Tag Dispatch Model with Social Network Regularization for Microblog User Tag Suggestion

Zhiyuan Liu   Cunchao Tu   Maosong Sun
Department of Computer Science and Technology
State Key Lab on Intelligent Technology and Systems
National Lab for Information Science and Technology
Tsinghua University, Beijing 100084, China
{lzy.thu,tucunchao}@gmail.com, sms@tsinghua.edu.cn

Abstract

Microblog is a popular Web 2.0 service which reserves rich information about Web users. In a microblog service, it is a simple and effective way to annotate tags for users to represent their interests and attributes. The attributes and interests of a microblog user usually hide behind the text and network information of the user. In this paper, we propose a probabilistic model, Network-Regularized Tag Dispatch Model (NTDM), for microblog user tag suggestion. NTDM models the semantic relations between words in user descriptions and tags, and takes the social network structure as regularization. Experiments on a real-world dataset demonstrate the effectiveness and efficiency of NTDM compared to other baseline methods.

Title and Abstract in Chinese

用于微博用户标签推荐的社会网络正则化的标签分发模型

微博是Web2.0的重要应用，其中包含了丰富的网络用户信息。在微博中，标签是一种表示用户兴趣和属性的简单有效的方式。一个微博用户的属性和兴趣也通常隐藏在他/她的文本和网络中。本文提出一种概率模型，网络正则化的标签分发模型（NTDM），用来进行微博用户标签推荐。NTDM对用户个人介绍中的词和标签之间的语义关系进行建模，同时将其所在的网络结构信息通过正则化的方式考虑进来。在真实数据上的实验表明，NTDM与其他方法相比更加有效。

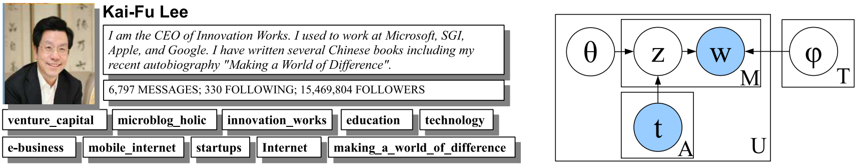
Keywords: user tag suggestion, microblog, tag dispatch model, random walks.

Keywords in Chinese: 用户标签推荐, 微博, 标签分发模型, 随机游走.

1 Introduction

As a popular application in Web 2.0 era, microblog provides a new scheme for sharing information and expressing opinion (Java et al., 2007). Microblog users are able to post short messages within a certain length, and may also follow other users that they are interested in. A microblog service is a typical social network of microblog users with rich text information.

In order to better model user profile and provide high-quality personalized services, many microblog services (e.g., Sina Weibo) allow a user to annotate itself with several tags, which may either describe their interests or attributes. As shown in Fig. 1a, we take Kai-Fu Lee as an example, who is the CEO of Innovation Works and also a famous IT activist. Lee describes himself with several short sentences under his name and also assigns ten tags for himself.



Kai-Fu Lee

I am the CEO of Innovation Works. I used to work at Microsoft, SGI, Apple, and Google. I have written several Chinese books including my recent autobiography "Making a World of Difference".

6,797 MESSAGES; 330 FOLLOWING; 15,469,804 FOLLOWERS

venture_capital | microblog_holic | innovation_works | education | technology | e-business | mobile_internet | startups | Internet | making_a_world_of_difference

(a) Kai-Fu Lee (b) TDM

Figure 1: (a) The example of Kai-Fu Lee. (b) Graphical model of TDM.

In order to collect more accurate tags, many Web services provide tag suggestion to help users annotate. Many studies have been done to suggest tags for products such as books, movies and restaurants (Jaschke et al., 2008; Rendle et al., 2009; Iwata et al., 2009; Si et al., 2010; Liu et al., 2011). However, it is still rarely explored to suggest tags for microblog users. Due to the huge gap between the hidden attributes/interests of microblog users and their tags, it is non-trivial to build an efficient tag suggestion system. In this paper, we focus on this problem and propose a framework for efficient user tag suggestion.

Microblog services contain rich information of users, which can be roughly divided into two major types: (1) **Text Information**. A microblog user may fill a short description about itself and also post many messages. Both of them reveal the attributes or interests of the user (Liu et al., 2012). (2) **Network Information**. A user may follow other users that it is interested in, and can also be followed by other users. Following-behaviors form a social network of microblog users. The neighborhood of a user in this social network also indicates the interests of the user (McPherson et al., 2001). It is intuitive to suggest tags for a user by comprehensively considering both text and network information of the user. The idea of incorporating text and network information has been explored in many tasks such as news recommendation (De Francisci Morales et al., 2012).

In this paper, we first propose Tag Dispatch Model (TDM) for user tag suggestion based on text information. In TDM, each user is represented as a probabilistic distribution over tags, while each tag is represented as a distribution over words. For each user, TDM will learn to dispatch the most appropriate tag to each word in the description. TDM does not take network information into consideration. By assuming that tag distributions do not change dramatically from a user to its neighbors all over the social network, we define a regularizer based on social network structure for TDM, and propose Network-Regularized

TDM (NTDM). In NTDM, the distributions of user tags are smoothed all over social network. When given a new user, NTDM will suggest tags based on its text and neighbors.

2 The Framework

In this section, we present Network-Regularized Tag Dispatch Model as our framework for user tag suggestion. The data to be analyzed is a set of microblog users with their text and network information. Without loss of generality, we use the description of a user as text information, and use the following-relation to build the network. We now formally give some related concepts.

Suppose we have a set of microblog users U. Each user $u \in U$ provides a short description d_u, which can be represented as a sequence of words $x_1, x_2, \ldots, x_{N_u}$, where N_u is the number of words in the description, and each word token x_i is from a fixed word vocabulary W, i.e., $x_i = w \in W$. Following the assumption of bag-of-words, d_u is represented as $\mathbf{x}_u = \{x_i\}_{i=1}^{M_u}$, where M_u is the number of unique words that occur in the description, and we use $c(d_u, w)$ to represent the number of times that word w occurs in the description. Microblog users also form a social network according to their following-behaviors. We denote the network as $G_U = (U, E)$, where U denotes the network nodes (i.e., microblog users) and E denotes the network edges. We denote the weight of an edge (u_i, u_j) as $e(u_i, u_j)$. We define the weights of all edges in E are equal. A microblog user may annotate itself with some tags. For a user u, we denote the annotated tags as $\mathbf{a}_u = \{z_i\}_{i=1}^{A_u}$, where A_u is the number of tags in $\mathbf{a}_u$ and each tag token z_i is from a fixed tag vocabulary T, i.e., $z_i = t \in T$.

The task of user tag suggestion is formalized as follows. Given a user u with no annotated tags, we have to find a set of tags $\mathbf{a}_u$ to maximize $\Pr(\mathbf{a}_u | u, \mathbf{x}_u, G)$. Under independent assumption of tags, we have $\arg\max_{\mathbf{a}_u} \Pr(\mathbf{a}_u | u, \mathbf{x}_u, G) = \arg\max_{\mathbf{a}_u} \prod_{t \in \mathbf{a}_u} \Pr(t | u, \mathbf{x}_u, G)$. Suppose the number of suggested tags A_u is pre-defined, the task becomes a problem of ranking tags according to $\Pr(t | u, \mathbf{x}_u, G)$, and select top-$A_u$ ones as user tags.

2.1 Tag Dispatch Model (TDM)

Tag Dispatch Model (TDM) is a probabilistic graphical model. Like Probabilistic Latent Semantic Analysis (PLSA) (Hofmann, 1999) and Latent Dirichlet Allocation (LDA) (Blei et al., 2003), TDM models each user description as a distribution over tags and generates each word from a tag. Hence, TDM is different from PLSA and LDA in the following two aspects. (1) TDM considers each tag as an *explicit* topic. In other words, TDM models with *explicit tags* rather than latent topics. TDM incorporates user-annotated tags by regarding each word in user descriptions as generated from a tag. This is similar to the setting of Labeled LDA (Ramage et al., 2009). (2) When learning the mixture of tags for the description of a user, TDM constrains the distribution only having values on those tags that have been annotated by the user.

PLSA and LDA are two popular statistical topic models in information retrieval and natural language processing. In this paper, we build TDM inspired by the idea of PLSA and incorporate the advantages of LDA to avoid over-fitting. Suppose descriptions of all users in U form a collection of documents D_U. The graphical model of TDM is shown in Fig. 1b, where the observed variables are shaded. Since the generative process is to select and dispatch a tag to each word in user descriptions, we name the model as Tag *Dispatch* Model. In order to fulfill the requirement that the tag distribution of a user is restricted to its annotated tags, we set $\Pr(t|u, \mathbf{a}_u) = 0$ for all $t \notin a_u$. In other words, $\sum_{t \in \mathbf{a}_u} \Pr(t|u, \mathbf{a}_u) = 1$. The log likelihood of generating a collection D_U in TDM is formalized as $L(D_U) = \sum_{d_u \in D_U} \sum_{w \in \mathbf{x}_u} c(\mathbf{x}_u, w) \sum_{t \in T} \Pr(w|t) \Pr(t|u, \mathbf{a}_u)$. In TDM, the parameters are θ and ϕ, where $\theta_{tu} = \Pr(t|u, \mathbf{a}_u)$ and $\phi_{wt} = \Pr(w|t)$. Since each d_u belongs to a user u, we also say $\Pr(t|u) = \Pr(t|u, \mathbf{a}_u)$, which indicates the probabilistic distribution over tags given a user.

The parameters of TDM (i.e., θ and ϕ) can be estimated using the Expectation Maximization (EM) algorithm (Dempster et al., 1977). EM algorithm will iteratively computes a local maximum of $L(D_U)$. In the E-step of $(p+1)$th iteration of TDM, the posterior probabilities of latent variables (i.e., the distribution over tags on each z_i corresponding to word $x_i = w$ in $\mathbf{x}_u$ with $\mathbf{a}_u$) are calculated according to the parameters estimated in the pth iteration (i.e., $\theta^{(p)}$ and $\phi^{(p)}$) as follows,

$$\Pr(z_i = t | x_i = w, u, \mathbf{a}_u) = \frac{\Pr^{(p)}(w|t) \Pr^{(p)}(t|u, \mathbf{a}_u)}{\sum_{t \in \mathbf{a}_u} \Pr^{(p)}(w|t) \Pr^{(p)}(t|u, \mathbf{a}_u)}. \tag{1}$$

Following the common practice as shown in PLSA (Hofmann, 1999), we obtain the update equations for the M-step of the $(p+1)$th iteration in TDM as follows:

$$\phi_{wt}^{(p+1)} = \Pr^{(p+1)}(w|t) = \frac{\sum_{u \in U} c(\mathbf{x}_u, w) \Pr(t|w, u, \mathbf{a}_u) + \beta}{\sum_{w \in W} \sum_{u \in U} c(\mathbf{x}_u, w) \Pr(t|w, u, \mathbf{a}_u) + |W|\beta}, \tag{2}$$

$$\theta_{tu}^{(p+1)} = \Pr^{(p+1)}(t|u, \mathbf{a}_u) = \frac{\sum_{w \in W} c(\mathbf{x}_u, w) \Pr(t|w, u, \mathbf{a}_u) + \alpha}{\sum_{t \in \mathbf{a}_u} \sum_{w \in W} c(\mathbf{x}_u, w) \Pr(t|w, u, \mathbf{a}_u) + A_u \alpha}. \tag{3}$$

In Equation (2) and Equation (3), we follow the comparative analysis of latent Dirichlet allocation (Blei et al., 2003), and introduce hyper-parameters α and β to avoid over-fitting. In this paper, we set $\alpha = 50/|T|$ and $\beta = 0.01$. EM algorithm of TDM will run iteratively until a termination condition is satisfied.

After estimating parameters θ and ϕ of TDM, we can suggest tags for a new microblog user u with description $\mathbf{x}_u$ as follows. Suppose all tags in T are candidates for this user. We perform EM algorithm to estimate $\Pr(t|u, \mathbf{x}_u)$ while keeping $\Pr(t|w)$ fixed. Then we rank candidate tags according to $\Pr(t|u, \mathbf{x}_u)$ and select top-A_u as suggested tags.

We have $c(\mathbf{x}_u, w)$ in Equation (2) and (3), which indicates the importance of w in $\mathbf{x}_u$. In practice, a word that occurs frequently doest not indicate it is important. In this paper, we estimate the importance of a word w in $\mathbf{x}_u$ using term frequency and inverse user frequency (TFIUF) as follows:

$$\Pr(w|\mathbf{x}_u) = \frac{c(\mathbf{x}_u, w)}{\sum_{w \in \mathbf{x}_u} c(\mathbf{x}_u, w)} \times \log \frac{|U|}{|\{w \in \mathbf{x}_u\}_{u \in U}|}. \tag{4}$$

Here the first part is term frequency of word w in $\mathbf{x}_u$, and the second is the inverse user frequency, where user frequency is the proportion of users who use word w in their descriptions. The idea of TFIUF is similar to term frequency and inverse document frequency (TFIDF) (Salton and Buckley, 1988) which is widely adopted in information retrieval.

2.2 Network-Regularized Tag Dispatch Model (NTDM)

We take the network structure into account as a regularization for TDM. In the context of social network, we assume that the users who are connected with each other should share more interests and attributes, and thus should have similar tag distributions, i.e., for a connected user pair $(u,v)\in E$, $\Pr(t|u,\mathbf{a}_u)$ is similar to $\Pr(t|v,\mathbf{a}_v)$.

Formally, given a collection of microblog users U with their descriptions D_U and social network G_U, we define the regularized likelihood as $L(D_U,G_U) = (1-\alpha)L(D_U) - \alpha R(D_U,G_U)$, where $L(D_U)$ is the log likelihood of generating user descriptions, and $R(D_U,G_U)$ is a harmonic regularizer defined on the social network G_U. Similar to graph harmonic function (Zhu et al., 2003), we define $R(D_U,G_U)$ as $R(D_U,G_U) = \frac{1}{2}\sum_{(u,v)\in E} e(u,v)\sum_{t\in T}\big(\Pr(t|u,\mathbf{a}_u)-\Pr(t|v,\mathbf{a}_v)\big)^2$, where $e(u,v)$ is the weight of edge (u,v), and α is the harmonic factor ranging from 0 to 1. Since $L(D_U)$ indicates the probability that user descriptions are generated from the model, we can maximize $L(D_U)$ to find optimal model parameters (i.e., θ and ϕ) with respect to user descriptions. $R(D_U,G_U)$ indicates the weighted average distance in terms of tag distributions between any two connected users in the social network. We maximize $-R(D_U,G_U)$ (i.e., minimizing $R(D_U,G_U)$) to smooth the tag distributions over the social network, i.e., the neighbored users will tend to share similar tag distributions. The harmonic factor α controls trade-off between data likelihood and regularization. When $\alpha=0$, the regularized likelihood will be the same to TDM. When $\alpha=1$, the regularized likelihood will only consider the network structure, which likes clustering based on network structure.

We will also use EM algorithms to estimate parameters of NTDM. We can see that NTDM and TDM share the same latent variables, i.e., tag distribution conditional over a word in user description $\Pr(t|w,u,\mathbf{a}_u)$. We can also use Equation (1) to compute the latent variables for NTDM. The M-step in NTDM is more complicated than that in TDM due to the harmonic regularization. The estimation of $\Pr(w|t)$ does not have relations to regularization. Hence we can update $\phi_{wt}=\Pr(w|t)$ in the same way as in Equation (2). Since θ is involved in the regularizer, we do not have a closed form solution to update θ. As proposed in (Mei et al., 2008; Cai et al., 2008), we can iteratively update and obtain

$$\Pr{}_{i+1}^{(p+1)}(t|u,\mathbf{a}_u) = (1-\lambda)\Pr{}_{i}^{(p+1)}(t|u,\mathbf{a}_u) + \lambda\frac{\sum_{(v,u)\in E} e(v,u)\Pr{}_{i}^{(p+1)}(t|v,\mathbf{a}_v)}{\sum_{(v,u)\in E} e(v,u)}, \tag{5}$$

where i is the number of the inner iterations, and λ is a damping factor ranging from 0 to 1. When $\lambda=0$, NTDM becomes into TDM without considering network structure. When $\lambda=1$, it indicates that the new tag distribution of u is the average of the old tag distributions of its neighbors. In experiments, we set $\lambda=0.15$ which follows most settings in random walks (Langville and Meyer, 2004). The iterative random walks with Equation (5) will make the tag distributions smoother over the microblog social network. In practice, not all users in the dataset have annotated themselves with tags. For a user u that has not annotated itself with tags, we set $\mathbf{a}_u = T$. In Equation (5) of NTDM, we set $\Pr{}_{i+1}^{(p+1)}(t|u,\mathbf{a}_u) = 0$ if $t\notin\mathbf{a}_u$. This will avoid tag drift during iteration.

2.3 User Tag Suggestion based on NTDM

Given a user u with its description $\mathbf{x}_u$, NTDM suggests tags as follows. If u belongs to the dataset (i.e., $u\in U$), we have obtained its tag distribution with Equation (5) with learning of NTDM, and can suggest top-ranked tags according to $\Pr^{(p+1)}(t|u,\mathbf{a}_u)$.

If the new user u does not belong to the dataset (i.e., $u\notin U$), we estimate $\Pr(t|u,\mathbf{a}_u)$ in two ways: (1) We use EM algorithm to estimate the tag distribution of u based on user description $\mathbf{x}_u$ and NTDM parameters ϕ. The difference is in the process we do not necessarily modify ϕ and just update θ_u. We denote the text-based tag distribution as $\Pr_T(t|u,\mathbf{a}_u)$. (2) We estimate the tag distribution of u based on its neighbors. We assume that all neighbors of u belong to U, and denote the set of neighbors as U_u. We estimate the tag distribution as $\Pr(t|u,\mathbf{a}_u) = \sum_{v\in U_u} e(v,u)\Pr(t|v,\mathbf{a}_v)/\sum_{v\in U_u} e(v,u)$, where $\Pr(t|v,\mathbf{a}_v)$ is the tag distribution of $v \in U$ estimated in NTDM. We denote the network-based tag distribution as $\Pr_N(t|u,\mathbf{a}_u)$. Finally, we integrate the text-based and network-based tag distribution together with smoothing factor λ: $\Pr(t|u,\mathbf{a}_u) = (1-\lambda)\Pr_T(t|u,\mathbf{a}_u) + \lambda \Pr_N(t|u,\mathbf{a}_u)$. Similar to Equation (5), we also set $\lambda = 0.15$.

3 Experiments

We crawled 2 million users from Sina Weibo for experiments. These users are all active and post messages frequently. In order to better demonstrate the effectiveness of our method, from these users we select $341,353$ users who have both descriptions and tags as the dataset. We further divide the dataset by randomly selecting $10,000$ users as test set and the rest users as training set. We use precision/recall for evaluation. For a user, we denote the original tags (gold standard) as T_a, the suggested tags as T_s, and the correctly suggested tags as $T_s \cap T_a$. Then, precision and recall are defined as $p = (T_s \cap T_a)/T_s$ and $r = (T_s \cap T_a)/T_a$.

3.1 Evaluation on User Tag Suggestion

To evaluate the performance of NTDM for social tag suggestion, we select two major types of baseline methods for comparison: context-based methods which suggest tags relying on user descriptions, and network-based methods which suggest tags according to the neighborhood information of users.

Text-Based Methods. There are many text-based methods proposed for social tag suggestion. In this paper, we use the following text-based methods as baselines. (1) *Feature Driven Methods*. We regard user tag suggestion as a multi-label classification task, and use feature driven methods to train classifiers. In these methods, the probability of a user u being annotated with tag t is computed as $\Pr(t|u) = \sum_{w\in\mathbf{x}_u} \Pr(t|w)\Pr(w|\mathbf{x}_u)$. We use TFIUF defined in Equation (4) to measure $\Pr(w|\mathbf{x}_u)$. There are various statistical measures to estimate $\Pr(t|w)$. We select Pointwise Mutual Information (PMI) (Lin, 1998) and Normalized Google Distance (NGD) (Cilibrasi and Vitanyi, 2007) for estimation. In experiments, we denote the two feature driven methods as PMI-T and NGD-T, respectively. (2) *k Nearest Neighbor (kNN)*. kNN is a classification method based on closest training instances in the feature space (Mishne, 2006; Li et al., 2009). In user tag suggestion, given a user u, kNN finds k nearest neighbors according to their description similarities with u and selects tags by majority vote of neighbors for suggestion. In experiments, we set $k = 5$ which achieves the best performance of kNN. In experiments, we denote the method as kNN-T. (3) *TagLDA*. TagLDA (Krestel et al., 2009; Si and Sun, 2009) is a representative latent topic model by extending latent Dirichlet allocation (LDA) (Blei et al., 2003). Using a collection of annotated users, TagLDA will learn the distributions over words and tags for each topic. Given a novel user, TagLDA will first infer the topic distribution according to the user's description and then suggest tags based on the topic distribution. (4) *Tag Dispatch Model (TDM)*. TDM can be regarded as a text-based version of NTDM, which only considers user descriptions for user tag suggestion. Different from TagLDA, TDM uses tags as *explicit* topics to directly build semantic relations between words and tags.

Network-Based Methods. Network-based methods consider the network structure for social tag suggestion. The basic idea is that a tag will be suggested to a user if the tag is widely annotated by the neighbors of the user. Similar to text-based methods, we use the tags annotated by neighbors as features to build feature-driven classifiers. We formalize the probability of t given a user u as $\Pr(t|u) = \sum_{s\in T}\Pr(s|t)\Pr(s|U_u)$, where U_u is the neighbors of u, $\Pr(s|U_u)$ is the importance of a tag s in neighbors of u, and $\Pr(s|t)$ indicates the probability of s given t. In this equation, $\Pr(s|U_u)$ is estimated as $\Pr(s|U_u) = (|U_{tu}|)/(|U_u|)$, where $|U_{tu}|$ is the number of neighbors that annotate tag s, and $|U_u|$ is the total number of neighbors. $\Pr(s|t)$ can be measured using either PMI or NGD. In experiments, we denote the two methods as PMI-N and NGD-N, respectively.

Hybrid Methods. We can also take text features and network features together and use NB, PMI and NGD as classifiers. Under the assumption of naive Bayes, it is straightforward to combine the two types of features as $\Pr(t|u) = \sum_{w\in\mathbf{x}_u}\Pr(t|w)\Pr(w|\mathbf{x}_u) + \sum_{s\in T}\Pr(t|s)\Pr(s|U_u)$. In hybrid methods, we can also use either PMI or NGD. Hence, in experiments, we denote the two hybrid methods as PMI-H and NGD-H.

3.1.1 Evaluation Results and Analysis

In Figure 2 we show the precision-recall curves of various baseline methods and NTDM on test set. Each point of a precision-recall curve represents different numbers of suggested tags from $M = 1$ (bottom right, with higher precision and lower recall) to $M = 6$ (upper left, with higher recall but lower precision) respectively. The closer the curve to the upper right, the better the overall performance of the method. Hence, in experiments we focus on evaluating the performance when $M \le 6$ since the average number of tags per user in the dataset is 6.0.

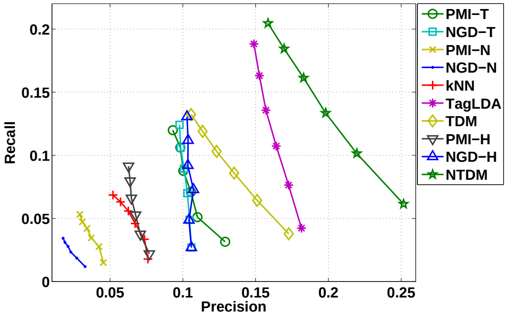


Figure 2: Evaluation results when suggesting tags from $M = 1$ to $M = 6$.

From Figure 2 we have the following observations. (1) NTDM significantly outperforms other methods when M ranges from 1 to 6. The significance test is performed by using bootstrap re-sampling with 95% confidence. This indicates that NTDM is efficient and effective for user tag suggestion. Other text-based and network-based methods perform poorly because independently using either text information or network information will be insufficient to capture the attributes and interests of users. Although TagLDA performs better than TDM, NTDM outperforms TagLDA significantly. This indicates that it is crucial to take network structure into consideration. (2) PMI-H and NGD-H perform poor compared to NTDM. Both methods are even worse than PMI-T and NGD-T. This suggests that naive hybrid of text and network information will not eventually lead to better results. Essentially, we have to find a smart way to combine the two types of information. This is what NTDM is proposed to do, and the experiment results demonstrate its effectiveness.

3.1.2 Case Studies

In Table 1 we show top words ranked by Pr($w|t$) for several tags of Kai-Fu Lee. We observe that NTDM can sufficiently capture the semantic relations between words and tags, while TDM introduces noise. For example, in TDM the top words “optimization”, “factory” and “Jinan” of the tag “e-business” are, to some extent, not tightly correlated with the tag.

Tag	Top Words Ranked by Pr($w|t$)
venture_capital	venture_capital, VC, early_stage, copartner, minor_enterprises
education	parents, children, coaching, normal_university, admission
e-business	B2C, supply, Alibaba, supermarket, B2B
mobile_Internet	Internet, terminal, LBS, summit, android

Table 1: Top words ranked by Pr($w|t$) for some tags of Kai-Fu Lee.

With accurate semantic relations, NTDM suggests better tags for microblog users. Take Kai-Fu Lee for example, top-5 tags suggested by NTDM are “startups”,“Internet”, “Google”, “e-business” and “mobile_Internet”. In this list, although “Google” is not annotated by Lee, it reflects the fact that Lee used to work as President of Google China from 2005 to 2009. Meanwhile, TDM suggests “Google”, “Apple”, “startups”, “photographing” and “post_80s”; TagLDA suggests “post_80s”, “Internet”, “music”, “movie” and “travel”. We have the following observations: (1) TagLDA tends to suggest common tags irrelevant to the user. This is the common issue shared by latent topic models, which project both descriptions and tags into topic space for measuring relatedness and suffer from the over-generalization problem. (2) The last two tags suggested by TDM are roughly not correlated to Lee, which is a natural consequence of not considering network structure for regularization.

Conclusion and Future Work

This paper presents NTDM for microblog user tag suggestion. NTDM models the semantic relations between words and tags, as well as taking social network structure as regularization. Experiments on the real-world dataset demonstrate that NTDM is sufficient to combine the text information and network information of users for user tag suggestion.

We design the following research plans. (1) NTDM considers edge weights of all connected users being equal for simplicity. In future, we plan to incorporate more microblog information to estimate edge weights, and further make the network regularization more accurate. (2) This paper does not take user posts into consideration. We plan to model more complex text and network information for user tag suggestion.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC) under the grant No. 61170196 and 61202140. The authors would like to thank Mr. Bin Liang for providing data.

References

Blei, D., Ng, A., and Jordan, M. (2003). Latent dirichlet allocation. *JMLR*, 3:993–1022.

Cai, D., Mei, Q., Han, J., and Zhai, C. (2008). Modeling hidden topics on document manifold. In *Proceedings of CIKM*, pages 911–920.

Cilibrasi, R. and Vitanyi, P. (2007). The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, 19(3):370–383.

De Francisci Morales, G., Gionis, A., and Lucchese, C. (2012). From chatter to headlines: harnessing the real-time web for personalized news recommendation. In *Proceedings of WSDM*, pages 153–162.

Dempster, A., Laird, N., Rubin, D., et al. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of SIGIR*, pages 50–57.

Iwata, T., Yamada, T., and Ueda, N. (2009). Modeling social annotation data with content relevance using a topic model. In *Proceedings of NIPS*, pages 835–843.

Jaschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., and Stumme, G. (2008). Tag recommendations in social bookmarking systems. *AI Communications*, 21(4):231–247.

Java, A., Song, X., Finin, T., and Tseng, B. (2007). Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 56–65.

Krestel, R., Fankhauser, P., and Nejdl, W. (2009). Latent dirichlet allocation for tag recommendation. In *Proceedings of ACM RecSys*, pages 61–68.

Langville, A. and Meyer, C. (2004). Deeper inside pagerank. *Internet Mathematics*, 1(3):335–380.

Li, X., Snoek, C., and Worring, M. (2009). Learning social tag relevance by neighbor voting. *IEEE Transactions on Multimedia*, 11(7):1310–1322.

Lin, D. (1998). An information-theoretic definition of similarity. In *Proceedings of ICML*, pages 296–304.

Liu, Z., Chen, X., and Sun, M. (2011). A simple word trigger method for social tag suggestion. In *Proceedings of EMNLP*, pages 1577–1588.

Liu, Z., Chen, X., and Sun, M. (2012). Mining the interests of chinese microbloggers via keyword extraction. *Frontiers of Computer Science*, 6(1):76–87.

McPherson, M., Smith-Lovin, L., and Cook, J. (2001). Birds of a feather: Homophily in social networks. *Annual review of sociology*, pages 415–444.

Mei, Q., Cai, D., Zhang, D., and Zhai, C. (2008). Topic modeling with network regularization. In *Proceedings of WWW*, pages 101–110.

Mishne, G. (2006). Autotag: a collaborative approach to automated tag assignment for weblog posts. In *Proceedings of WWW*, pages 953–954.

Ramage, D., Hall, D., Nallapati, R., and Manning, C. (2009). Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of EMNLP*, pages 248–256.

Rendle, S., Balby Marinho, L., Nanopoulos, A., and Schmidt-Thieme, L. (2009). Learning optimal ranking with tensor factorization for tag recommendation. In *Proceedings of KDD*, pages 727–736.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing and management*, 24(5):513–523.

Si, X., Liu, Z., and Sun, M. (2010). Modeling social annotations via latent reason identification. *IEEE Intelligent Systems*, 25(6):42 – 49.

Si, X. and Sun, M. (2009). Tag-LDA for scalable real-time tag recommendation. *Journal of Computational Information Systems*, 6(1):23–31.

Zhu, X., Ghahramani, Z., and Lafferty, J. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of ICML*, pages 912–919.