Latent Friend Mining from Blog Data

Dou Shen[1], Jian-Tao Sun[2], Qiang Yang[1], Zheng Chen[2]
[1]Department of Computer Science and Engineering
Hong Kong University of Science and Technology, Hong Kong
{dshen, qyang}@cse.ust.hk
[2]Microsoft Research Asia, 49 Zhichun Road, China
{jtsun, zhengc}@microsoft.com

Abstract

The rapid growth of blog (also known as "weblog") data provides a rich resource for social community mining. In this paper, we put forward a novel research problem of mining the latent friends of bloggers based on the contents of their blog entries. Latent friends are defined in this paper as people who share the similar topic distribution in their blogs. These people may not actually know each other, but they have the interest and potential to find each other out. Three approaches are designed for latent friend detection. The first one, called cosine similarity-based method, determines the similarity between bloggers by calculating the cosine similarity between the contents of the blogs. The second approach, known as topic-based method, is based on the discovery of latent topics using a latent topic model and then calculating the similarity at the topic level. The third one is two-level similarity-based, which is conducted in two stages. In the first stage, an existing topic hierarchy is exploited to build a topic distribution for a blogger. Then, in the second stage, a detailed similarity comparison is conducted for bloggers that are close in interest to each other which are discovered in the first stage. Our experimental results show that both the topic-based and two-level similarity-based methods work well, and the last approach performs much better than the first two. In this paper, we give a detailed analysis of the advantages and disadvantages of different approaches.

1. Introduction

Web communities have risen rapidly in recent years with benefits for different types of users. For individuals, the Web community helps the users in finding friends of similar interests, providing timely help and allowing them to share interests with each other. For commercial advertisers, they can exploit the Web community to find out what the users are interested in, in order to focus their targets. It would be straightforward to discover the Web community if we had the detailed and up-to-date profiles of the relations among Web users. However, it is not easy to obtain and maintain the profiles manually. Therefore, the automatic approaches in mining users' relationship are badly needed.

In the past decades, much research has been conducted with an aim to discover the relationships among people. An example is the well studied social network analysis (SNA) [22, 4, 29]. However, most SNA approaches only model the explicit links from one entity to another, which limits the possibility of discovering latent friends. In fact, people with no links can still be friends potentially, as long as they share the same interests. Our goal in this research is to develop methods for mining the potential relationship among people, and we call this problem "latent friend detection".

Besides the traditional SNA, some other research work based on homepages and emails can also be used to mine relationship among people [29, 19]. Such work cannot completely solve the latent friend detection problem due to the characteristics of the source data. In fact, to find out latent friends, the source data is as important as the mining approaches. The data should satisfy at least three conditions: (1) The data should be extensive in scale, which covers a sufficiently large number of Web users; (2) The data should be available without interfering with people's privacy; For example, it is possible to mine latent relationship from email data, however, Web users are reluctant to provide their email data due to privacy consideration; (3) The data should reflect the Web user's up-to-date interest. Take the Web homepages as an example. Although most people may include their basic information on the homepages, such information often does not reflect their true and complete interests well. As an example, a computer science professor may list the research papers on his Web page, but he may not discuss his interest in poetry.

In this paper, we use a new type of data - blog or weblog - for the latent friend mining problem. After the term "blog" was coined in 1997, the blog as a latent social community has risen rapidly in popularity. More and more Web providers including MSN (http://spaces.live.com/) and Google (http://www.blogger.com) provide free space for bloggers. Recently, there appeared some papers on blog analysis, most of which focus on analyzing the usage, style, development of blogs [9, 11, 33]. There are also some research papers on community mining [15, 12], finding important bloggers [24], and blog entries ranking [5]. Most of these papers are based on the analysis of the link structure among blogs. Several other papers which analyzed the contents of blogs considered each blog entry as an individual object [8, 2, 6, 21]. However, they do not consider a user's different entries as a coherent unit. In this paper, we address these problems for the discovery of latent friends of bloggers based on the contents of their blog data. Our task is challenging due to the following reasons: (1) A blogger usually posts more than one entry in his/her blog space. All the entries together reflect the blogger's interests. (2) A blogger's interests are usually distributed over a certain range of topics. (3) The contents of the entries posted by a blogger are a function of time.

Three approaches are put forward in this paper. The first approach is to determine the similarity between bloggers by calculating the cosine similarity between the contents of their blogs at the literal level. This is the cosine similarity-based approach. The second approach is topic-based. It is based on a probabilistic topic model which can model the topic distribution on a blogger's entries. In this approach, we determine two bloggers' similarity by computing the similarity between the topic distributions of on their blogs. Our third approach is two-level similarity-based, which applies a two-stage procedure. In the first stage, the entries from a blogger are classified according to an existing topic hierarchy. Based on the classification results, we can calculate the "coarse" similarity among bloggers at the topic level. After that, we further compute the "finer" similarity between candidate friends discovered in the first stage at the literal level. In this stage, we can take the published time into consideration, since the time can help us detect the evolution of a bloggers' interest.

The contributions of this paper can be summarized as follows: (1) We put forward a new research problem of finding latent friends from blog data; (2) A novel two-level similarity-based approach is proposed to solve the problem effectively and efficiently, which takes into account the time related content information as well as the topic-distribution information.

We describe the related work in Section 2. Section 3 presents the problem statement. Section 4 provides our solutions and we verify the solutions empirically in Section 5. In Section 6, we conclude our work and put forward the directions of the future work.

2. Related Work

Social Network Analysis (SNA) became a hot research topic after the seminal work by Milgram [22]. SNA is the study of mathematical models for relationships among entities such as people, organizations and groups in a social network. The relationships can be various. For example, they can be friendship, business relationship, and common interest relationship. A social network is often modeled by a graph, where the nodes represent the entities, and an edge between two nodes indicates that a direct relationship exists between the two entities. Some typical problems in SNA include discovering groups of individuals sharing the same properties [29] and evaluating the importance of individuals [4]. Previously, the research in the field of SNA has emphasized binary interaction data, with direct and/or weighted edges [18] and focused almost exclusively on very small networks, typically, in the low tens of entities[34]. Moreover, only considering the connectivity properties of networks without leveraging the information of the entities limits the application of SNA.

With the popularity of Internet, more and more data is available for SNA study. The referralWeb project mined social networks from a wide variety of publicly-available online information [14]. In this project, Kautz et al. modeled social networks statically as graphs and study various aspects of their performance, such as the accuracy of the referrals, or the distance between a referrer and a questioner. Some other researchers mined the relationship from email logs [29]. Adamic and Adar tried to discover the social interactions between people from the information on their homepages [1]. As we can see, homepages and emails are widely used to discover relationships. However, they are not proper for the latent friend detection problem due to their limitations as shown in the introduction part. Therefore, in this paper, we propose to mine the relationships from blog data, which are abundant, open on the Internet and contain much information about Web users' interests.

In [17], Liben-Nowell and Kleinberg proposed a link prediction problem to infer which new interactions among the members in a social network are likely to occur. Similar to their work, we also aim at finding potential interactions among people. However, we focus on leveraging the contents of blogs in an unsupervised manner, without requiring an existing social network as in [17].

Since we rely on the contents of blogs for friend detection, another bunch of research work on topic model (TM) is also related [3, 10]. Some variations of topic model have been developed and used to mine relationships between people based on the text written by people. By extending Latent Dirichlet Allocation (LDA) [3], [27] proposed a generative model named author-topic model which simultaneously models the document contents and the interests of authors. In that paper, each author is associated with a multinomial distribution over topics and each topic is associated with a multinomial distribution over words. By defining the distance between authors i and j as the symmetric Kullback-Leibler (KL) divergence between the topics distribution conditioned on each of the authors, their method can find similar pairs of authors. In our work, since each entry is written by a single blogger, we can backtrack to use LDA for latent friend detection. In [19], Andrew McCallnum et al. proposed Author-Receipt-Topic model which learns the topic distribution and the role of entities simultaneously based on the direction-sensitive messages sent between entities. However, this model is specific to model messages with send-and-receipt relationships such as email data.

Some research work has been conducted on community mining from the blog data. Kumar et al. studied the burst of blogspace [15]. They examined 25,000 blog sites and 750,000 links to the sites. They focused on clusters of blogs connected via hyperlinks named blogspaces and investigated the extraction of blog communities and the evolution of the communities. In [12], Kazunari Ishida proposed to discover latent blog communities of people who are talking about similar topics by looking for connected sets of links (connected bipartite graphs). Shinsuke Nakajima worked on discovering important bloggers based on analyzing blog threads where a blog thread is a set of entries connected to each other via hyperlinks [24]. All of the above work on blog data relies on the hyperlinks in the blog data. However, as Ko Fujimora et al. reported, of the 9,280,000 blog posts they collected, 1,520,000 (16.3%) have one or more hyperlinks. Only 116,000 entries (1.25%) are linked to other blogs [5]. The sparseness of links among blogs will greatly limit the above approaches. Therefore, in our paper, we make use of the contents of the blogs instead of the links. We plan to consider the links and contents simultaneously in our future work.

As we calculate the similarity among bloggers based on the textual information contained in bloggers' entries, our work is related to the conventional document clustering and classification problems [32, 35, 13]. However, in document clustering and classification, each document is a coherent entity though it may relate to more than one topic. In our problem, each blogger is represented by a set of entries and the similarity between two bloggers cannot be regarded as a simple summation of the pair similarity of the basic entries.

3. PROBLEM STATEMENT

Our goal in this paper is to find latent friends from the blog data. In this section, we will describe the concept of latent friends as well as the blog data.

The concept of latent friends is compared with that in the traditional SNA. In SNA, the friendship between people is constructed through the direct or indirect links between them. In our approach, the friendship relation is discovered among people without considering any kind of existing links between them. Instead, it is based on the similar interests between people where the interests of people are reflected by their writings. Therefore, this new kind of relationship is latent in nature. The definition of latent friend is given as follows:

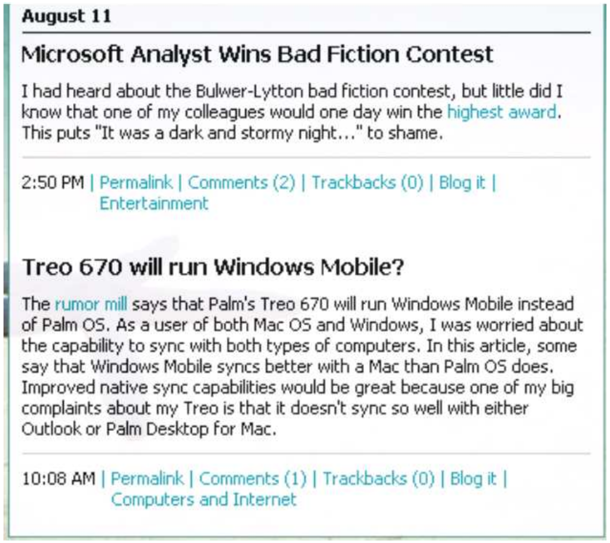
Latent friends of a person refer to the persons who share the similar interests with the target person while the interests are reflected by the topics of their writings.

By this definition, latent friends may not know each other, and may not have direct links to each other's writings. We choose blog data as the source data for latent friends detection, because such data satisfy the three conditions we set forward in the Introduction section. As shown in the Web site Blogger , "a blog is your easy-to-use web site, where you can quickly post thoughts, interact with people, and more". More specific description is also given: "a blog gives you your own voice on the web. It's a place to collect and share things that you find interesting- whether it's your political commentary, a personal diary, or links to web sites you want to remember." From the description of the functions of blog, we can claim that the contents posted by the bloggers are expected to reflect their interests. Therefore, it is reliable to mine friendship based on the blog data.

The primitive entity of the blog data is a blog entry, as shown in Figure 1. The typical elements in an entry include the title, permalink, post time, comments and category. Permalink refers to the unique URL to access the entry. The category information is not obligatory and many entries do not provide the category information as shown in Figure 1. What's more, the category structures across different blog-hosting sites may be different. "Comments" records the comments from other people about the post. Since comments include the interaction between people, it may provide valuable hints for discovering friendships. In this paper, we just consider the content of the comments without utilizing the information about links between bloggers implied in the comments.

Besides the entries, bloggers give their basic information as well as much interesting information on the blogs. Take MSN spaces as an example, the bloggers may put their favorite songs, sports, pictures on the blogs. Bloggers can also put the hyperlinks to their friends or any other bloggers they feel interested in. Therefore, we can see that blog data provides a good resource for data mining which contains not only the links but also informative contents which reflect the author's specific interests. As we mentioned in section 2, we leave out the links in this paper because of their sparseness and noisy nature.

After making clear the concept of latent friend and blog data, we can interpret our problem more accurately. We pro-



August 11

Microsoft Analyst Wins Bad Fiction Contest

I had heard about the Bulwer-Lytton bad fiction contest, but little did I know that one of my colleagues would one day win the highest award. This puts "It was a dark and stormy night..." to shame.

2:50 PM | Permalink | Comments (2) | Trackbacks (0) | Blog it | Entertainment

Treo 670 will run Windows Mobile?

The rumor mill says that Palm's Treo 670 will run Windows Mobile instead of Palm OS. As a user of both Mac OS and Windows, I was worried about the capability to sync with both types of computers. In this article, some say that Windows Mobile syncs better with a Mac than Palm OS does. Improved native sync capabilities would be great because one of my big complaints about my Treo is that it doesn't sync so well with either Outlook or Palm Desktop for Mac.

10:08 AM | Permalink | Comments (1) | Trackbacks (0) | Blog it | Computers and Internet

Figure 1. A piece of the blog data.

pose to determine the relationships between bloggers only based on the contents of the entries they post. A blogger is regarded as a "latent" friend of another blogger so long as they share the similar interests, no matter they known each other or not. The similarity of two person's interests is measured in term of the similarity between the distributions of the topics contained in the blogs. The latent friend detection can be regarded as a kind of SNA where we focus on finding the latent friends for a single person. In fact, it is not hard to extend our approach to find out communities who share the similar interests.

4. Our Approaches

4.1. Cosine Similarity-based Method

A straightforward solution for the latent friend detection problem is to calculate the similarity between the contents of the entries from two bloggers. The more similar between the entries, the more probable the two bloggers share the same interests. Now we introduce how to represent the bloggers and then the way to compute the similarity.

As shown in Section 3, there are several fields in each entry, such as title, body and comments from other people. All these fields reflect the topics of the entry from different aspects. Therefore, it is necessary to combine them together. During the combination, we need to determine the weight for each field since different fields play different roles for reflecting the topic. The problem of weighting schema has been well studied in Web page classification [26]. We can study the importance of the above fields in a similar way. In this paper, we give equal weight to each field for simplicity. After stemming, stop-word removal and feature selection,

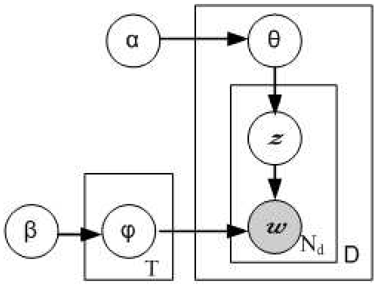


Figure 2. Latent Dirichlet Allocation.

we obtain a vector representing each entry under the Vector Space Model (VSM). Terms in the vector are statistically weighted using the term frequency (TF). In order to calculate the similarity between the bloggers, we can collect the text from all the entries of each blogger and construct a vector in the above way for each blogger. After that we use the cosine value of two vectors to measure the similarity, which is defined as follows:

$$S(i, j) = \frac{\sum_k n_{ik} n_{jk}}{\sqrt{\sum_k n_{ik}^2 \sum_k n_{jk}^2}} \qquad (1)$$

where n_{ik} is the term frequency of term k in blogger i's blog. Given a blogger i, after calculating the similarity between him/her and all other bloggers, we can sort the bloggers according to the similarity. Then the top bloggers in the list can be recommended as blogger i's latent friends. The above approach gives a baseline for the latent friend detection problem. In the following sections, we will put forward two more sophisticated approaches to determine the similarity among bloggers.

4.2. Topic Model based Method

In statistical language processing, one common way of modeling a document is to treat a document as a probabilistic mixture of topics where each topic is a probability distribution over words. Latent Dirichlet Allocation (LDA [3]) is one such model. Recently, some variations of LDA have been used to mine relationships between people based on the texts written by them [27, 19]. In this paper, we will apply a variant of LDA together with the idea of calculating similarity between authors proposed in [27] to solve the latent friend detection problem.

In LDA, the generation of a document collection is modeled as a three-step process. First, for each document, a distribution over topics is sampled from a Dirichlet distribution. Second, for each word in the document, a single topic is chosen according to this distribution. Finally, each word is sampled from a multinomial distribution over words specific to the sampled topic. This generative process is illustrated in Figure 2. In Figure 2, φ denotes the matrix of topic distributions, with a multinomial distribution over V vocabulary items for each of T topics being drawn independently from a symmetric Dirichlet(β) prior[1]. θ is the matrix of document-specific mixture weights for these T topics, each being drawn independently from a symmetric Dirichlet(α) prior. For each word w, z denotes the topic responsible for generating that word, drawn from the distribution for that document, and w is drawn from the topic distribution φ corresponding to z. Estimating φ and θ provides information about the topics that participate in a corpus and the weights of those topics in each document respectively. A variety of algorithms have been used to estimate these parameters, including variational inference [3], expectation propagation [23], and Gibbs sampling [27]. In this paper, we apply Gibbs sampling.

Rosen-Zvi et al. claim in [27] that the above topic model provides no explicit information about the interests of authors. The reason is that authors may produce several documents - often with co-authors - and it is consequently unclear how the topics used in these documents might be used to describe the interests of the authors. Therefore, they propose an author-topic model to model an author's interests explicitly. However, in our problem, the blogs from each blogger are written by the blogger alone. Therefore it is reasonable to describe the interests of the bloggers in terms of the topics included in the entries written by the bloggers. By applying LDA, we can obtain the topic distribution on each entry, and then the blogger's interest can be regarded as a mixture of the distributions. To simplify the procedure, we can concatenate the entries to construct a virtual document. The topic distribution on the virtual document can be used to represent the blogger's interest.

Similar to [27], we define the distance between blogger i and blogger j as the symmetric KL divergence between the topic distributions conditioned on each blogger:

$$D(i,j) = \sum_{t=1}^{T} \left[\theta_{it} log \frac{\theta_{it}}{\theta_{jt}} + \theta_{jt} log \frac{\theta_{jt}}{\theta_{it}} \right] \tag{2}$$

where T is the number of topics; θ_{it} is the probability of topic t conditioned on blogger i.

4.3. Two-Level Similarity-based Method

Although the topic model is well-established on statistics, it is not easy to learn the parameters even in an approximation way. What's more, it is not straightforward to extend this model to take the dimension of time into consideration. In fact, time is an important feature in our problem since the interests of a blogger can evolve with time. Another reason for us to seek a new way for our problem is that we do not need to estimate the complex parameters for a generative model since we just need to find out the latent friends for a certain blogger based on the contents of their blogs. This problem is more likely to be solved with a discriminative approach instead of a generative approach. Thus, in this paper, we propose a two-stage discriminative approach. In the first stage, the topics contained in the entries of each blogger are predicted according to a predefined topic hierarchy. Based on the topics we obtained, a "coarse" similarity is calculated between bloggers at the topic level which can produce a tentative latent friend-list for each blogger. In the second stage, we calculate the "finer" similarity between bloggers at the literal level by taking the real content of each blog into consideration, where, in particular, the temporal feature associated with the blog entries is considered. With the finer similarity, the tentative list generated in the first stage is reordered. The two stages are detailed in the following sections.

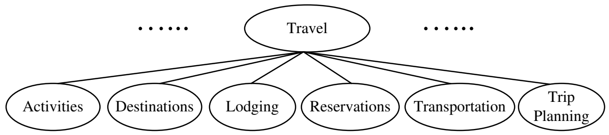


Figure 3. Parts of an existing topic hierarchy.

4.3.1 Stage I: Coarse Similarity at Topic Level

Unlike the topic model which treats the topics as latent variables, we can use an existing topic hierarchy. Such a topic hierarchy can be obtained from the online Web taxonomy, such as those provided by Looksmart[2] and ODP [3] . These Web taxonomies have shown to be effective for providing background knowledges [7, 30]. After obtaining the topic hierarchy, we can treat the problem of topic prediction for an entry as a text classification problem. This greatly simplifies the problem. The detailed process is shown below. As we mentioned in Section 3, although some entries are labeled by the blogger, most posts do not have labels. Moreover, the topic hierarchy provided by different blog-hosting sites may be different. Therefore, our topic prediction step through text classification is necessary.

Training Classifiers: Text classification is a well studied field. There are many kinds of classification algorithms such as Naïve Bayes [16, 20], k-nearest neighbor [35], support vector machines (SVM) [13], boosting [28] and rule learning algorithms [31]. If we extract the topic hierarchy

[1] A k-dimensional Dirichlet random variable φ has the following probability density: $p(\varphi|\beta) = \frac{\Gamma(\sum_i^K \beta_i)}{\prod_i^K \Gamma(\beta_i)} \varphi_1^{\beta_1 - 1} ... \varphi_K^{\beta_k - 1}$ where the parameter β is a k-vector with components $\beta_i > 0$, and where $\Gamma(x)$ is the Gamma function.

[2] http://search.looksmart.com/

[3] http://dmoz.org/

from an online Web taxonomy, we can also obtain the training data for each topic at the same time. With the training data and any proper classification algorithm, we can train a text classifier. In this paper, the topic hierarchy is extracted from Lookmart, which contains the top two levels, with 10 first level and 64 second level topics. Figure 3 shows a part of the topic hierarchy. Among the classification algorithms, we choose SVM because of its high generalization performance when used for text classification task [13]. We use the SVM^{light} software package . A linear kernel is used and the one-against-rest approach is applied for the multi-class case. Information Gain (IG) is used for feature selection [36]. Some standard preprocessing steps are employed, such as stemming and stopwords removal.

Topic Prediction for Blogs: In our approach, we treat the topic prediction of blogs as a text classification problem. To classify an entry with the trained classifier, we need to construct a vector representation of the entry in the way as shown in Section 4.1. After that, we feed each entry to the classifiers and obtain the classification results. In the results, each topic is assigned a score indicating the confidence that the entry is related to the corresponding topic.

Topic Vector Construction: Since the topic hierarchy is fixed, we can represent the topics in a blogger's entries by a topic vector b. There are two ways to build the vector. In the first way, we take the top n topics for each entry. n may be larger than one since an entry can contain multiple topics. n should be determined according to the diversity of the topics. Then we count the number of occurrences of topic i among the blog entries and set it as b_i. In the second way, we add the confidence of each topic among all the entries together. That is: $b_i = \sum_k p_{ik}$ where p_{ik} denotes the confidence that entry k contains topic i, which is measured by the probability that can be obtained from the classification results by SVM through the method given in [25]. The second method is smoother than the first one. However, it may cause some problems, since the accumulation of low confidences about a certain topic which may be noisy among all entries can exceed the accumulated confidence of a topic which is certain in a few entries.

Coarse Similarity Calculation: After obtaining the topic vector for each blogger, we can calculate the similarity between two bloggers based on the distribution of topics in their blogs. Two approaches can be employed in this step. The simple one is cosine similarity which is defined in equation (1). Another way is to normalize the topic vector for each blogger such that the vector represents the empirical probability distribution of topics among the blogs. Then the similarity can be calculated according to equation (2).

After we get the similarity between bloggers, it is straightforward to recommend the latent friends to a blogger. However, the results may not be satisfying in that the similarity is based on the topic distribution. We notice that the contents of two blogs with the same topic may be much different. For example, two blogs are both about "Lifestyle\Food". But one is about "Chinese food" and another is about "Western food". Therefore, we can say that this kind of similarity provide only a "coarse" measurement of the relationship between bloggers which can help narrow down the candidates of latent friends for a blogger. To make the "latent" friend more precise, we need to find a "finer" measurement to decide the final list of latent friends.

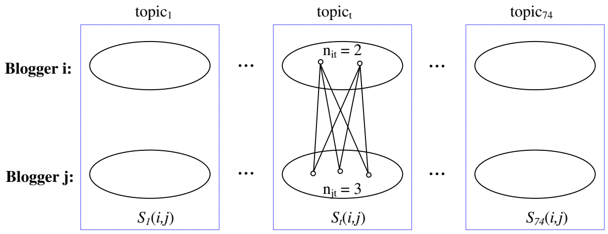


Figure 4. Illustration of finer similarity calculation.

4.3.2 Stage II: Finer Similarity at Literal Level

To find a "finer" similarity measurement, one way is to define a more detailed topic hierarchy. However, more detailed topic hierarchy will certainly improve the complexity of the solution. Moreover, we cannot avoid the above problem no matter how detailed the topic hierarchy is. Therefore, we rely on the actual content of the blogs to further define the similarity between bloggers at the literal level.

Through the first stage, we can distribute the entries of each blogger into the topic hierarchy. As shown in Figure 4, two and three entries are assigned to topic t for blogger i and j respectively. Then, we can calculate the finer similarity between blogger i and blogger j in the following way:

We calculate the similarity between blogger i and j on topic t firstly, as shown in equation (3). E_{it} and E_{jt} denote the set of entries of blogger i and j which are assigned to topic t. n_{it} and n_{jt} represents the size of E_{it} and E_{jt}. $s(entry_k, entry_l)$ is the cosine value between vectors of $entry_k$ and $entry_l$. $|m(k) - m(l)|$ is the difference of published time of $entry_k$ and $entry_l$ in terms of month. λ reflects our sensitivity of time difference. If $\lambda = 0$, we do not care about the time difference. The larger the value of λ is, the more important is the role of time. The reason for considering time is that people's interests may change with the time. Therefore, similar entries from two bloggers with a large difference at the post time are not sufficient to indicate that the two bloggers still have the same interest.

$$S_t(i,j) = \frac{\sum\limits_{k \in E_{it}} \sum\limits_{l \in E_{jt}} s(entry_k, entry_l) * e^{-\lambda |m(k) - m(l)|}}{n_{it} * n_{jt}} \tag{3}$$

Then we take the average of the similarity between bloggers i and j on all topics as shown in equation (4). We weight the similarity on each topic by the number of blog entries on that topic to remove the effect of noise.

$$S(i,j) = \frac{\sum_t S_t(i,j) * (n_{it} + n_{jt})}{\sum_t n_{it} + n_{jt}} \quad (4)$$

After obtaining the similarity between any two bloggers in stage II, we can recommend latent friends to any blogger as in the previous approaches.

The advantages of the two-level similarity-based method are shown below:

- The proposed strategy is aimed at reducing the time complexity of computation;

- The strategy is easy to be extended to find friends with a specific interest. For example, in order to find the latent friends for a blogger who shares the same interest on "Travel", we just need to calculate the similarity between blogs which belong to the topic "Travel".

5. Experiments

In this section, we demonstrate the utility of our three methods empirically. We show that the last method, two-level similarity-based method, is superior to the first two.

5.1. Dataset

To train the classifier for two-level similarity-based approach, we crawled 153,019 pages from the LookSmart Web directory, which distribute among the top two level categories (74 categories in total) on the LookSmart Website.

The blog data we used in this paper are crawled from MSN Spaces at the mid of August, 2005. We collected more than 10,000,000 blog URLs in total. Some of these blogs cannot be crawled due to the access constraint. Among the crawled blogs, we randomly select 15,000 ones written in English. Among the selected collection, the earliest entry was posted in December 2004. The average number of entries per blogger is 13.45 and the average size of each entry is 935 byte. To increase the accuracy of the detection, the bloggers with less than 10 entries are removed which results in a set of 9,918 bloggers.

5.2. Evaluation

Though the blog data provide valuable resource for data mining and machine learning, there are no public benchmark data for research yet. Therefore, in this paper, we asked three human evaluators to judge the performance of our approaches. After applying our approaches, we can recommend a list of latent friends for each blogger in the pool of the 9,918 bloggers. Then, we randomly selected 30 bloggers as our test data. Since our problem is actually a ranking problem, we can use the classical evaluation method adopted in Information Retrieval. We use precision (P) at top N results to measure the performance:

$$P@N = \frac{\#Latent\ Friends}{N} \quad (5)$$

where $\#Latent\ Friends$ is the number of manually tagged correct latent friends. That is, after generating an ordered list of latent friends for a blogger A, we ask the three evaluators to read the blogs of A and assume that they are blogger A. Then they read the blogs of the top N bloggers in the ordered list to see whether they want to make "friends" of them. The three evaluators work separately without knowing how our detection algorithms work. The averaged results from them as well as the standard deviations are reported in this paper. It is clear that the evaluation approach is somewhat subjective. However, it is not easy to construct an objective test dataset for our problem at the current stage. In the future, we would invite some bloggers to judge the friend detection results for them by themselves.

An alternative way is to take the friend lists on the blogs provided by the bloggers as the ground truth. However, when we analyze the blogs, we find that many persons on the friend list of a blogger do not share the same interests with the blogger at all. The reasons for adding them to the friend list are various. For example, they are once classmates or colleagues. Therefore, in this paper, we do not adopt this method.

5.3. Parameters Setting

There are several parameters in our proposed approaches. They are restated as follows for clarity and their values are set empirically.

To apply topic model, we need to predetermine the number of topics. In this paper, we set the number equal to 74 which is same as the number of topics used in the two-level similarity-based approach.

In the first stage of the two-level similarity-based approach, one way to construct the topic vector is by counting the number of each topic among the top n topics of each entry. In this paper, n is set in the following way: among the top 3 topics for each entry, if the gap between the confidences of the i^{th} and $i+1^{th}$ topic is too large (say, in this paper, the former is two times greater than the later), we will set n equal to i; otherwise, we set n equal to 3.

We mentioned two ways to calculate the coarse similarity in the first stage of the two-level similarity-based approach: one is cosine similarity and the other is KL dis-

	N=10	N=20	N=30
Baseline	0.463(0.035)	0.411(0.041)	0.337(0.053)
TM	0.662(0.029)	0.557(0.037)	0.459(0.051)
CS_C	0.560(0.026)	0.512(0.030)	0.435(0.041)
CS_S	0.538(0.031)	0.451(0.033)	0.383(0.045)
Two-Stage	0.816(0.022)	0.654(0.029)	0.558(0.037)

Table 1. P@N of different algorithms

tance. In the following experiments, we use the cosine similarity for simplicity.

As we described, the two-level similarity-based approach can take the time information into consideration when computing the finer similarity through the parameter λ . We set λ equal to 0.5 in this paper. In fact, the value of λ totally depends on Web users' goals. In order to obtain friends with the same interests during the same period, we can increase the value of λ , otherwise, we can decrease it.

For a target blogger, the second stage of the two-level similarity approach works on a subset of the bloggers decided by the first stage. In this paper, the subset is constructed by collecting the top 10% bloggers according to the coarse similarity with the target blogger.

5.4. Experimental Results

Table 1 shows the performance of different algorithms. There are two numbers in each cell. The first number is the averaged precision from the three evaluators and the second one in parentheses is the standard deviation. In Table 1, "Baseline" represents the cosine similarity-based approach which takes all the entries for each blogger as a whole and calculates the cosine similarity between bloggers. "TM" refers to the probabilistic Topic Model based approach. "CS_C" means that we just apply the first stage in the two-level similarity-based approach to calculate the Coarse Similarities between bloggers and the topic vector of each blogger is constructed by ***C***ounting the top n topics of each entry. "CS_S" is similar to CS_C except that we construct topic vector by ***S***umming up the confidence of each topic across all entries. "Two-Stage" represents the two-level similarity-based approach where we adopt CS_C in the first stage since CS_C is much better than CS_S.

From Table 1 we can see that Two-Stage achieves the best result while Baseline is worst as we expect. The performance of TM is better than that of the approaches only based on the coarse similarity though they work in a similar way in that they all calculate the similarity between bloggers at the topic level. CS_S is not as good as CS_C. The reasons for the different performance are explained below.

The baseline algorithm takes the entries from a blogger as a whole and determines the similarity between two bloggers in terms of the cosine similarity between the bloggers' entries. The main reason for its bad performance is that when the entries are taken as a whole, the small entries will be overwhelmed by the large entries. For example, a blogger is interested in both music and sports. However, he posts several short entries about music but just a long entry about a football competition. Then this blogger tends to be more similar to those whose blogs are mainly about football. The role of the short entries about music is overlooked. Another reason is that the baseline algorithm cannot capture the similarity between bloggers at the topic level as other approaches do. Given the first three bloggers in Figure 5, blogger 1 and blogger 2 share the same topic T_1 represented by the ellipses but different subtopic T_{11} and T_{12} respectively (denoted by the small solid circle). Blogger 3 has the totally different topic represented by the solid triangle. It is possible that not any two among the three bloggers share common terms in their blogs. But by intuition, blogger 1 is more similar to blogger 2 than to blogger 3. In this case, the baseline approach fails to discover the similarity between blogger 1 and blogger 2, but other approaches can.

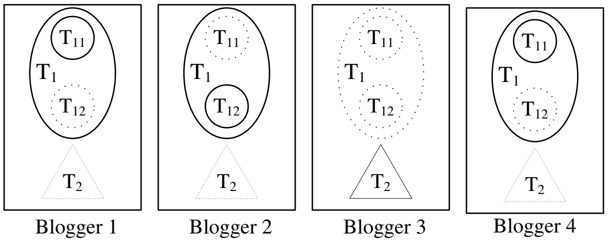


Figure 5. Illustration of the reasons for different performance of the algorithms.

TM works in a similar way to CS_C and CS_S in that they all determine the similarity between bloggers at the topic level, which makes them overcome the second disadvantage of the baseline approach. However they are different in two aspects: 1) TM treats all the entries from a blogger as a whole while CS_C and CS_S treat them individually; 2) the topics in TM are learned automatically from the blogs while they are predefined in CS_C and CS_S. It is clear that the topics discovered by TM are more suitable to reflect the contents of the blogs than the predefined topics in CS_C and CS_S. Therefore TM performs better than CS_C and CS_S when calculating the similarity on the topic level. TM faces the same problem as the baseline approach in that they both treat the entries from a blogger as a whole. However, by calculating the topic distributions, TM works more smoothly than the baseline's literal matching which can alleviate the first disadvantage of the baseline approach as shown above. All these three approaches working at the topic level are sensitive to the granularity of the topics. If the topics are at the granularity like T_1, all the three methods cannot distinguish blogger 1 and blogger 2.

CS_C works better than CS_S as shown in Table 1. The reason is that the small confidence returned by the classifier is not reliable. Given an entry e, the classifier will return the confidence scores that e belongs to each topic. Even for the most unlike topic, the confidence is usually not equal to zero. Therefore, noise will be introduced if we summate the confidence scores of a topic across all the entries. However, if we can develop some better classifiers which could reduce the confidence of unlikely topics, CS_S can provide a more smooth approach than CS_C which is expected to achieve better performance.

The two-level similarity-based approach overcomes all the disadvantages of the above approaches by first calculating the coarse similarity at the topic level and then the finer similarity at the literal level. For example, given blogger 1 in Figure 5, the two-level similarity-based approach can filter out blogger 3 using the coarse similarity and then put blogger 2 after blogger 4 according to finer similarity. As we have discussed, the TM approach could learn the topics in the blogs automatically which is more suitable for describing the blogs than the predefined topic hierarchy. By applying it at the coarse similarity calculation stage, we can expect to improve the performance of the two-stage approach. However we prefer the predefined topic hierarchy due to the following two reasons: 1) TM is a generative model and it needs enough data to learn the parameters. The parameters are usually hard to be estimated even by approximate strategies. 2) The estimated topics are specific to the training data collection which is not as general as the predefined topics used in the two-stage approach. Therefore, we cannot apply the estimated models on new bloggers.

5.5. Case Study

Table 2 shows the top 10 bloggers returned by the two-level similarity-based approach for the blogger worshippersplace[4] . This is a relatively simple example in that the entries in worshippersplace's blog are highly concentrated on the belief of Christmas. We choose it for the case study to simplify the judgment about the performance of our proposed method. By checking the returned bloggers, we can see that all of them have the belief of Christmas. Three of them (shown in bold font) joined the "Awesome MSN Spaces" [5] as worshippersplace did. That space consists of only about 110 members with 6 members appearing in our used data set. Such an observation provides the extra evidence for the fact that they do share the same interests and our method is effective.

Rank	Name	Similarity
1	**mvanburen**	0.58
2	plaintruth	0.50
3	thespirit	0.38
4	joelblog	0.37
5	rasrobinson	0.37
6	hyperion1984	0.36
7	lutheranmatt	0.33
8	**fireheart31**	0.33
9	seangoodwin	0.31
10	**grimeaus**	0.31

Table 2. Top 10 bloggers returned by the two-level similarity-based approach for the blogger "worshippersplace"

6. Conclusion and Future Works

We propose to exploit the contents of blogs for discovering latent friends of bloggers which extend the field of community mining. Latent friends are defined in this paper as people who share the same topic distribution in their blogs. These people may not actually know each other, but they have the interest and potential to find each other out. A straightforward approach based on the cosine similarity between the contents of the entries from bloggers is proposed as the baseline. Two more sophisticated approaches are also put forward. One of them is based on the discovery of latent topics using a latent topic model. In this method the similarity between two bloggers is measured by the KL divergence between the bloggers' topic distributions along the latent topics. Another approach is based on two-level similarity which is conducted through two stages. In the first stage, an existing topic hierarchy is exploited to build a topic distribution for a blogger and the coarse similarity between bloggers is calculated at the topic level. Then, in the second stage, a detailed similarity comparison at the literal level is conducted for bloggers that are judged to be close in interest to each other at the first stage. The experimental results on the blog data collected from MSN Spaces show that the two-level similarity-based approach can achieve much better performance than the other two approaches. Besides the empirical results, the analysis of the advantages and disadvantages of all the approaches are also given in this paper.

Among the collected blog data used in this paper, each blogger only posted 13.45 entries on average. With more and more entries posted by the bloggers, our approaches are supposed to perform better. To test our proposed approaches on some much larger data sets is part of our future work. Another more fundamental future work is to find some more objective metrics for evaluating the results of latent friend mining, though this problem is reasonably solved

[4] http://spaces.msn.com/members/worshippersplace/

[5] http://usa.ultimatetopsites.com/religion/msnspaces/

in a subjective manner in this paper. With a more objective method of evaluation, we can compare different methods more easily.

7 ACKNOWLEDGMENTS

Dou Shen and Qiang Yang are supported by a grant from NEC (NECLC05/06.EG01). We thank the anonymous reviewers for their useful comments.

References

[1] L. A. Adamic and E. Adar. A friends and neighbors on the web. Technical report, 2002.

[2] P. Avesani, M. Cova, C. Hayes, and P. Massa. Learning contextualised weblog topics. In *WWW2005, 2nd Annual Workshop on the Weblogging Ecosystem*, 2005.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[4] P. Domingos and M. Richardson. Mining the network value of customers. In *SIGKDD*, pages 57–66, 2001.

[5] K. Fujimura, T. Inoue, and M. Sugisaki. The eigenrumor algorithm for ranking blogs. In *WWW2005, 2nd Annual Workshop on the Weblogging Ecosystem*, 2005.

[6] T. Fukuhara, T. Murayama, and T. Nishida. Analyzing concerns of people using weblog articles and real world temporal data. In *WWW2005, 2nd Annual Workshop on the Weblogging Ecosystem*, 2005.

[7] E. Gabrilovich and S. Markovitch. Feature generation for text categorization using world knowledge. In *IJCAI*, pages 1048–1053, 2005.

[8] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *WWW*, pages 491–501, 2004.

[9] S. C. Herring, L. A. Scheidt, S. Bonus, and E. Wright. Bridging the gap: A genre analysis of weblogs. *hicss*, 04, 2004.

[10] T. Hofmann. Probabilistic latent semantic analysis. In *UAI*, 1999.

[11] D. A. Huffaker and S. L. Calvert. Gender, identity, and language use in teenage blogs. *Journal of Computer-Mediated Communication*, 10(2), 2005.

[12] K. Ishida. Extracting latent weblog communities: A partitioning algorithm for bipartite graphs. In *WWW2005, 2nd Annual Workshop on the Weblogging Ecosystem*, 2005.

[13] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In *ECML*, 1998.

[14] H. Kautz, B. Selman, and M. Shah. Referral web: combining social networks and collaborative filtering. *Commun. ACM*, 40(3):63–65, 1997.

[15] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins. On the bursty evolution of blogspace. In *WWW*, pages 568–576, New York, NY, USA, 2003. ACM Press.

[16] D. D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In *ECML*, 1998.

[17] D. Liben-Nowell and J. M. Kleinberg. The link prediction problem for social networks. In *CIKM*, 2003.

[18] F. Lorrain and H. C. White. The structural equivalence of individuals in social networks. *Journal of Mathematical Sociology*, 1:49–80, 1971.

[19] A. McCallum, A. Corrada-Emmanuel, and X. Wang. Topic and role discovery in social networks. In *IJCAI*, 2005.

[20] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classication. In *AAAI-98 Workshop on Learning for Text Categorization*, 1998.

[21] Q. Mei, C. Liu, H. Su, and C. Zhai. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In *WWW*, 2006.

[22] S. Milgram. The small world problem. *Psychology Today*, 2, 1967.

[23] T. P. Minka and J. D. Lafferty. Expectation-propogation for the generative aspect model. In *UAI*, pages 352–359, 2002.

[24] S. Nakajima, J. Tatemura, Y. Hino, Y. Hara, and K. Tanaka. Discovering important bloggers based on analyzing blog threads. In *WWW2005, 2nd Annual Workshop on the Weblogging Ecosystem*, 2005.

[25] J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In A. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, 1999.

[26] D. Riboni. Feature selection for web page classification. In *EURASIA-ICT 2002 Proceedings of the Workshop*, 2002.

[27] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *UAI*, Arlington, Virginia, 2004.

[28] R. E. Schapire and Y. Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.

[29] M. F. Schwartz and D. C. M. Wood. Discovering shared interests using graph analysis. *Commun. ACM*, 36(8):78–89, 1993.

[30] D. Shen, J.-T. Sun, Q. Yang, and Z. Chen. Building bridges for web query classification. In *SIGIR*, 2006.

[31] S. Slattery and M. Craven. Combining statistical and relational methods for learning in hypertext domains. In *ILP*, pages 38–52, 1998.

[32] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. Technical Report 00-034. University of Minnesota, 2002.

[33] G. Thompson. Visual factors in constructing authenticity in weblogs. Paper presented in the Communication and Technology Division, International Communication Association, 2004.

[34] S. Wasserman and K. Faust. Social network analysis: methods and applications. pages Cambridge University Press, Cambridge, UK, 1994.

[35] Y. Yang. An evaluation of statistical approaches to text categorization. *Inf. Retr.*, 1(1-2), 1999.

[36] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *ICML*, 1997.