Chapter 43
Tags Recommending Based on Social Graph

Benyang Xu and Hongming Zhu

Abstract As fast development of social network, UGC (user generated content) has played a very important role in "Web 2.0". However most of UGC is non-structured data, which is hard to be used by search engine or user recommending system. Social mining is the way to make UGC accessible. But UGC are trivial, noisy, sparse, causing social mining methods inefficient. In this paper, we propose a tag recommending approach based on social graph. Social graphic recommending can reduce mining depending on UGC, thus be able to generate high quality tags. Our most contribution is to combine social graph with LDA algorithm to find users' latent common interest, thus extract tags. We did experiment on real data crawled from Sina Weibo. The evaluation showed that our approach archived much better precision and recall than baseline methods.

Keywords Social mining · Tags recommendation · LDA · Social graph

43.1 Introduction

Social network sites (SNSs), such as Facebook, Twitter, Sina Weibo has attracted millions of users since introduced. SNSs is playing a very important role in "Web 2.0" with countless of UGC are generated. However, the lack of centralized

B. Xu (✉) · H. Zhu
School of Software Enginering, Tongji University Jading Campus, Caoan Rd 4800 201804 Shanghai, China
e-mail: xubenyang@gmail.com

H. Zhu
e-mail: hongming.zhu@gmail.com

W. Lu et al. (eds.), *Proceedings of the 2012 International Conference on Information Technology and Software Engineering*, Lecture Notes in Electrical Engineering 211,
DOI: 10.1007/978-3-642-34522-7_43,

organizing and well-structured makes UGC become a very big challenge to mine the latent value in the contents. Make it hard to analyze the social networks [1].

To solve the problem, tagging is an effective way to encode interests and characters in the contents and users. Tags play a very important role in tag-based social interest discovery [2], making UGC accessible by search engines, recommending system and advertising systems. Social Tagging [3, 4] has emerged as an effective way to alleviate some of these challenges. Many methods and studies have focused on tag generation and recommending. There are two general approaches to generate tags for items:

1. *Social Tagging System* allows users to create and manage tags of social items. This idea is simple: users can select any meaningful tags or keywords to encode the characteristic attributes of items.
2. *Automatic Tag Recommendation* has been proposed in multiple studies [5–8]. Given an item, the task of automated tag recommendation is to suggest several most relevant tags to the contents using machine learning algorithms.

However, all these approaches have their own defects. For manually tags, since they are not restricted to a certain vocabulary, users can pick any tags they prefer to describe the resource. So these tags can be inconsistent, trivial, or false, duo to the users' personal terminology, depending too much on the users' choice [10]. For automatic-generated tags, most approaches assume the pre-existing tags or the text description of the items. As a result, we do not expect these approaches could still perform well while applied to social sites or systems, that does not built around tags and text description.

This paper aims to improve quality of tag recommendation, overcoming the problem of automatic tag recommending approaches, which depending too much on presenting of text and pre-existing tags. To archive this, we use LDA with social graph together to infer tags on a large scale of group. Social graph implies people are connected by common interest. People with common interest will group together automatically which is called clustering, we can see that in Fig. 43.1. Our approach try to learn user's common interest instead of analyze the basic text description of the user. This will be more accurate and precise to generate high quality tags.

The rest of this paper is organized as follow. In Sect. 43.2 we introduce LDA algorithm and then present our approach. Next we will show our experiments on real data from Sina Weibo and compare it with our baseline approaches in Sect. 43.3. In Sect. 43.4, we conclude this paper.

43.2 Tags Recommendation

In this section, we introduce LDA first, and then explain our approach how to extract tags by adapting LDA and follower graph together.

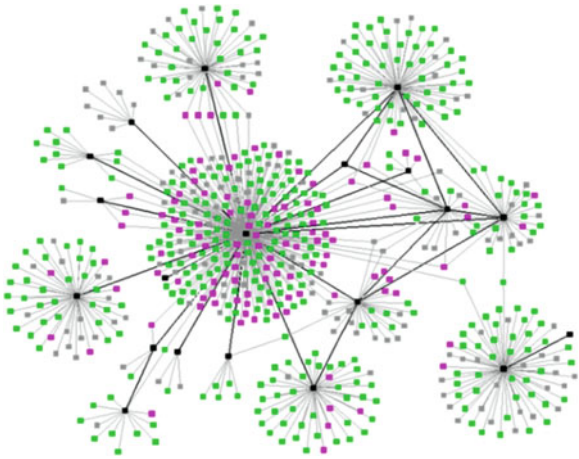

Fig. 43.1 Users are clustering with common interests on social sites

43.2.1 Latent Dirichlet Allocation

Latent Dirichlet Allocation is a probabilistic latent topic model introduced by Blei in 2003 [9]. The general idea of LDA is based on the assume that a person writing a document with certain topics in mind. When wring a document, first pick some topics for the document, and then pick words from vocabulary that with a certain probability about that topic. By repeating this step, a document finally is generated with a mixture of topics, representing by a collection of words. Here, LDA do not care the order of the words, so the probabilities of different words with different orders are treated the same, known as "Bag-of-Words" [11].

In this model, topic is the latent variable added. With topic, LDA helps to explain the similarity of words by grouping them into different topics. The modeling process of LDA can be described finding a mixture of topics for each resource. If there are T number of topics specified, and probability of the ith word w_i can be formalized as following:

$$p(w_i|d) = \sum_{j=1}^{z} p(w_i|z_j)p(z_j|d) \tag{43.1}$$

where w_i represents the ith word in vocabulary. d represents the document. z_j represents the latent topic for ith word. $P(w_i|d)$ is the probability of ith word in the given document d, with latent topic variable z_j. $P(t_i|z_j)$ is the probability of ith word in topic z_j. $p(z_j|d)$ is the probability of picking a word in topic in document d. LDA use latent topic variables to link words and documents. Figure 43.2 shows the plate graph representation of LDA.

The probability of latent topics $p(z|d)$ follows a multinomial distribution with parameter θ that has a Dirichlet distribution with parameter α as it prior. The probability of a word in a topic $p(w|z)$ follows a multinomial distribution with

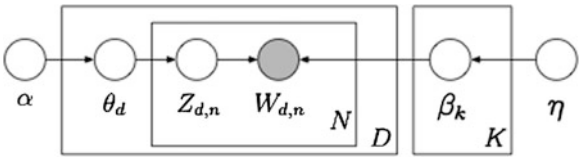

Fig. 43.2 Plate graph of latent Dirichlet allocation

parameter φ that has a Dirichlet distribution with parameter β. With this notion, the text generation process can be explained as follow:

1. For all aspects k, sample $\varphi_k \sim Dir(\beta)$
2. For all entities e, sample $\theta_e \sim Dir(\alpha)$
3. For each term-slot in d_e
   a. Sample an aspect $z_j \sim Mult(\theta_e)$
   b. Sample a term $w_i = w \sim Mult(\varphi_{zj})$

Then parameter Θ and Φ can be learned by many infer methods, such as EM algorithm and Gibbs sampling method. Compared to EM, Gibbs sampling can guarantee a better convergence. Here we choose Gibbs sampling as the training method for LDA. It iterates multiple times over each word w_i in document, and samples a new topic j for the word based on the core probability method $P(z_j|t_i, t-_i, z_{-j})$, until the LDA model parameters converge.

$$P(z_j|t_i, t_{-i}, z_{-j}) \propto \frac{n_{j|e,-i} + \alpha}{|d_{e,-i}| + \alpha|K|} * \frac{n_{w|j,-i}}{n \cdot |j, -i} \tag{43.2}$$

where $n_{j|e,-i}$ is the number of times aspect j is observed for entity e, $n_{w|j,-i}$ is the number of times word w is sampled from aspect j, $|d_{e,-i}|$ is number of word occurs associated with e, and $n_{\cdot|j,-i}$ is the total number of words generated from aspect j. After learning the Dirichlet distribution Θ and Φ. The posterior probabilities $p(w|z)$ *and* $p(z|d)$ can be figured out.

43.2.2 Tags Recommendation Based on Social Graph

In this paper, we focus on micro blogging service which is one-directional site, so the social graph is the follower graph. As we can see from Fig. 43.1, people on social sites are clustering with the same common interest [13]. And some other studies have focused on taking advantages of social networks [1, 12]. Our approach does not try to recommend tags on the text description of users, we take the latent advantage of social graph. The general purpose of a user willing to follow somebody is because they share the common interests. So our approach add "common interest" as the latent variable in social graph, and concretely involves two steps to generate tags for a user:

1. Learn users' interest probability distribution.
2. Recommend tags from interest probability distribution

43.2.2.1 Use LDA to Learn User's Interest

As we introduced in Sect. 43.2.2.1, LDA is a topic model method mostly used to do text categorizing or tag extraction on documents. With the same idea, we take LDA on social graph to infer the probability of user distribution. While LDA adopt "word > topic > document" pattern to generate documents. In social graph, we can treat each follower as "words" in LDA, treat common interests as "topics", treat the followed user as the "document". So the follower graph can be generated with pattern: "follower > common interest > followed-user". If there are M number of common interests are given, the probability of ith user ui follow the target user as follow:

$$\mathrm{P}(\mathrm{u}_i|\mathrm{u}) = \sum_{j=1}^{M} P(u_i|I_j)P(I_j|u) \tag{43.3}$$

Similar with original LDA, u_i represents the ith user. $P(u_i|u)$ indicates the probability that u_i will follow u. I_j is the latent variable representing ith interest. $P(u_i|I_j)$ represents the probability that I_j is ui's interest. While $P(I_j|u)$ represents the probability that I_j is the followed user's interest. Then we can learn the user's interest distribution, with the same approach we introduced in Sect. 43.2.2.1.

43.2.2.2 From Interest to Tags

After learning the users' interest distribution, then we can use this to extract the actual tags of users, with representing with form of words. So the probability of word ti can the the tag of u can be formalized as follow:

$$P(t|u) = \sum_{j=1}^{M} P(t|I_j)P(I_j|u) \tag{43.4}$$

$$P(t|I_j) = \sum_{i=1}^{N} P(t|u_i)P(u_i|I_j) \tag{43.5}$$

where $p(t|u)$ is the probability of word t_j be the tag of user. $P(t|I_j)$ means the probability of word t_i is interest I_j. So we can break $P(t|I_j)$ into $P(t|u_i)$ and $P(u_i|I_j)$. Because $P(I_j|u)$ and $P(u_i|I_j)$ has learnt during process in Sect. 43.2.2.1, so these are known variables. Then what we must focus on is to learn $P(t|u_i)$, which is the probability of a word be tag of a user.

It seems a recursive routine that we have to figure out $p(t|u_i)$ first in order to get $p(t|u)$. However we can define $p(t|u_i)$ as the "basic tag" probability distribution of all the followers, which not including the followed user's probability $p(t|u)$. So $p(t|u_i)$ is absolute different with $p(t|u)$, we can other approaches to generate "basic

tags" for followers. Then use above formulas get followed user's tag. And in this paper, we adopt method introduced in *Latent dirichlet allocation for tag recommendation* [8] to generate basic tags of followers.

43.3 Experiments

In this section, we will illustrate the efficacy of our approach with experiments on real data that crawled from Sina Weibo. It shows that our approach outperforms with all the baseline methods. We will discuss the experiment in detail.

43.3.1 Datasets

We chose data from Sina Weibo, which is the most popular micro blogging service in China. You can follow anyone you like, such as you favorite movies starts, singers. And also you can be followed by anyone found of you. In our experiments, we made a crawler to crawl through the network based on then open platform of Sina Weibo. In order to get the text associated with each user, we collected each user's tweets from April to October in 2012.

43.3.2 Baselines

We chose two classic tag recommendation methods: TFIDF and LDA-Tag Recommendation.

1. *TFIDF* is a simple method to obtain tags from text associated with entities. It treats such text as documents and the use TFIDF method to score all the words, then choose most rating words as the tags for entities.
2. *LDA-Tag Recommendation* treats each tweets of user as different documents [8]. Then use LDA to infer topics-user probabilities and words-topics probabilities. So tags can also be generated with most rating words.

43.3.3 Evaluation

In this experiment, we set 100 as the number of interests of user. The performance of different algorithms is evaluated by the average fraction of wins. There are two views to present the results: followers count and tweets count. From followers count view, we separated users into 8 groups with followers count [0–100], [100–1000], etc. This grouping is to test the performance of algorithm on different popularities. From tweets count view, we separated users into 7 groups with tweets

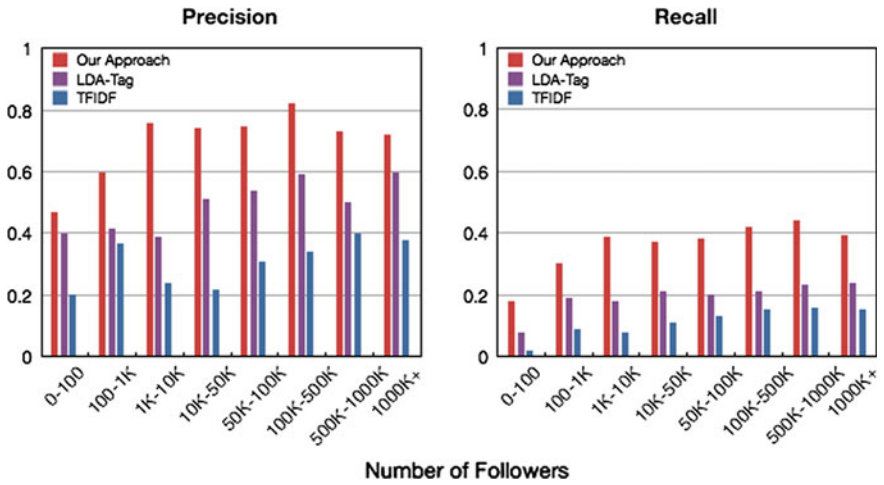

Fig. 43.3 Precision and recall grouped by number of followers

count [0–100], [100–200], etc. This grouping is to test the performance on different frequencies of user on social sites. Then we randomly select 50 users from each group as samples.

Then we showed our result to two annotators for each approach. The annotators are asked to pick the approach with the best tags-set. For each group, we report the average fraction of wins. Figure 43.3 present the precision and recall grouped by followers count for each approach. Figure 43.4 present the precision and recall grouped by tweets count.

It is clear that our approach outperforms all the baseline approaches as expected. However, there is a fall of precision when the count of followers is greater than 500 K in Fig. 43.3. Because it is not easy to determine the actual interest attracted to be followed for users with too much followers. Someone followed the user may just because his friends followed him or the user is so famous that to be followed without obvious interest in common. Figure 43.4 shows us that there is no much difference while taking our approach on different tweet count, while baseline approaches become more precise as the count increasing. It is because our approach is based on the social graph not the text of user, so it's much more stable when the quality of text itself is unpredictable.

43.4 Conclusion

In this paper, we investigate the use of Latent Dirichlet Allocation on social graph to recommend tags. After experimenting, it turns out that our approach outperformances TFIDF and LDA-Tag Recommendation approaches. Our approach not

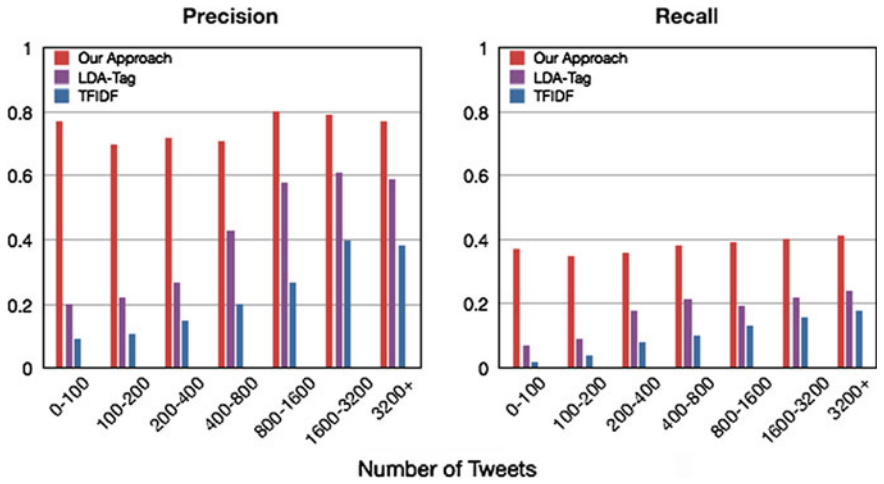

Fig. 43.4 Precision and recall grouped by number of tweets

only recommends more precise and appropriate tags for users, but also works more stably on the range of text qualities. The most contribution of this paper is to combine the latent value of social graph with LDA, archiving better tags recommending.

References

1. Scott J (1988) Social network analysis. Sociology 22:109. doi:10.1177/0038038588022001007
2. Li X, Guo L, Zhao YE (2008) Tag-based social interest discovery. In: Proceeding of the 17th international conference on World Wide Web (WWW'08), pp 675–684
3. Golder S, Huberman BA (2005) The structure of collaborative tagging systems. J Inf Sci 32(2):198–208
4. Marlow C, Naaman M, Boyd D, Davis M (2006) Ht06, taggingpaper, taxonomy, flickr, academic article, to read. In Proceedings of ACM HYPERTEXT'06
5. Basile P, Gendarmi D, Lanubile F, Semeraro G (2007) Recommending smart tags in a social bookmarking system. In: Bridging the gap between Semantic Web and Web 2.0 (SemNet 2007)
6. Matsuo Y, Ishizuka M (2002) Keyword extraction from a document using word co-occurrence statistical information. Trans Jpn Soc Artif Intell 17(3):217–223
7. Jäschke R, Marinho L, Hotho A, Schmidt-Thieme L, Stumme G (2007) Tag recommendations in folksonomies. In: Proceedings of PKDD 2007
8. Krestel R, Frakhauser P, Nejdl W (2009) Latent dirichlet allocation for tag recommendation. In: Proceedings of the third ACM conference on recommender systems, pp 61–68
9. Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res 3:993–1022
10. Golder S, Huberman BA (2006) Usage patterns of collaborative tagging systems. J Inf Sci 32(2):198–208

11. Wallach HM (2006) Topic modeling: beyond bag-of-words. In: Proceedings of the 23rd international conference on Machine learning (ICML'06), pp 977–984
12. Roth M, Ben-David A, Flysher G et al (2010) Suggesting friends using the implicit social graph. In: Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining, pp 233–242
13. Viegas FB, Donath J (2004) Social network visualization: can we go beyond the graph? In: Proceedings of the computer supported collaborative work conference (CSCW'04)