

Kernel based Collaborative Topic Regression for Tag Recommendation

Yanwei Guo¹, Hongrong Cheng^{1, a,*}, Mingshuang Tang¹ and Jiaqing Luo¹ and Shijie Zhou¹

¹School of Computer Science and Engineering,
University of Electronic Science and Technology of China, Chengdu 611731, China;

^ahrcheng@uestc.edu.cn

Keywords: Kernel, Collaborative Topic Regression, Tag Recommendation.

Abstract. Tag recommendation is very helpful for users to organize or categorize online resources like music, photos and articles. In recent years, some models, such as collaborative topic regression (CTR) and its variants, have demonstrated promising performance for tag recommendation. In this paper, we propose a novel Bayesian model, called Kernel based CTR (KCTR) to combine kernel based probabilistic matrix factorization which exploits social networks with topic modeling. In contrast to CTR and its existing variants, KCTR model is capable of keeping the real correlation among items rather than ideally assuming the relations among items are mutually independent, which is hardly satisfied in real world. Experimental results on two real datasets show that our method outperforms the state-of-the-art approaches.

Introduction

Tagging has been a popular feature of many websites. For example, CiteULike adopts tags to categorize millions of articles and Flickr allows users to use tags to label and organize photos. However, different users might give different words for the same resource, which will lead to the result that tags created by various users can be inconsistent and idiosyncratic for other users to understand the resource. To alleviate such problems, tag recommendation has become a very hot research topic in recent years [1]. One of the tag recommendation tasks is to predict tags from the already given tags that the resources (such as articles) would like to be used. Thus the tag recommendation system can help to offer relatively stable tags and limit vocabulary of tags to prevent the divergence and ambiguity of tag meaning, with which users only need a few clicks to finish the tagging process [2].

Tag recommendation (TR) needs to combine both item-tag matrix and item content information. Furthermore, there may exist social networks between items. This makes tag recommendation more complicated than traditional recommendation. In such situation, how to exploit the information of the ternary relation is a key challenge for TR research. The existing tag recommendation methods can be roughly divided into three categories [2]: content-based methods, co-occurrence based methods and hybrid methods. Neither content-based methods [3] nor co-occurrence based methods can achieve satisfactory performance. Hence, hybrid methods like Collaborative Topic Regression (CTR) model [4] and its variants have been presented for TR by researchers in recent years.

In this paper, we introduce Kernel function into CTR and propose a novel model, called Kernel based CTR (KCTR) model. Our contributions are as follows: we explore the possibility of extending CTR in conjunction with KPMF to boost tag recommendation performance. KCTR is able to explicitly capture the covariances across the rows, that is the key difference from CTR who assumes an independent latent vector for each row. KCTR is a scalable model which allows to add kernel functions both in tag and item sections to further improve the performance. We test our method on two real-world datasets from CiteULike. The experiment results show that our method outperforms the existing and commonly used methods. Note that although we focus on tag recommendation for articles in this paper, our proposed KCTR model is flexible enough to be used for other recommendation applications such as movie or image tagging.

Problem Statement and Notation

Assume we have a set of items (like articles etc.) X_c to be tagged, with $X_{c,j}^T \in R^D$ denoting the content of item j . In the case of tagging articles in CiteULike, the items refer to papers, and the content information can be their abstracts. Assume we have a set of I tags $\{t_1, t_2, \dots, t_I\}$ as candidates to be recommended to tag each item. Then a tag-item matrix R can be used to represent the tagging information for all the J items. Each matrix entry R_{ij} is a binary variable, where $R_{ij} = 1$ means that tag t_i is associated with item j or $R_{ij} = 0$ otherwise. Tag recommendation is to predict the missing values in $R_{*j} = [R_{1j}, R_{2j}, \dots, R_{Ij}]^T$. Besides, we use I_D to denote a D -dimensional identity matrix and $S = [s_1, s_2, \dots, s_J]$ to denote the relational latent matrix with s_j representing the relational properties of item j .

Proposed Approach

Kernel based Collaborative Topic Regression. The items linked in social networks are most likely to get the same tags, namely, they show similarities in topics or semantics [5]. Hence, our KCTR model integrates the information of social networks among items into the CTR model through kernel functions. The real relations among items in KCTR can be kept by kernel which can be used to describe the social networks. Besides the social networks among items, the social networks among tags also can be presented by kernel in KCTR. The generative process for KCTR is given as follows:

1. For the tags, generate $U_{:,d} \sim GP(0, K_u), [d]_1^d$.
2. For each item j ,
 - a. Draw topic proportions $\theta_j \sim Dirichlet(\alpha)$.
 - b. For each word w_{jn} of item w_j ,
 - i. Draw topic assignment $z_{jn} \sim Mult(\theta_j)$.
 - ii. Draw word $w_{jn} \sim Mult(\beta_{z_{jn}})$.
3. For the items, generate $V_{:,d} \sim GP(\theta_{:,d}, K_v), [d]_1^d$.
4. For each tag-item pair (i,j) , draw the tagging information r_{ij} ,

$$r_{ij} \sim N(u_i^T v_j, c_{ij}^{-1}).$$

Where $K_u \in R^{I \times I}$ and $K_v \in R^{J \times J}$ denote the full covariance matrix for rows and columns of tag-item matrix R respectively. Then K_u and K_v are constructed as the kernel matrices suitable to our problem. Here, we set the kernel function $K = (I + \lambda L)^{-1}$ which is a normalized lapacian kernel, where I is the identity matrix, λ is the kernel parameter and $L = A^{-\frac{1}{2}} \hat{L} A^{-\frac{1}{2}}$ where \hat{L} is the Lapacian matrix incorporating the social network information. $\hat{L} = A - W$ where A is a diagonal matrix whose diagonal elements $A_{ij} = \sum_j W_{ij}$. Here W is an undirected graph of social network with nodes and edges representing items and their links respectively. If there is an edge between item j and j' , $W_{jj'} = 1$. Otherwise, $W_{jj'} = 0$. For simplicity, we denote K_v^{-1} by S_v and K_u^{-1} by S_u .

To learn the KCTR model parameters, we develop an EM-style algorithm to calculate the Maximum A Posteriori (MAP) similar to the approaches in CTR. Maximization of the posterior is equivalent to maximizing the complete log likelihood of U, V, R and θ , as given S_u, S_v , and β .

$$L = -\frac{1}{2} \sum_{d=1}^D u_{:,d}^T S_u u_{:,d} - \frac{1}{2} \sum_{d=1}^D (v_{:,d} - \theta_{:,d})^T S_v (v_{:,d} - \theta_{:,d}) + \sum_j \sum_n \log(\sum_d \theta_{jd} \beta_{d,w_{jn}}) - \sum_{i,j} \frac{c_{ij}}{2} (r_{ij} - u_i^T v_j)^2. \quad (1)$$

We have omitted a constant and set the Dirichlet prior α to 1. Note that the first two terms in (1) could be rewritten as $-\frac{1}{2}\text{Tr}(U^T S_u U)$ and $-\frac{1}{2}\text{Tr}[(V - \theta)^T S_v (V - \theta)]$, where $\text{Tr}(\cdot)$ denotes the trace of matrix. As [6] mentioned, we have

$$\text{Tr}[(V - \theta)^T S_v (V - \theta)] = \sum_{j=1}^J \sum_{j'=1}^J S_v(j, j') [(v_j - \theta_j)^T (v_{j'} - \theta_{j'})] \quad (2)$$

and similarly

$$\text{Tr}(U^T S_u U) = \sum_{i=1}^I \sum_{i'=1}^I S_u(i, i') u_i^T u_{i'}. \quad (3)$$

In the case of tags, there is no information of tags relations, the equation (3) can be simplified to $\lambda_u \sum_i u_i^T u_i$.

We estimate MAP by using coordinate ascent and iteratively optimize the collaborative filtering variables $\{u_i, v_j\}$ and the topic proportions θ_j . Given the current estimate of θ_j , taking the gradient of L with respect to u_i, v_j and setting it to zero, we have

$$u_i \leftarrow (VC_i V^T + \lambda_u I_D)^{-1} VC_j R_i. \quad (4)$$

$$v_j \leftarrow (UC_j U^T + s_{jj} I_D)^{-1} \left[UC_j R_j + \theta_j s_{jj} - \sum_{j' \neq j} S_v(j, j') (v_{j'} - \theta_{j'}) \right]. \quad (5)$$

where C_i is a diagonal matrix with c_{ij} ($j = 1, \dots, J$) as its diagonal entries and R_j is the j th row of R . $S_v(j, j')$ is the entry that the j th row and j' th column of S_v . s_{jj} is the diagonal entry of the j th row of S_v .

Given updated variables $\{u_i, v_j\}$, we can learn the topic proportions θ_j . We separate the items that contain θ_j . Then θ_j is updated by applying Jensen's inequality [4]:

$$\begin{aligned} L(\theta_j) &\geq -\frac{1}{2} \sum_{j'=1}^J S_v(j, j') [(v_j - \theta_j)^T (v_{j'} - \theta_{j'})] \\ &+ \sum_n \sum_d \phi_{jnd} (\log \theta_{jd} \beta_{d, w_{jn}} - \log \phi_{jnd}) \\ &= L(\theta_j, \phi_j) \end{aligned} \quad (6)$$

where $\phi_{jnd} = q(z_{jn} = d)$ and $\phi_j = (\phi_{jnd})_{n=1, d=1}^{N \times D}$. The $L(\theta_j, \phi_j)$ gives the tight lower bound of $L(\theta_j)$.

The optimal ϕ_{jnd} satisfies:

$$\phi_{jnd} \propto \theta_{jd} \beta_{d, w_{jn}}. \quad (7)$$

We use projection gradient approaches to optimize θ_j . Given updated variables $\{u_i, v_j, \theta_j\}$, we can optimize β as in LDA [4].

CTR vs. KCTR. We illustrate the difference between CTR and KCTR. In CTR, v_i (i from 1 to I) is conditionally independent given the prior. As a result, it is unlikely for CTR to capture the correlations which generally exist among columns of matrix V in real scenarios. Conversely, in KCTR, v_i (i from 1 to I) is sampled from a Gaussian process. The covariance K_v of Gaussian process captures the column's underlying correlations. In this way, K_v makes it available to connect each column with the other columns during training. In recommender systems, when two items (columns) share similarities according to the social network, they are more likely to get the same tags. As a result, after training, the corresponding latent factors of the items (columns) sharing some similarities would also be similar.

Experiments

Datasets. Both of the two real-world datasets are from CiteULike, but are collected in different ways. The first dataset, called citeulike-a, is extended by Wang H. et al. [7] by adding the tag information into the original dataset [4]. The second dataset, called citeulike-t, is from [4]. By removing the little-used tags which are used less than 5 times to reduce noise, we get the final datasets as showed in Table 1.

Table 1 Statistics of the Dataset

Dataset	users	items	tags	user-item pairs	citations
citeulike-a	5551	16980	7386	204987	44709
citeulike-t	7947	25975	8311	134860	32565

The content information of the items in the two datasets is extracted from the titles and abstracts of articles. After removing the stop words, we choose the *top N* distinct words according to the tf-idf values. The final sizes (i.e. *top N*) of the vocabulary are 8000 and 2000 for the two datasets respectively.

We use the user-item information to construct the social network similar to [4] which considers that two items exist relations in social network if they have four or more users in common. Here the social network is a undirected graph, in which every value is a binary variable. If two items are linked in social network, we treat the relation between the items as '1', otherwise '0'. After merging the social network and the citation network between papers, we get the final network with 294072 links and 180103 for citeulike-a and citeulike-t respectively.

Experimental Settings. In order to show the improvement of performance by using our KCTR model, we compare it with some state-of-the-art models: PMF [8], KPMF [6] and CTR.

For PMF, we used grid search and find that PMF obtains good performance when $\lambda_u = 0.01$, $\lambda_v = 100$, $a=1$, $b=0.01$ and $D=200$. Note that a and b ($a > b > 0$) are tuning parameters for confident parameters c_{ij} . For CTR, we set parameters $\lambda_u = 0.01$, $\lambda_v = 100$, $a=1$, $b=0.01$. For KPMF, we set parameters $\lambda_u = 0.01$, $\gamma = 100$, $a=1$, $b=0.01$. For KPMF+LDA, $\lambda_u = 0.01$, $\gamma = 100$, $a=1$, $b=0.01$. For PMF+LDA, $\lambda_u = 0.01$, $\lambda_v = 500$, $a=1$, $b=0.01$. For KCTR, we choose the parameters similar to the baseline approaches mentioned above. We set parameters, $\lambda_u = 0.01$, $\gamma = 80$, $a=1$, $b=0.01$.

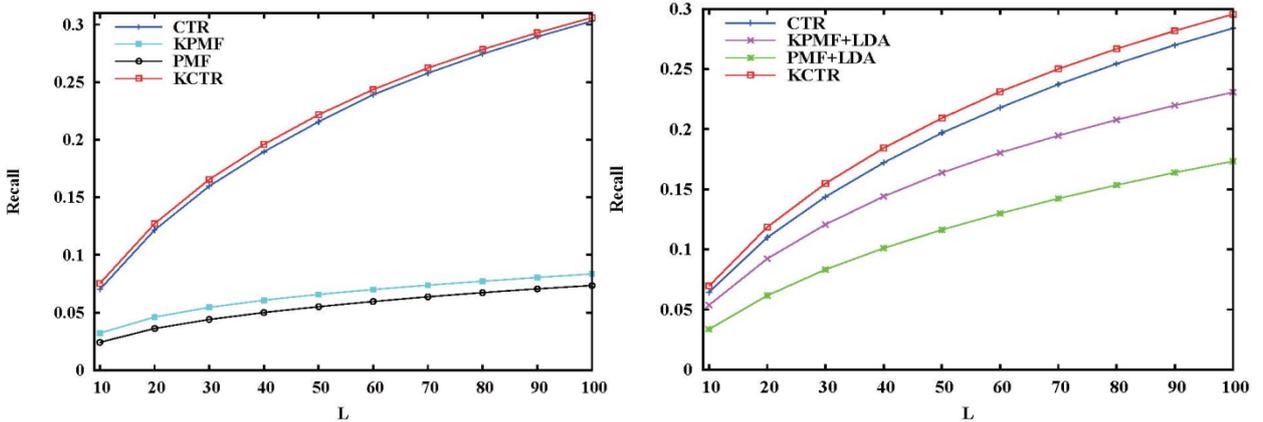


Fig. 1 Experimental result on both datasets. Left shows the $recall@L$ on citeulike-t. Right shows the $recall@L$ on citeulike-a. L ranges from 10 to 100.

Evaluation Metrics. In our experiments, the recommendation is item-oriented. Namely, we recommend tags to items. We present each item (article) with L tags sorted by their predicted rating for the entries in the testing sets. For a fair and easy comparison with the state-of-the-art models, we use recall as our performance evaluation metrics. The $recall@L$ for each item is defined as:

$$recall @ L = \frac{\text{number of tags in top } L \text{ the item is associated with}}{\text{total number of tags the item is associated with}}$$

Results and Analysis. During our experiments, we find that the PMF+LDA and KPMF+LDA whose prior means are initialized based on the topic proportions through LDA outperform PMF and

KPMF respectively. So on the citeulike-a dataset, we use PMF+LDA and KPMF+LDA instead of PMF and KPMF for comparison baselines. Fig. 1 shows the recall@L of the baseline models and KCTR models when we set $L=10, 20, \dots, 100$ respectively in dataset citeulike-a and citeulike-t. As we can see, the performance gap between PMF+LDA, KPMF+LDA and CTR, KCTR is reduced. However, CTR and KCTR still obviously outperform PMF+LDA, KPMF+LDA (or PMF, KPMF) by jointly performing both collaborative filtering and probabilistic topic modeling. Furthermore, KCTR can achieve even higher recall and success by incorporating social network information based on kernel function.

Summary

Since tagging information is usually very sparse, how to effectively combine the users, items and tags to improve recommendation performance is a key challenge for tag recommendation. In this paper, we propose KCTR as a novel hierarchical Bayesian model to incorporate the information of item-tag, item content and social network into the model. KCTR model keeps the real correlation between the rows and columns of the item matrices, rather than treats them independently as in CTR and its variants. Our model is a scalable model which allows to add kernel functions both in tag and item sections to improve the performance. Experiments on two real datasets successfully demonstrate that the proposed model is effective and has better performance than the state of the art.

Acknowledgments

This research is supported by National Nature Foundation under Grant 61133016, 61300094 and the Fundamental Research Funds for the Central Universities under Grant ZYGX2013J083.

References

- [1] Liu J, Li Z, Tang J, et al.: Personalized Geo-Specific Tag Recommendation for Photos on Social Websites. *IEEE Transactions on Multimedia*, 2014, 16(3): 588-600.
- [2] Wang M, Ni B, Hua X S, et al.: Assistive tagging: A survey of multimedia tagging with human-computer joint exploration. *ACM Computing Surveys (CSUR)*, 2012, 44(4): 25.
- [3] Ju S, Hwang K B.: A weighting scheme for tag recommendation in social bookmarking systems. *the ECML/PKDD 2009 Discovery Challenge Workshop*. 2009: 109-118.
- [4] Wang C, Blei D M.: Collaborative topic modeling for recommending scientific articles. In *SIGKDD*, 2011: 448-456.
- [5] Hu X, Tang L, Tang J, et al.: Exploiting social relations for sentiment analysis in microblogging. In *WSDM*, 2013: 537-546.
- [6] Zhou T, Shan H, Banerjee A, et al.: Kernelized Probabilistic Matrix Factorization: Exploiting Graphs and Side Information. In *SDM*. 2012, 12: 403-414.
- [7] Wang H, Chen B, Li W J.: Collaborative topic regression with social regularization for tag recommendation. *AAAI Press*, 2013: 2719-2725.
- [8] Mnih A, Salakhutdinov R.: Probabilistic matrix factorization. *Advances in neural information processing systems*. 2007: 1257-1264.