2.1. Parameters Setting

There are very few parameters to tune in the model, which is one advantage of the nonparametric Bayesian model. One parameter that needs to be set is the latent dimension number. Figure 2 shows the influence of the latent dimension to the performance for the nonlinear matrix factorization model on a subset of MovieLens dataset. We can observe that the performance converges with respect to the dimension after the number of latent dimension reaches 10. Therefore, we set the parameter to 10 in the following experiments. For the M-GP model, we use the parameters learned by the I-GP as the initial values. Similarly,

[2] http://www.informatik.uni-freiburg.de/∼cziegler/BX
[3] http://www.douban.com

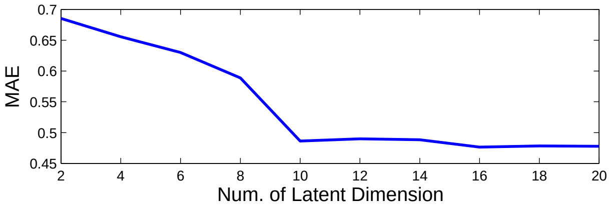

Figure 2. The influence of latent dimension on a subset of MovieLens dataset.

Table 1. The overall experimental results on the three datasets.

MovieLens	I-GP	M-GP	CMF	CLP-GP
-Link	1.4827	0.6569	0.7120	0.6440
+Link	1.3487	0.6353	-	0.6385
Book-Crossing	I-GP	M-GP	CMF	CLP-GP
-Link	0.9385	0.7018	0.8054	0.6547
+Link	0.9317	0.6488	-	0.6014
Douban	I-GP	M-GP	CMF	CLP-GP
-Link	0.7789	0.7772	0.9917	0.7446
+Link	0.7726	0.7625	-	0.7418

for the CLP-GP model we use the parameters learned by the M-GP as initial values and set the similarities between all different tasks as 0.5. We use the RBF kernel as the kernel function.

6.2.2. Experimental Results

Table 1 shows the experimental results on the three datasets. We can observe that our proposed model has the best performance among different datasets compared to the baselines. As we can see, the models that take multiple tasks into consideration (CMF, M-GP, CLP-GP) perform better than the one treating the tasks to be independent. Our proposed CLP-GP model that can learn the similarities between different link prediction tasks performs better than the M-GP model, which indicates that it is useful to explore the relationship between different tasks. Another observation is that the Bayesian model performs better than the classical matrix factorization model, as shown by the performance comparison between CMF and M-GP.

Table 1 also shows the performance comparison between CLP-GP models and its version without link functions. We can observe that the link function introduce a small performance gain consistently over most tasks. The reason why the performance gain is small on some datasets may be that the rating data of these datasets are still distributed similar to Gaussian.

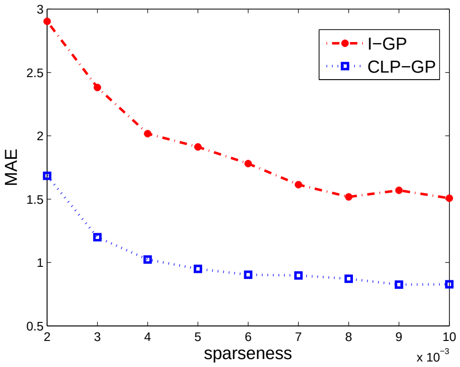

Figure 3. The influence of sparseness on MovieLens dataset for a collective link prediction problem.

6.2.3. Influence on Sparseness

A motivation of this work is to solve the sparseness problem by considering a collection of similar link prediction tasks and transferring knowledge among them. In this experiment, we control the sparseness of the data to see its influence on the performance. Figure 3 shows the performance changes of the algorithms. We can observe that the performance gain increases as the sparseness becomes serious, which is consistent with our intuition. Another observation is that when the data is extremely sparse, the performance gain becomes small again. This phenomenon is because we cannot achieve a satisfiable performance without enough data for any model.

We further consider the setting of transferring the knowledge from some source link prediction tasks to an extremely sparse target link prediction task. Therefore, in this experiment, we only change the sparseness of one target task and keep other tasks relatively dense. Then we check the performance only on the target task. Figure 4 shows the results compared with the baseline of non-transfer setting. We can observe that when the target task is sparse, other related tasks can help boost the performance significantly. The encouraging results demonstrate our proposed model can be used to solve the cold start problem when a new service is added while data of other related services already exist.

6.2.4. Similarities between Tasks

Table 2 shows the similarities learned between five tasks on MovieLens dataset, which is consistent with our intuition. For example, the least similar genre pairs are *Comedy* and *Thriller*, while the most similar genre pairs are *Romance* vs. *Drama*. For the genre *Comedy*, we can see the other genres are ranked into

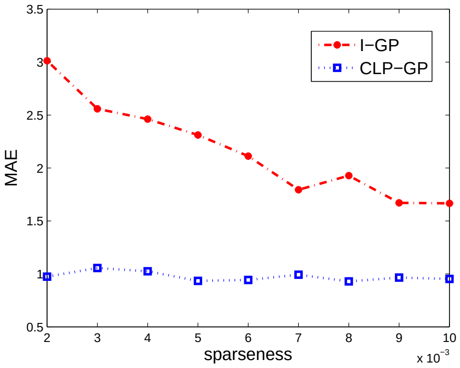

Figure 4. The influence of sparseness on MovieLens dataset for the cold start task in a collective link prediction problem.

Table 2. The similarity matrix cross five link prediction tasks on MovieLens.

	Action	Comedy	Drama	Romance	Thriller
Action	1	0.8479	0.8814	0.8953	0.9253
Comedy	0.8479	1	0.8750	0.8936	0.8422
Drama	0.8814	0.8814	1	0.9392	0.8911
Romance	0.8953	0.8936	0.9392	1	0.8862
Thriller	0.9253	0.8422	0.8911	0.8862	1

the order *Romance*, *Drama*, *Action* and then *Thriller*, which is also reasonable.

7. Conclusion

In this paper, we addressed the collective link prediction problem where several related link prediction tasks are jointly learned. We proposed a nonparametric Bayesian model that considers the similarity between tasks when leveraging all the link data together. We conducted experiments on three real world datasets. We found that transfer learning could help boost the performance of all tasks. Our results confirm the effectiveness of our proposed method.

8. Acknowledgement

We thank the support of project Hong Kong CERG/China-NSFC Grant N_HKUST624/09.

References

Abernethy, J., Bach, F., Evgeniou, T., and Vert, J.P. A new approach to collaborative filtering: Operator estimation with spectral regularization. *J. Mach. Learn. Res*, 10, 2009.

Berkovsky S, Kuflik T, Ricci F. Cross-Domain Mediation in Collaborative Filtering. In *User Modeling* 2007.

Getoor, Lise and Diehl, Christopher P. Link mining: a survey. *SIGKDD Explor. Newsl.*, 7(2), 2005.

Lawrence, Neil D. Gaussian process latent variable models for visualisation of high dimensional data. In *NIPS*, 2003.

Lawrence, Neil D. and Urtasun, Raquel. Non-linear matrix factorization with gaussian processes. In *ICML*, 2009.

Liben-Nowell, David and Kleinberg, Jon. The link prediction problem for social networks. In *CIKM*, New York, NY, USA, 2003. ACM.

Salakhutdinov, Ruslan and Mnih, Andriy. Probabilistic matrix factorization. In *NIPS*, 2007.

Salakhutdinov, Ruslan and Mnih, Andriy. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *ICML*, Helsinki, Finland, 2008. ACM.

Singh, Ajit P. and Gordon, Geoffrey J. Relational learning via collective matrix factorization. In *SIGKDD*, Las Vegas, Nevada, USA, 2008. ACM.

Snelson, Edward, Rasmussen, Carl Edward, and Ghahramani, Zoubin. Warped gaussian processes. In *NIPS*, 2003. MIT Press, 2003.

Tipping, Michael E. and Bishop, Chris M. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61, 1999.

Xu, Zhao, Kersting, Kristian, and Tresp, Volker. Multi-relational learning with gaussian processes. In *IJCAI*, 2009.

Yu, Kai and Chu, Wei. Gaussian process models for link analysis and transfer learning. In *NIPS*. 2007.

Yu, Kai, Chu, Wei, Yu, Shipeng, Tresp, Volker, and Xu, Zhao. Stochastic relational models for discriminative link prediction. In *NIPS*, 2006.

Yu, Kai, Lafferty, John D., Zhu, Shenghuo, and Gong, Yihong. Large-scale collaborative prediction using a nonparametric random effects model. In *ICML*, 2009.

Zhu, Shenghuo, Yu, Kai, Chi, Yun, and Gong, Yihong. Combining content and link for classification using matrix factorization. In *SIGIR*, 2007. ACM.

Zhu, Shenghuo, Yu, Kai, Chi, Yun, and Gong, Yihong. Combining content and link for classification using matrix factorization. In *SIGIR*, 2007. ACM.