Social-Network Analysis Using Topic Models

Youngchul Cha
UCLA Computer Science Dept
Los Angeles, CA 90095
youngcha@cs.ucla.edu

Junghoo Cho
UCLA Computer Science Dept
Los Angeles, CA 90095
cho@cs.ucla.edu

ABSTRACT

In this paper, we discuss how we can extend probabilistic topic models to analyze the relationship graph of popular social-network data, so that we can "group" or "label" the edges and nodes in the graph based on their topic similarity. In particular, we first apply the well-known Latent Dirichlet Allocation (LDA) model and its existing variants to the graph-labeling task and argue that the existing models do not handle *popular nodes* (nodes with many incoming edges) in the graph very well. We then propose possible extensions to this model to deal with popular nodes.

Our experiments show that the proposed extensions are very effective in labeling popular nodes, showing significant improvements over the existing methods. Our proposed methods can be used for providing, for instance, more relevant friend recommendations within a social network.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*clustering*; D.2.8 [**Software Engineering**]: Metrics—*performance measures*

General Terms

Experimentation, Algorithms

Keywords

social-network analysis, topic model, Latent Dirichlet Allocation, handling popular nodes

1. INTRODUCTION

Social network services are gaining popularity. A growing number of users use major social network services, such as Facebook and Twitter, to share their thoughts and whereabouts with their friends and followers. On a social network, a user indicates that she wants to get notified of another user's updates by "following" (or making friends with) that user. Then when the followed user "updates her status" (or shares a new thought), all users who follow her are immediately notified. This "follow relationship" among users often looks unorganized and chaotic, because follow relationships are created haphazardly by each individual user and not controlled by a central entity.

In this paper, we explore techniques to provide more structure to this follow relationship (1) by "grouping" the users based on their topic interests, and (2) by "labeling" each follow relationship with the identified topic group. More formally, we consider each user in a social network as a node in a graph and each follow relationship as a directed edge between two nodes. Then our goal is to "group" a set of nodes in the graph based on their topics and "label" each edge in the graph with a topic group number.

Inferring a structure within the social-network relationship graph can be useful for many reasons. For example, a novice user on a social network often encounters the *bootstrapping* problem: discovering relevant users to connect with. To mitigate this problem, social network services may recommend potentially interesting users to new users if they can group users of similar interests and infer why the new user has decided to follow a certain initial set of other users. Similarly, we can identify a small set of "influential" users on a certain topic (say, for marketing and advertising purposes) if we can identify the users' topic interests.

Roughly, we can consider our goal as a clustering (or classification) problem, where many popular solutions such as K-means [15] and DBSCAN [6] exist. These existing methods, however, are not appropriate for our task because they either (1) associate each node with a *single* group (hard clustering) or (2) can associate each node with multiple groups (soft clustering), but require a completely separate method to label edges as well as nodes (since a node may be associated with multiple groups). Given the diversity of interest a user may have, it is too restrictive to associate a user with a single topic group. For example, Kevin Rose, one of the most popular users on Twitter, may belong to the entrepreneur group as he is the founder of Digg.com, but may also belong to the Internet commentator group since he runs a popular Internet podcast. Since many users on social networks are active in more than one community, we believe it is too unrealistic to require that every user should belong to just one group.

In this paper, we apply a well-known probabilistic topic model, called Latent Dirichlet Allocation (LDA), to the follow-relationship graph of the social network, in order to label the nodes and the edges in the graph with (possibly) multiple

SIGIR'12, August 12–16, 2012, Portland, Oregon, USA.

topics. Unfortunately, LDA was developed for labeling *documents* and *words* with topics, (not nodes and edges in a graph) so some of the assumptions on which LDA was built are not applicable to social-network graph data. In particular, the direct application of LDA to our task requires that every node in the graph should be of roughly *equal popularity* and that we should remove nodes of high popularity from the dataset. This is particularly problematic because these popular nodes are really the ones that we want to label accurately; many users are particularly interested in identifying the topic groups of these popular users. Earlier work on the application of the LDA model to social graph [9, 26] has not addressed the handling of popular nodes.

To address the issues arising from popular nodes in the graph, we first explore existing variations of LDA. We then propose our extensions, *two-step labeling* and *threshold noise filtering*, to minimize the labeling noise introduced by popular nodes.

In summary, we make the following contributions in this paper:

- We explore the application of the well-known LDA model and its existing variations for this task and propose two extensions to make LDA suitable for the social-network relationship graph.

- We conduct extensive experiments using a real-world Twitter dataset. Through these experiments, we demonstrate that (1) the application of probabilistic topic models to social-network graphs leads to useful edge/node topic labeling and that (2) our proposed extensions significantly improve the labeling quality over existing methods.

The rest of the paper is organized as follows. In Section 2, we briefly review previous work related to our research. In Section 3, we describe LDA and justify why we use LDA to solve this labeling problem. Then we introduce four different approaches to handle the noise generated by popular users in Section 4. In Section 5, we analyze our approaches using the Twitter dataset. We summarize this paper with our conclusion in Section 6.

2. RELATED WORK

The problem that we are trying to solve can be viewed as clustering in a social network using a topic model. In this section, we briefly review related work in social-network cluster analysis and topic-model-based social-network analysis.

In his seminal work for social-network cluster analysis, Kleinberg [13] proposed Hyperlink-Induced Topics Search (HITS). In this work, he modeled the Web as a huge graph and extracted *hubs* and *authorities* in order to cluster communities on the Web. Recently, Mislove et al. [18] identified communities in Facebook based on the social graph structure and inferred unknown attributes through this community information. The methods they used to cluster communities are based on pruning edges from the whole social graph and adding edges from some seed nodes, both of which are very common and widely used approaches in social-network analysis. However, these approaches produce mutually exclusive groups and cannot support *multiple memberships*, which is important in our scenario where users have a variety of interests.

Topic models, the other class of related work, have also been extended to analyze small social-network data. Though not directly related to social-network analysis, the concept of author/user was initially introduced in the Author-Topic (AT) model [22]. It was used to extract hidden research topics and trends from CiteSeer's abstract corpus. Zhou et al. [27] modified the AT model and proposed the Community User Topic (CUT) model to capture semantic communities. McCallum et al. [16] extended the AT model and proposed the Author-Recipient-Topic (ART) model and the Role-Author-Recipient-Topic (RART) model in order to analyze the Enron e-mail corpus and an academic e-mail network. Pathak et al. [19] modified the ART model and suggested the Community-Author-Recipient-Topic (CART) model similar to the RART model. Besides these members of the AT model family, Wang et al. [24] introduced the Group Topic (GT) model and applied it to voting data from US Senate and the General Assembly of the UN. Mei et al. [17] also introduced a regularized topic modeling framework incorporating a graph structure in the data. Other LDA extensions and probabilistic topic models were also proposed for annotation data analysis [12], chat data analysis [23], tagging data analysis [8], and pairwise data analysis [2]. While most of the LDA approaches introduced above attempted to utilize the authorship information of a given text by adding the author component to the LDA's text generative model, our approach focuses on using only the social part of the data (i.e., the social graph) and is generally applicable to many large social networks.

Perhaps the work by Zhang et al. [26] and by Henderson et al. [9] is closest to our work. In both works, the authors applied LDA to academic social networks. However, their respective focus was quite different from ours. For example, [26] focused on the issue of how to convert the co-authorship information into a graph (e.g., direct co-authorship or indirect co-authorship, and edge weighting scheme based on collaboration frequency). Henderson et al. [9] addressed the issue of a large number of topic clusters generated due to *low popularity* nodes in the network, while our primary focus is the effective clustering of *high popularity* nodes.

3. APPLYING LDA TO SOCIAL-NETWORK ANALYSIS

In this section, we briefly describe LDA. Because LDA evolved from Probabilistic Latent Semantic Indexing (PLSI) [11], we first describe the concept of PLSI and then explain what differentiates LDA from PLSI. We also justify the reason why we use LDA for social-graph mining and discuss some different aspects between the standard LDA and our model.

3.1 Topic Models

Topic models assume that there are latent (hidden) topics behind words in human language. Thus, even though an author uses the word *automobile* in a document and a searcher uses the word *vehicle* in a query, topic models assume that they might have the same concept (topic) *car* in mind. Based on this assumption, topic models provide methods to infer those latent topics from visible words.

PLSI introduced a probabilistic generative model to topic

models. Equation (1) represents its document generation process based on the probabilistic generative model:

$$P(d, w) = P(d)P(w|d) = P(d) \sum_{z \in Z} P(w|z)P(z|d). \quad (1)$$

$P(d, w)$ is the probability of observing a word w in a document d and can be decomposed into the multiplication of $P(d)$, the probability distribution of documents, and $P(w|d)$, the probability distribution of words given a document. This equation describes a word selection for a document, where we first select a document then a word in that document. If we iterate this selection multiple times, we can generate a document and eventually a whole document corpus.

By assuming that there is a latent topic z, we can rewrite the equation above with the multiplication of $P(w|z)$, the probability distribution of words given a topic, and $P(z|d)$, the probability distribution of topics given a document. This equation describes adding an additional topic selection step between the document selection step and the word selection step. As there are multiple latent topics where a word may come from, we sum the multiplication over a set of all the independent topics Z.

PLSI and other probabilistic topic models support multiple memberships using the probabilities $P(w|z)$ and $P(z|d)$. For example, if $P(w_{vehicle}|z_{car}) > P(w_{automobile}|z_{car})$, the word *vehicle* is more closely related to the topic *car* than the word *automobile*, though they are all related to the topic *car*. In this way, we can measure the strength of association between a word w and a topic z by the probability $P(w|z)$. Similarly $P(z|d)$ measures the strength of association between a topic z and a document d.

Equation (2) represents the log-likelihood function of PLSI:

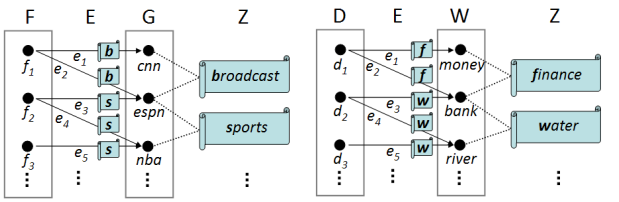
$$\begin{aligned} L &= \log[\prod_{d \in D} \prod_{w \in W} P(d, w)^{n(d,w)}] \\ &= \sum_{d \in D} \sum_{w \in W} n(d, w) \log P(d, w), \end{aligned} \quad (2)$$

where D and W denote a set of all d and w respectively, and $n(d, w)$ denotes the term frequency in a document (i.e., the number of times w occurred in d).

By maximizing the log-likelihood function L, we can maximize the probability to observe the entire corpus and accordingly estimate the $P(w|z)$ and $P(z|d)$ that most likely satisfy Equation (1).

3.2 Latent Dirichlet Allocation

Though PLSI is equipped with a sound probabilistic generative model and a statistical inference method, it suffers from the overfitting problem and does not cope well with unobserved words. To solve this problem, Blei et al. [4] introduced Dirichlet priors α and β to PLSI, to constrain $P(z|d)$ and $P(w|z)$, respectively. α is a vector of dimension $|Z|$, the number of topics, and each element in α is a prior for a corresponding element in $P(z|d)$. Thus, a higher α_i implies that the topic z_i appears more frequently than other topics in a corpus. Similarly, β is a vector of dimension $|W|$, the number of words, and each element in β is a prior for a corresponding element in $P(w|z)$. Thus, a higher β_j implies that the word w_j appears more frequently than other words in the corpus. As a conjugate prior for the multinomial distribution, the Dirichlet distribution can also simplify



(a) Subscription graph representation of our model (b) Subscription graph representation of the standard LDA

Figure 1: Subscription graph representation of models

the statistical inference. By placing Dirichlet priors α and β on the multinomial distributions $P(z|d)$ and $P(w|z)$, those multinomial distributions are smoothed by the amount of α and β and become safe from the overfitting problem of PLSI. It is also known that PLSI emerges as a specific instance of LDA under Dirichlet priors [7, 10].

3.3 Applying LDA to the Relationship Graph in a Social Network

Before justifying our approach, we briefly explain Twitter and a few interesting aspects of its data, which we use in our performance evaluations later in this paper. In contrast to a mutual friendship in other social networks, Twitter's relationships are unidirectional (i.e., a Twitter user does not need an approval from a user with whom she wants to make friends). Thus, we use the term *follow* when a user adds another user as her friend. Formally, when a user f follows a user g, f generates a *follow edge*, or simply an *edge*, $e(f, g)$ from a *follower* f to a *followed user* g. We also use $e'(f, g)$ to denote an edge from g to f (indicating that g is followed by f), $e(f)$ to denote the set of all outgoing edges from f, and $e'(g)$ to denote the set of all incoming edges to g. To refer to the set of all followers, the set of all followed users, and the set of all edges in the dataset we use F, G, and E, respectively.

Figure 1(a) depicts this notation using a graph that we refer to as a *subscription graph*. For example, we observe $e(f_1, cnn) = e'(cnn, f_1) = e_1$, $e(f_1) = \{e_1, e_2\}$, and $e'(espn) = \{e_2, e_3\}$ in Figure 1(a). Given this subscription graph, our goal is to label each edge with a correct label (interest) and group (label) each followed user based on those labeled edges. For example, since e_2 and e_3 are labeled with *broadcast* and *sports*, respectively, *espn* is labeled with both. Now we can frame our problem as the *graph labeling* problem of automatically associating each user g_i in G with a set of accurate interests z_k in Z based on its labeled incoming edges $e'(g_i)$. (We also label f_j in F as well.)

We can view the interest here as a topic in a document generative model. As briefly mentioned in Section 3.1, a document topic model assumes that an author has a topic in mind when selecting a word for a document. Likewise, when a user follows another user in Twitter (i.e., when a user generates a follow edge), the follower has an interest in the followed user. This interest may be caused by reasons such as sharing a common interest, having an off-line relationship, being popular, etc. Among these reasons for following someone on Twitter, the two most common reasons are sharing a common interest and being popular, since the

unidirectional nature of Twitter relationships allows a user to follow another user without requiring that user to follower her in return, as in the case of a blog subscription graph.

Furthermore, we can consider a follower f to be a document d, a followed user g as a word w, and a list of followed users for the follower as the content of the document. Figure 1 illustrates this equivalence between our *edge generative model* and the standard LDA and justifies our approach of applying LDA to a relationship graph in a social network without changing LDA's generative model.

We use the same notation with LDA. z denotes a labeling of a followed user with a topic (interest), or simply a topic, $P(z|f)$ denotes the multinomial distribution of topics given a follower, and $P(g|z)$ denotes the multinomial distribution of followed users given a topic. α and β are Dirichlet priors constraining $P(z|f)$ and $P(g|z)$, respectively.

3.4 Differences between LDA and Edge Generative Model

In the previous section, we described the equivalence between our edge generative model and the standard LDA. However, there is a subtle difference between the two generative processes. While words are sampled *with replacement* in the standard LDA, followed users are sampled *without replacement* in our model. For example, in a document generative model, a document may contain the word *car* multiple times. On the other hand, in our edge generative model, a user cannot follow the same user *Barack Obama* multiple times. As a result, the probabilistic distribution in our model does not follow a multinomial distribution but follows a *multivariate hypergeometric* distribution. Fortunately, the multinomial distribution can also be used for our model because it is known that a multivariate hypergeometric distribution converges to a multinomial distribution as the sample size grows large [1]. In our case, since sampling is done on millions of nodes, the two distributions become practically indistinguishable.

Also, when we represent E in matrix form by putting F in the rows and G in the columns as $E \in B^{F \times G}$, where $B = \{0, 1\}$, some differing aspects are noticed:

1. The rows and the columns are from the same entity U, a set of all Twitter users ($F, G \subseteq U$). In a matrix formed from a document corpus, documents are located in the rows and words are located in the columns.

2. The matrix is very big and sparse. Because users follow each other, the size of the rows and columns is almost equal to that of all Twitter users ($|F| \simeq |G| \simeq |U|$). The size of the matrix becomes $|F| \times |G|$ and most of its values are 0. This aspect is different from a matrix formed from a document corpus, where the size of the columns (words) is orders of magnitude smaller than that of the rows (documents).

3. The matrix is a binary matrix, in which 1 indicates the existence of an edge and 0 indicates no edge. Each element in a matrix formed from a document corpus is the number of occurrences of each word in a document.

4. As in a matrix formed from a document corpus, the distribution of the column sums show a power-law distribution, in which some small portion of the columns accounts for a majority of the total column sum. In a document corpus, these columns correspond to words such as *the* and *is*, which we call *stop words*, and are generally removed. Those columns in our matrix, however, correspond to users such as *Barack Obama* and *Britney Spears*, which we can call *popular users*, and should be taken into account.

Among the four major differences described above, the first and the third do not affect our effort to apply LDA to the follow edge dataset. Only appropriate pre-processing is required. The second limits the size of the analysis but can be solved by adding more resources or by dividing the work into multiple machines [20].

However, the last difference has a significant effect on our analysis because it is related to the quality of our analysis results. In a document corpus analysis, the stop words are generally removed before analysis, since an analysis without removing these stop words produces a very *noisy* result where the frequent stop words are labeled with every topic [4]. However, in our analysis, popular users are very important to include in the analysis because most users are keenly interested in following famous and influential users whose topic interests are similar to theirs. Unfortunately, it is not sufficient to simply include the popular users for LDA analysis, because the inclusion of popular users produces the same noisy result seen in the text analysis case: when stop words (or popular users) are included, they get included in every topic group, producing a very noisy result.

In the following section, we explore some LDA extensions to deal with the noise generated by popular users. Note that the earlier work on the application of LDA to an authorship graph dataset [9, 26] did not address how we can handle popular users. For example, [9] focused on the issue of how to transform co-authorship relations to a graph and [26] focused on how to deal with a large number of topic groups that are produced from the nodes whose connectivity is very *low*.

Before moving to the next section, we summarize the symbols used in this paper in Table 1.

Table 1: Symbols used throughout this paper and their meanings

Symbol	Meaning
u	A Twitter user
U	A set of all u
f	A follower
F	A set of all f
g	A followed user
G	A set of all g
z	A topic (interest)
Z	A set of all z
$e(f, g)$	A follow edge from f to g
$e'(f, g)$	A follow edge from g to f
$e(f)$	A set of all outgoing edges from f
$e'(g)$	A set of all incoming edges to g
E	A set of all $e(f, g)$ $\forall f \in F,\ \forall g \in G$

4. HANDLING POPULAR USERS

As discussed in Section 3.4, popular users generate noise if they are not dealt with carefully. This section describes how to handle this issue efficiently (i.e., how to label popular

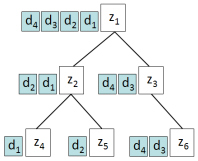


Figure 2: Topic hierarchy and documents generated in the hierarchy

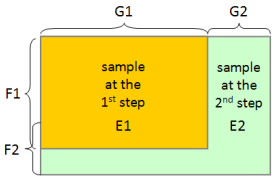


Figure 3: two-step labeling

users with correct labels). We first attempt to use the standard LDA with different settings (asymmetric priors). Then we explore the most appropriate LDA approach to this issue (Hierarchical LDA [3]) among a variety of LDA extensions we considered. Finally we propose two new LDA extensions of our own (*two-step labeling* and *threshold noise filtering*). The two new extensions can be combined together for better labeling quality.

4.1 Setting Asymmetric Priors

As mentioned in Section 3.2, LDA constrains the distributions of topics and words with Dirichlet priors α and β, respectively. Though each element of vectors α and β may take different values in principle, in the standard LDA, each element of α and β is assumed to have the same value (often referred to as the *symmetric prior assumption*). Intuitively, this assumption implies that every topic and word in a document corpus is equally likely.

Though the former sounds agreeable, the latter sounds unrealistic since it is very well known that the probability distribution of words follows a power-law distribution by Zipf's law. It is also the reason why stop words are removed before applying LDA to a document corpus, since stop words correspond to the head of the power-law distribution.

The most intuitive approach to address this issue would be to set a different prior for each followed user. Between the two priors α and β, we are only interested in β, the prior over the distribution of words given a topic, because a followed user corresponds to a word in the standard LDA. As a higher prior value implies a higher likelihood of being observed in the corpus, we set each prior value proportional to the number of incoming edges of each followed user. It is expected to associate popular users with more accurate labels as they are given adequate prior values.

We set β_{g_i}, the prior for the followed user g_i in the vector β, as in Equation (3):

$$\beta_{g_i} = \frac{0.98|e'(g_i)| + 0.01max(|e'(g)|) - 0.99min(|e'(g)|)}{max(|e'(g)|) - min(|e'(g)|)} \quad (3)$$

Note that we set the lowest value for each element in β_{g_i} as 0.01 and the highest value as 0.99 to make prior values skewed between 0 and 1.

4.2 Hierarchical LDA

Hierarchical LDA (HLDA) is also a good candidate for our problem because it generates a topic hierarchy and more frequent topics are located at higher levels. Figure 2 shows an example of topic hierarchy and document paths in the topic tree, where z_k denotes a topic and d_i denotes a document. In HLDA, when a document is generated according to Equation (1), words are chosen from topics in a document path. Since the top level topic is associated with all the documents, common words in every document (i.e., stop words) are expected to be labeled with the top level topic. On the contrary, the bottom level topics are expected to be more specific as they are associated with a small number of documents. For example, if z_2 is a topic about *network*, z_4 and z_5 would be a topic about *queueing* and *routing*, respectively. As z_1 is usually a topic consisting of the stop words, a document d_1 from the document tree path of z_1-z_2-z_4 consists of words from topic z_1, z_2, and z_4 and becomes a document about *network queueing*. Similarly in our model, z_1 is involved in every user's follow edge generation process and is expected to be associated with popular users.

This topic hierarchy is established because HLDA is based on the Nested Chinese Restaurant Process (NCRP), a tree extension to Chinese Restaurant Process, which probabilistically generates a partition of a set $\{1, 2, \ldots, n\}$ at time n. In NCRP, a document is considered as a Chinese restaurant traveler who visits L restaurants along a restaurant tree path, where L refers to the level of the tree (i.e., the length of the path).

4.3 Two-Step Labeling

In the previous sections, we explored existing LDA approaches to handle the popular user issue. Now we propose a new LDA extension: *two-step labeling*. We decompose the labeling process into two sub processes of establishing topics and labeling users with the established topics. In the first *topic establishment step*, we run LDA after removing popular users from the dataset similar to how we remove stop words before applying LDA to a document corpus. This step generates clean topics free from the noise generated by popular users. In the second *labeling step*, we apply LDA only to popular users in the dataset. As we use the collapsed Gibbs sampling algorithm [21], edges to popular users are labeled according to the pre-established topics as represented in Equation (4):

$$P(e(f_i, g_j) = z|\cdot) \propto \frac{N_{g_j z} + \beta}{\sum_{k=1}^{|G|}(N_{g_k z} + \beta)} \frac{N_{f_i z} + \alpha}{\sum_{k=1}^{|Z|}(N_{f_i z_k} + \alpha)}, \quad (4)$$

where $P(e(f, g) = z|\cdot)$ denotes the probability of labeling the edge from a follower f to the followed user g with a topic z given all conditions, N_{gz} denotes the number of times g is labeled with z, and N_{fz} denotes the number of times f is labeled with z.

This equation implies that an edge to a followed user is assigned to a topic according to how many times that user has been assigned to the topic, as well as how many times that topic has been previously assigned to the follower following that user. Thus, if some assignments are made in the first step, they affect assignments in the second step. For example, if a user f_1 follows non-popular users g_1 and g_2 and a popular user g_3, g_1 and g_2 are sampled at the first step and g_3 is sampled at the second step with a higher likelihood to

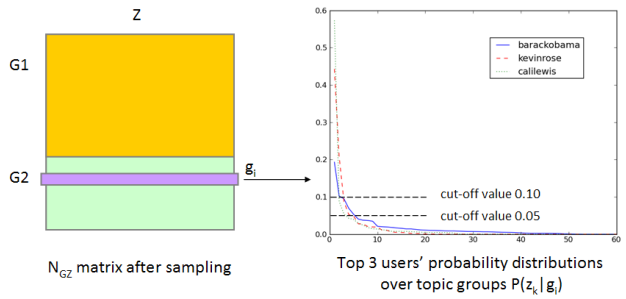


Figure 4: An example of threshold noise filtering process

be labeled with the topic that g_1 or g_2 are labeled with at the first step.

This approach is illustrated in Figure 3, where the $E1$ part of the dataset is sampled at the first step and the $E2$ part is sampled at the second step. Note that *two-step labeling* does not increase computational complexity because it samples a different part of the dataset at each sampling step. $(O(|Z| \cdot |E|) = O(|Z| \cdot (|E1| + |E2|))$

There is related research in the literature. Online LDAs introduced in [5] are designed to deal with a growing corpus and sample topics for words in a document as it comes in. Though both *two-step labeling* and online LDAs use multi-step sampling, their goals are totally different. While online LDAs try to change topics as the corpus grows over time, *two-step labeling* fixes a set of topics and labels users with this fixed set of topics. From Figure 3, $G1$ and $G2$ (words) are mutually exclusive in *two-step labeling* while $F1$ and $F2$ (documents) are mutually exclusive in online LDAs.

4.4 Threshold Noise Filtering

As briefly described in Section 3.1, the association between a user (a word) and a topic has varying association strengths represented by $P(g|z)$ ($P(w|z)$) in a probabilistic topic model. Thus, we can list users labeled with the same topic in descending order of $P(g|z)$ and regard the top entries in the list (*topic group*) as more important than the entries at the bottom because they are more strongly associated with the topic. Similarly, we can measure the association strength from the user's viewpoint using $P(z|g)$. Though *two-step labeling* may help label popular users with the right topics, popular users may take top positions even in less-relevant topic groups because even the smallest number of times a popular user is assigned to a topic group could outnumber the largest number of times a non-popular user is assigned to that topic group.

To mitigate this problem, we propose a new approach, *threshold noise filtering*, which sets a cut-off value to determine whether to label a user with each topic. By ignoring assignments below the cut-off value, we can expect smoothing and noise reduction effects as in the *anti-aliasing filter*. We set $N_{g_i z_k}$, the number of times a followed user g_i is assigned to a topic group z_k, as 0 if it is below the cut-off value C:

$$N_{g_i z_k} = \begin{cases} 0 & \text{if } P(z_k|g_i) = \frac{N_{g_i z_k}}{\sum_{j=1}^{|Z|} N_{g_i z_j}} < C \\ N_{g_i z_k} & \text{otherwise.} \end{cases}$$

As *threshold noise filtering* process is done after sampling, it does not increase computational complexity similar to *two-step labeling*. Figure 4 illustrates this process and shows the top three popular users' distributions over topic groups. (Other normal users also show similar non-linear distributions.) Alternatively, we may filter out less relevant topics by keeping only the top-K topics for each user, for a reasonably small K value. We tested both schemes (threshold noise filtering and top-K filtering), but we couldn't see any practical differences. Due to lack of space, we only report the results with the former scheme. Though *threshold noise filtering* can be used with any other approaches, we combine it with the two most representative cases, *two-step labeling* and the standard LDA in our experiments.

5. EXPERIMENTS

In this section, we experimentally compare our proposed extensions to LDA, *two-step labeling* and *threshold noise filtering*, against existing approaches. We ran experiments on a real-world dataset obtained from Twitter and compared the performance of various approaches under two evaluation metrics, *perplexity* and *quality*. As we will see from our results, our proposed extensions show performance equivalent to or significantly better than existing approaches in our experiments.

In Section 5.1, we describe our dataset and the overall experimental setup. In Sections 5.2 and 5.3, we report our results on *perplexity* and *quality*, respectively. We also provide a qualitative comparison of different approaches in Section 5.4.

5.1 Dataset and Experiment Settings

For our experiments, we use a Twitter dataset we collected between October 2009 and January 2010. The original downloaded dataset contained 273 million follow edges, but we sampled 10 million edges from this dataset to keep our experiment manageable. To ensure that all the follow edges were preserved in our sampled dataset, we first sampled *followers* randomly and included all follow edges from the sampled users, until we obtained 10 million follow edges.

Figure 5 shows the distributions of incoming and outgoing edges in the sampled dataset. The horizontal axis shows the number of edges at a node and the vertical axis shows how many nodes have the given edge count. Both axes are shown in a logarithmic scale. From the graph, it is clear that the number of incoming edges follows a power-law distribution, which is often the case for this type of dataset. Interestingly, we observe that the number of outgoing edges is quite uniform between edge counts 1 to 100, which is different from the distribution reported in [14]. We do not believe this difference is due to our sampling, because the graph from the complete dataset shows the same flat curve between the edge counts 1 and 100 [25]. It could be an interesting future work to investigate where this difference comes from. In our sampled dataset, *barackobama* has the most followers, 7410, and *zappos* follows the most users, 142,669. Table 2 shows some basic statistics of the sampled dataset.

Since we are mainly interested in investigating how different approaches handle popular users, we categorized the followed users G into two distinct sub groups according to their incoming-edge counts. One group is the *normal* user group, where $|e'(g)| \leq V$, a boundary value, and the other group is the *popular* user group, where $|e'(g)| > V$. We tested three

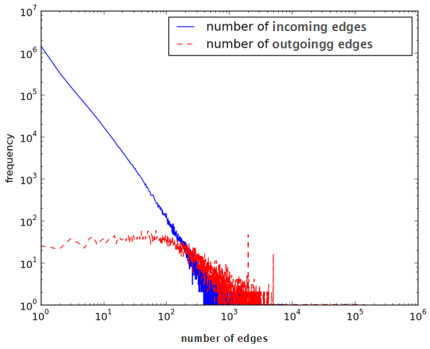


Figure 5: Distributions of incoming and outgoing edges

Table 2: Statistics of our Twitter dataset

Statistics	Value
$\|E\|$	$10,000,000$
$\|U\|$	$2,430,237$
$\|F\|$	$14,015$
$\|G\|$	$2,427,373$
$max(\|e(f)\|)$	$142,669$ (f: *zappos*)
$max(\|e'(g)\|)$	$7,410$ (g: *barackobama*)

boundary values $V = 50$, 100, and 500. Since all the results from the three boundary values show consistent patterns, we only report the result from the case when $V = 100$, where the popular user group consists of only 0.3% of all followed users but accounts for 20.2% of all follow edges.

Table 3 summarizes the eight representative experimental cases we report on in this section. *base* is the standard LDA experiment over the whole group. *non-popular* is also the standard LDA experiment but this experiment is applied only to the normal user group to remove the noise generated by popular users. For *beta*, we use asymmetric Dirichlet priors for the hyperparameter β, where we have tested proportional, inversely-proportional, and ladder-shape prior schemes. *hlda-2lv* and *hlda-3lv* are HLDA experiments with two and three levels each. *2step* is our *two-step labeling* experiment. For *threshold noise filtering*, where we have tested multiple threshold settings including $C = 0.01$, 0.05, 0.10, and $K = 1$, 2, 3, we generate *filter-base* and *filter-2step* by combining *threshold noise filtering* with *base* and *two-step labeling* each. Due to space limits and to clarify our discussion, we only report the results for a subset of our experimental settings in this paper. In particular, we report the result from proportional β priors for *beta* and $C = 0.05$ for *threshold noise filtering*. The results that are not reported in this paper are very similar to what we report here.

In all of our experiments, we generated 100 topic groups (50 groups for *hlda-2lv*) and picked the top ten entries in each topic group according to their probabilities $P(g|z)$ in each group.

5.2 Perplexity

In this section, we report our results on the *perplexity* metric, which was also used in earlier related work such as [4, 9,10,26]. *Perplexity* is a well-known standard metric used in IR. It tries to quantify the accuracy of a model by measuring

Table 3: Experimental cases and descriptions

Case	Experiment Description
base	LDA over the whole group dataset
non-popular	LDA over the normal user group dataset
beta	LDA with asymmetric β priors
hlda-2lv	HLDA with 2 levels
hlda-3lv	HLDA with 3 levels
2step	Two-step labeling
filter-base	Threshold noise filtering after *base*
filter-2step	Threshold noise filtering after *2step*

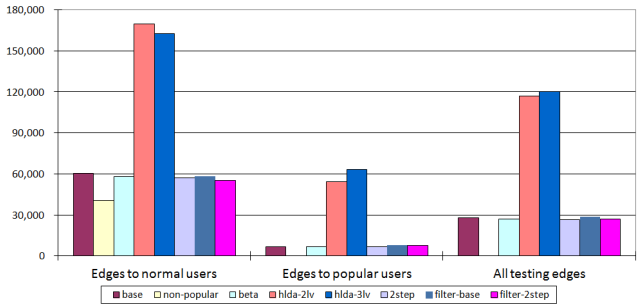


Figure 6: *Perplexity* comparison

how well the trained model deals with an unobserved test data. More precisely, perplexity is defined to be [26]:

$$Perplexity(E_{test}) = \exp^{-\frac{\sum_{e \in E_{test}} \log P(e)}{|E_{test}|}},$$

where E_{test} denotes all the edges in the test dataset.[1] We calculated *Perplexity* values for the 10% held-out dataset after training the models on the remaining 90% dataset. In general, lower perplexity means better generalizability.

Note that the original LDA model is designed to derive an optimal model that minimizes perplexity. Therefore, it is unlikely that extensions to base LDA show lower perplexity. Our primariy goal of comparing the results under the perplexity metric is to see whether our proposed extensions significantly degrade perplexity. Figure 6 compares *Perplexity* values of the eight experimental cases. The last set of bars show the *Perplexity* values on the overall test dataset. The first set of bars show *Perplexity* only on the edges to normal users and the second set of bars show *Perplexity* only on the edges to popular users.

From the graphs, we first notice that HLDA (*hlda-3lv* and *hlda-2lv*) shows significantly worse *Perplexity* than others. This is due to the fact that the standard LDA is designed to minimize *Perplexity* while HLDA is not necessarily designed for this task. We also note that the *Perplexity* value for *non-popular* is significantly lower than others. This is because we eliminate all edges to popular users in *non-popular* and do *not* include these edges in computing the *Perplexity* value. Therefore, *non-popular* had an unfair advantage of "ignoring" all noise from popular users and not being evaluated on them. Overall, we find that our *two-step labeling* and *threshold noise filtering* show similar perplexity values

[1]The *Perplexity* values of *hlda-2lv* and *hlda-3lv* are calculated with the empirical likelihood values provided by Mallet (http://mallet.cs.umass.edu).

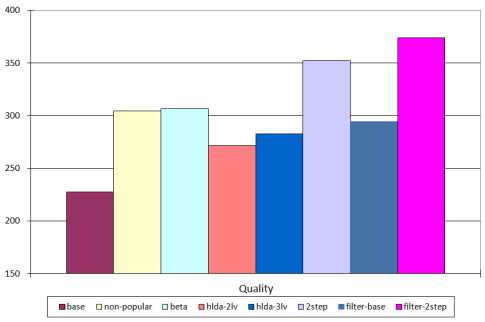


Figure 7: *Quality* comparison

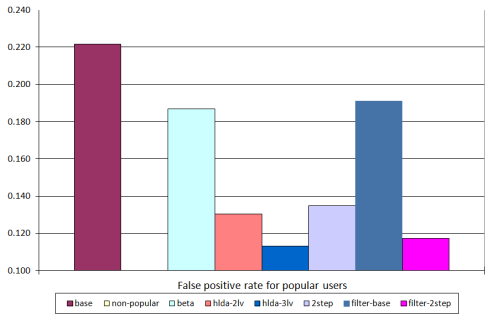


Figure 8: False positive rate of popular users

to the standard LDA model.[2] That is, our extensions to LDA do not introduce noticeable *Perplexity* degradation to the standard LDA model.

5.3 Quality

Ultimately, the effectiveness of various approaches should be determined by how users perceive the quality of the identified topic groups from each approach. To measure human-perceived quality, we conducted a survey with a total of 14 participants. The participants in our survey were presented with a random group of ten Twitter users (identified by one of the eight approaches described in Section 5.1) and were asked to indicate if each user in the topic group was relevant to the group or not. Overall, we collected 23 judged topic groups per each approach with a total of 161 judged topic groups.

Given the survey results, we computed the human-perceived *Quality* value of the topic groups Z identified by an approach as:

$$Quality(Z) = \sum_{k=1}^{|Z|} \sum_{i=1}^{|z_k|} \delta(g_i, z_k) \log(|e'(g_i)|),$$

where δ function is defined as:

$$\delta(g_i, z_k) = \begin{cases} 1 & \text{if user } g_i \text{ is related to topic group } z_k \\ -1 & \text{if user } g_i \text{ is not related to topic group } z_k. \end{cases}$$

Note that in the above *Quality* formula, the factor $\log(|e'(g_i)|)$ is added to assign higher weights to more popular users, because most people are interested in following popular users and pay more attention to the correct topic clustering of those users.

Figure 7 reports the results from this survey, where each bar shows the *Quality* value of one of the eight approaches. From this graph, we observe the following:

1. Both of our extensions, *two-step labeling* and *threshold noise filtering* are very effective in improving the human-perceived quality of identified topic groups. For example, when compared to *base*, *filter-2step* achieves a 1.64 times higher *Quality* value.

2. As *two-step filtering* initially forms clean topics free from popular user generated noise, its gain is more significant than that of *threshold noise filtering*. However, *threshold noise filtering* can be easily combined with any other approaches to improve *Quality*, as in *filter-base*.

3. If we apply the standard LDA to popular users as well without any pre-filtering, the produced topic groups are quite noisy. For example, the *Quality* value of *base* is 1.34 times lower than *non-popular*, which filters out all popular users first.

4. Using an asymmetric Dirichlet prior for the hyperparameter β improves *Quality*, but the result is still noisier compared to our extensions.

5. The HLDA model shows only marginal improvements in the *Quality* metric when compared to *base*. This low performance is because most of the low-level groups in HLDA have users with very few followers (who are less likely to be interesting when recommended) and contribute only small weights to *Quality*.

In Figure 8, we also report the false positive rates for popular users, which measures incorrect labeling of popular users. That is, whenever a user in the popular user group is placed in a topic group from our analysis, our survey participants evaluated whether the topic group is a good fit for that user. Figure 8 reports what fraction of them were considered to be a "bad fit" or an "incorrect" association. Different from Figure 7, the HLDA models seemingly perform as well as or even better than our two approaches. This is because HLDA tends to place popular users at top-level groups. When our survey participants looked at such a group, they simply considered it as a "popular user group" and determined that the popular users in the group were a good fit, even though the group itself did not exhibit any coherent topic. Therefore, even though the false positive rate for HLDA is low, it does not necessarily improve the topic-based recommendation accuracy for popular users. We can observe that our *2step* and *filter-2step* show comparably low false positive rates similar to those of the HLDA approaches.

5.4 Qualitative Comparison

In this section, we examine the topic groups generated from each approach more closely to get a better sense of the quality of produced topic groups.

Figure 9 shows the top ten users in a topic group from *base*, which we name as the *cycle* group. Together with their Twitter usernames, we show their follower counts $|e'(g)|$, and the bio of each user. By going over the users' bios, we can

[2]In fact, the *Perplexity* values of our two approaches are lower than the standard LDA in our experiments, but we expect that this is simple experimental fluctuation that does not have a significant meaning.

g	|e′(g)|	Bio
lancearmstrong	2370	7-time Tour de France winner, full time cancer fighter - LIVESTRONG!
levileipheimer	285	Pro Cyclist
ghincapie	234	*[No Bio] Professional Cyclist*
johanbruyneel	203	9X winning Tour De France Sports Director
barackobama	7410	44th President of the United States
teamradioshack	141	*[No Bio] Pro Cycle Team RadioShack*
dzabriskie	137	Professional Bike Racer
phillliggett	126	Pro-cycling commentator.
christianvdv	114	Professional Cyclist
stephenfry	3384	British Actor, Writer, Lord of Dance, Prince of Swimwear & Blogger

Figure 9: Topic group *cycle* from *base* showing the popular user issue

g	|e′(g)|	Bio
googleimages	94	News, tips, and tricks direct from the Google Images team
androidandme	84	Meet your new Android friend.
androidguys	73	Your trusted source for Android news and opinion since November 5, 2007.
prethinking	66	PreThinking.com - your source for all things Palm Pre
engadgetmobile	97	Official Twitter account of Engadget Mobile!
skyfire	88	Skyfire 3.0 for Android and Skyfire 2.0 for iPhone are changing the way people browse on thier mobile devices! www.get.skyfire.com
precentral	75	All about Palm webOS devices
boygenius	95	President, General Manager, Editor-in-chief of BGR.com. Professional smack talker.
theandroidsite	71	News and reviews of Google's Android OS
nokconv	58	A team of Nokia tweeters. Follow us for various updates, info, and the occasional banal banter.

Figure 10: Topic group *mobile gadget blog* from *non-popular*

g	|e′(g)|	Bio
google	3143	News and updates from Google
twitter	3715	Always wondering what's happening.
techcrunch	3565	Breaking Technology News And Opinions From TechCrunch
tweetdeck	2649	TweetDeck is your personal browser for staying in touch with what's happening now, connecting you with your contacts across Twitter, Facebook and more.
mashable	3938	Breaking social media, tech and digital news and analysis from Mashable.com, the top resource and guide for all things web. Updates from @mashable staff.
cnnbrk	3787	CNN.com is among the world's leaders in online news and information delivery.
wefollow	2861	A user powered twitter directory located at http://wefollow.com
googlereader	691	News, tips and tricks from the Google Reader team
hootsuite	1708	Updates about the social media dashboard which allows teams to broadcast, monitor and track results. For support, follow @hootsuite_help + for news @hootwatch.
breakingnews	2210	The latest breaking news alerts around the world from hundreds of sources.

Figure 11: Topic group from *2step* corresponding to Figure 10

g	|e′(g)|	Bio
google	3143	News and updates from Google
techcrunch	3565	Breaking Technology News And Opinions From TechCrunch
tweetdeck	2649	TweetDeck is your personal browser for staying in touch with what's happening now, connecting you with your contacts across Twitter, Facebook and more.
mashable	3938	Breaking social media, tech and digital news and analysis from Mashable.com, the top resource and guide for all things web. Updates from @mashable staff.
wired	1293	Official Twitter feed for Wired magazine & Wired.com. Steering the ship this week: sports editor @erikmal
googlereader	691	News, tips and tricks from the Google Reader team
firefox	750	I make the web a little faster, safer, smarter, better.
youtube	673	Tweets on YouTube news, trends, and — of course — videos.
engadget	827	Official Twitter account of Engadget!
seesmic	1075	I make updates and announcements for Seesmic.com Send feedback/questions to @askseesmic

Figure 12: Topic group from *filter-2step* corresponding to Figure 10

see that many of the users in the group have the same interest *cycle*. However, we also observe that among the three users with the highest number of incoming edges, *barackobama*, *stephenfry*, and *lancearmstrong*, only *lancearmstrong* is related to *cycle*. The other two users, *barackobama* and *stephenfry* are included in this group simply because they are famous and followed by many users. In particular, we note that *barackobama* appears in 15 groups out of 100 topic groups produced by *base*. Among the 15 groups, only one of them is related to his specialty, *politics*, which clearly shows that the standard LDA suffers from the noise from popular users if applied directly to our social graph dataset.

Figure 10 shows an example topic group produced by *non-popular*. Note that in this group, all users have $|e'(g)|$ values of smaller than 100, because popular users are removed from the dataset. Therefore, none of the popular users will belong to a topic group under this approach. When we go over the users' bios, we see that *all* users in this group is somewhat related to *mobile gadgets*. That is, the topic group produced by *non-popular* is significantly cleaner (less noisy) than that from *base* at the expense of not being able to group any popular users.

Figure 11 shows an example topic group from *2step*. Note that many users in this group are very popular and they are mainly about the same topic, *tech media*, indicating that *2step* is able to group popular users into the right topic group in general. However, we observe that *2step* still suffers from the presence of a few popular, yet less relevant users in the group (in this example, *cnnbrk* and *breakingnews* may be considered less relevant to the group than others).

With *threshold noise filtering*, we achieved less noisy results. Figure 12 shows a result topic group from *filter-2step* corresponding to the group in Figure 11 from *2step*. We observe that *cnnbrk* and *breakingnews*, popular users on general media, are now removed from Figure 11 and more technology-centric media such as *firefox*, *youtube*, and *engadget* are added. Note that *firefox* and *youtube* are Twitter accounts publishing news related to FireFox and YouTube. As we pick only a few most probable topics for popular users in *filter-2step*, they have less chance to appear in less-relevant topic groups.

6. CONCLUSION

In this paper, we applied LDA to analyze the relationship graph in a large social network. Different from the usual approaches that extract topics from textual data, such as bio and tweets, our approaches rely purely on the social-

network graph consisting of follow edges. Even with this relatively limited type of data compared to those of previous approaches, our approaches generated very well-organized results and showed the great potential of applying LDA to these kinds of clustering applications. Our approaches are especially useful when only linkage data is available.

We also dealt with popular user generated noise, which is inevitable in a large social network. Unlike the stop words in the standard LDA applications, popular users have important meanings and should be dealt with carefully. We explored four extensions to the standard LDA and quantitatively and qualitatively analyzed their results using a Twitter dataset. Our proposed approaches, two-step labeling and threshold noise filtering, are very effective in handling this popular user issue, showing 1.64 times improvement in the quality of the produced topic groups, compared to the standard LDA model.

7. ACKNOWLEDGEMENTS

We would like to thank Jong Han Park, and Christopher Moghbel for their help and feedback throughout this research.

8. REFERENCES

[1] Multinomial distribution. `http://en.wikipedia.org/wiki/Multinomial_distribution`.

[2] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. In *J. Mach. Learn. Res.*, 2008.

[3] D. M. Blei, T. L. Griffiths, M. I. Jordan, and J. B. Tanenbaum. Hierarchical topic models and the nested chinese restaurant process. In *Neural Information Processing Systems*, 2003.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[5] K. R. Canini, L. Shi, and T. L. Griffiths. Online inference of topics with latent dirichlet allocation. In *Artifical Intelligence and Statistics*, 2009.

[6] M. Ester, H. peter Kriegel, J. S, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining*, pages 226–231. AAAI Press, 1996.

[7] M. Girolami and A. Kaban. On an equivalence between plsi and lda. In *SIGIR*, 2003.

[8] M. Harvey, I. Ruthven, and M. J. Carman. Improving social bookmark search using personalised latent variable language models. In *WSDM*, 2011.

[9] K. Henderson and T. Eliassi-Rad. Applying latent dirichlet allocation to group discovery in large graphs. In *Proceedings of the 2009 ACM symposium on Applied Computing*, 2009.

[10] M. D. Hoffman, D. M. Blei, and F. Bach. Online learning for latent dirichlet allocation. In *In NIPS*, 2010.

[11] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR*, 1999.

[12] T. Iwata, T. Yamada, and N. Ueda. Modeling social annotation data with content relevance using a topic model. In *NIPS*, 2009.

[13] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 1999.

[14] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 591–600, New York, NY, USA, 2010. ACM.

[15] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, pages 129–137, 1982.

[16] A. Mccallum, X. Wang, and A. Corrada-Emmanuel. Topic and role discovery in social networks with experiments on enron and academic email. *Journal of Artificial Intelligence Research*, 30:249–272, 2007.

[17] Q. Mei, D. Cai, D. Zhang, and C. Zhai. Topic modeling with network regularization. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pages 101–110, New York, NY, USA, 2008. ACM.

[18] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel. You are who you know: Inferring user profiles in online social networks. In *WSDM*, 2010.

[19] N. Pathak, C. DeLong, A. Banerjee, and K. Erickson. Social topic models for community extraction. In *The 2nd SNA-KDD Workshop*, 2008.

[20] A. Smola and S. Narayanamurthy. An architecture for parallel topic models. In *VLDB*, 2010.

[21] M. Steyvers and T. L. Griffiths. Probabilistic topic models. *Handbook of Latent Semantic Analysis*, 2007.

[22] M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths. Probabilistic author-topic models for information discovery. In *SIGKDD*, 2004.

[23] V. Tuulos and H. Tirri. Combining topic models and social networks for chat data mining. In *In Proc. of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*, 2004.

[24] X. Wang, N. Mohanty, and A. Mccallum. Group and topic discovery from relations and text. In *In Proc. 3rd international workshop on Link discovery*, pages 28–35. ACM, 2005.

[25] M. J. Welch, U. Schonfeld, D. He, and J. Cho. Topical semantics of twitter links. In *WSDM*, 2011.

[26] H. Zhang, B. Qiu, C. L. Giles, H. C. Foley, and J. Yen. An lda-based community structure discovery approach for large-scale social networks. In *In IEEE International Conference on Intelligence and Security Informatics*, pages 200–207, 2007.

[27] D. Zhou, E. Manavoglu, J. Li, C. L. Giles, and H. Zha. Probabilistic models for discovering e-communities. In *World Wide Web Conference*, 2006.