Structural and Temporal Analysis of the Blogosphere Through Community Factorization

Yun Chi Shenghuo Zhu Xiaodan Song Junichi Tatemura Belle L. Tseng
NEC Laboratories America, 10080 N. Wolfe Rd, SW3-350, Cupertino, CA 95014, USA
{ychi,zsh,xiaodan,tatemura,belle}@sv.nec-labs.com

ABSTRACT

The blogosphere has unique structural and temporal properties since blogs are typically used as communication media among human individuals. In this paper, we propose a novel technique that captures the structure and temporal dynamics of blog communities. In our framework, a community is a set of blogs that communicate with each other triggered by some events (such as a news article). The community is represented by its structure and temporal dynamics: a *community graph* indicates how often one blog communicates with another, and a *community intensity* indicates the activity level of the community that varies over time. Our method, *community factorization*, extracts such communities from the blogosphere, where the communication among blogs is observed as a set of subgraphs (i.e., threads of discussion). This community extraction is formulated as a factorization problem in the framework of constrained optimization, in which the objective is to best explain the observed interactions in the blogosphere over time. We further provide a scalable algorithm for computing solutions to the constrained optimization problems. Extensive experimental studies on both synthetic and real blog data demonstrate that our technique is able to discover meaningful communities that are not detectable by traditional methods.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering*; H.2.8 [**Database Management**]: Database Applications—*Data mining*

General Terms

Algorithms, Experimentation, Measurement, Theory

Keywords

Blog, Blogosphere, Community Factorization, Non-negative Matrix Factorization, Regularization, Iterative Search

KDD'07, August 12–15, 2007, San Jose, California, USA.

1. INTRODUCTION

Blog is self-publishing media on the Web that has been growing quickly and becoming more and more important. At the time of this writing, Technorati, a well-known blog search engine, is tracking 57 million blogs, with 1.3 million new entries generated everyday. Furthermore, those numbers have doubled every 6 months for the past 3 years[1]. The *blogosphere*, the universe of blogs, provides a lot of applications in the areas of economics and finance (e.g., virus marketing and opinion extraction), social relations (e.g., online friendships), politics (e.g., as media outlets for political candidates), and so on. The blogosphere raises interesting research challenges since blogs have various distinct features compared with general Web pages.

A key difference between the blogosphere and the Web is the lifetime of their contents (i.e., pages and links). Note that a blog is typically used as a tool for communication and it consists of a temporal sequence of entries. Driven by an event (such as a news), blogs publish their entries that refer to each other. Most of such entries quickly become obsolete and will never be referred to by later entries. Thus, links among blog entries have significant temporal locality, resulting in a number of dense subgraphs (each of which is a thread of discussion) with a short lifetime. On the other hand, the content of the Web has longer lifetime, and it is common that a new page refers to a very old page (such as an authoritative page). When a new page appears with links to an old page in a subgraph, the subgraph grows. Thus, the structure of the subgraph gradually and incrementally changes over time.

Given this unique structure of the blogosphere, the analysis of temporal dynamics in blog communities should be different from the traditional web analysis. In the general web analysis, a dense subgraph of web pages is often considered as a community, and its temporal dynamics is captured by observing how the subgraph grows over time. However, in the blogosphere, a dense subgraph grows only within a short time span. The traditional analysis can only capture dynamics within a short-term activity (such as a single thread of discussion).

Alternatively, we may accumulate such links over a very long period, generating a graph of blog that represents how much they communicate with each other (i.e., a social network of blogs). Here, we can apply similar analysis to the Web pages since the graph is relatively static and changes incrementally. However, it misses more detailed temporal behavior. For instance, assume that there are two commu-

[1]http://www.sifry.com/alerts/archives/2006_11.html

nities, a politics community and an economics community, and that one community becomes inactive while the other community becomes active during the observation period due to change of trends in the real world. For example, the politics community becomes inactive after a particular political issue is addressed, then people start worrying about its impact to economics. Some blogs are interested both in politics and economics, moving from one community to another. Since the two communities have overlap, they will be seen as a single community in the aggregated graph.

In this paper, we propose a novel technique, *community factorization*, to extract communities and their temporal dynamics in the blogosphere. Our technique identifies the longer-term graph structure (e.g. the politics community and the economic community) from a series of short-term subgraphs (i.e., threads of discussion among blogs).

In our framework, a community is a set of blogs that communicates with each other in a synchronized manner, i.e., communication is triggered by some events (such as a news article), resulting in a number of short-term subgraphs. A community has its structure, called a *community graph*, which represents how much one blog communicates with another. By observing a number of short-term subgraphs, we estimate the structure of a community graph. A community also has its temporal aspect: It can become active and inactive over time. We introduce a *community intensity* to represent the activity level of a community at a particular time. Then the blogosphere at a particular time should be represented as mixture of community graphs weighted by their intensities. Our technique formulates this as a problem of non-negative matrix factorization.

Our main contributions in the paper can be summarized as follows.

(1) We provide a new model of community that has both structural and temporal aspects, i.e., community graphs and community intensities, in order to handle the unique structure of the blogosphere: a series of short-term subgraphs representing the communication among blogs.

(2) We pose and solve the problem of extracting communities as a factorization problem in the framework of constrained optimization, in which the objective is to best explain the observed interactions in the blogosphere over time.

Experimental results on both synthetic data and two real-life data show that our algorithm is able to detect significant communities with temporal dynamics while many of these communities are not detectable by traditional methods on aggregated blog graphs.

The rest of the paper is organized as follows. We discuss background information and related work in Section 2. In Section 3 we present our algorithm for community factorization. In Section 4 we provide a computational procedure to solving the community factorization problem. We show results of experimental studies in Section 5 and give conclusion and future directions in Section 6.

2. BACKGROUND AND RELATED WORK

2.1 Some Mathematical Notations

We introduce some mathematical notations that will be used in later sections. We denote scalars by lower-case letters (a,b,...), matrices by capital letters (A,B,...), and tensors by calligraphic letters ($\mathcal{A}$,$\mathcal{B}$,...). In addition, for a matrix A we use $A(j)$ to denote the j-th column of A, and we use both a_{ij} and $(A)_{ij}$ to represent the element at the i-th row and j-th column of A. We use $\mathcal{R}$ to represent the set of real numbers and $\mathcal{R}_+$ for non-negative real numbers. A *non-negative* matrix is a matrix whose elements are in $\mathcal{R}_+$. For a general matrix A we use A_+ to represent A's positive part and A_- to represent A's negative part. That is, $A_+ = (A+|A|)/2$ and $A_- = (|A|-A)/2$ where both A_+ and A_- are non-negative matrices. For a tensor $\mathcal{A} \in \mathcal{R}^{m\times n\times p}$, a *3-mode product*[2] of $\mathcal{A}$ by a matrix $U \in \mathcal{R}^{p\times q}$ results in a tensor $\mathcal{B} \in \mathcal{R}^{m\times n\times q}$ where $b_{ijk} = \sum_{l=1}^{p} a_{ijl}u_{lk}$. Informally, we can say the k-th "slice" of $\mathcal{B}$ is a linear combination of all "slices" of $\mathcal{A}$ where the coefficients are the k-th column of U. Finally, unless stated otherwise, all matrix (tensor) norms $||\cdot||$ in this paper means Frobenius norm, i.e., $||A|| = \sqrt{\sum_{i,j} a_{ij}^2}$.

2.2 Related Work

Community extraction and analysis has been studied extensively and is mainly studied as a graph problem. Flake et al. [7] defined communities as dense subgraphs and proposed algorithms for identifying communities by using a maximum flow/minimum cut framework. Ino et al. [12] proposed different algorithms based on ideas similar to that of Flake. These studies extract communities from a static (aggregated) graph and miss the details on the dynamic behavior about the communities.

There is also a large body of work on analyzing online social networks. Kumar et al. [13] studied the bursty evolution of links among different communities in the blogosphere by using a Markov chain model. Gruhl et al. [9] proposed a generative model—transmission graph—to model the information diffusion in the blogosphere in a way similar to modeling disease-propagation in epidemic studies. Leskovec et al. [16] studied the patterns of growth for graphs in various fields and proposed generators that produce graphs exhibiting the discovered patterns. Kumar et al. [14] analyzed the temporal dynamics of the structures of the social networks of Flickr and Yahoo! 360. Most of these studies, however, only address high-level macro statistics of the online networks such as their size, density, degree distribution as well as the evolution of these macro statistics. In comparison, our algorithm can analyze the micro structure and temporal trends down to the level of individual communities.

Recently, there have been many research works on mining temporal patterns from a collection of documents with time stamps. Mei et al. [17] proposed a model that takes time and location of blogs are into consideration and mines spatio-temporal theme patterns by using a probabilistic approach. Wang et al. [23] proposed the *Topic over Time* (TOT) model to capture temporal topical trends by extending the well-known Latent Dirichlet Allocation (LDA) model with a temporal component. However, in these studies, documents in the collection are treated as separated observations that are generated independently and therefore the interactions among communities are ignored.

There are some other recent studies that are closely re-

[2]To be consistent with the standard NMF, in our definition of 3-mode product, U is multiplied from *right*, which is different from that defined in [5].

lated to our work. Backstrom et al. [1] analyzed two large-scale time-resolved social networks and studied the dynamic community formation. However, in their study, the community memberships are explicitly available in the data sets. Chakrabarti et al. [2] proposed a novel evolutionary clustering algorithm in which the current clusters are affected by the historic cluster memberships. Our algorithm has similar effect to Chakrabarti's algorithm, i.e., consistency of community structures over time is emphasized. However, instead of discovering clusters at each time window as Chakrabarti's algorithm did, our algorithm extracts communities that exist over all the time history. Falkowski et al. [21] proposed to cluster communities obtained in each time window in order to visualize the evolution of communities and their clustering is based on a heuristic similarity between communities at different time windows. Qamra et al. [19] proposed a community-based temporal clustering using the Chinese Restaurant Process. Unfortunately, the inference of their model requires Gibbs sampling, which is notoriously slow for large scale data. In [3], we applied the high-order singular value decomposition (HOSVD) to extract dynamic structural changes of the blogosphere for a given query keyword. There are two weak points in that work: First, only some high-level signatures (e.g.,*hub* and *authority* scores) of the communities were extracted. Second, because HOSVD is an orthogonal decomposition ([5]), most community structures and temporal trends discovered in [3] contain negative values and they do not map directly to communities and temporal trends in real world.

3. COMMUNITY FACTORIZATION

In this section, we propose a new technique, called *community factorization*, to extract communities from the blogosphere.

3.1 Basic Idea

As discussed in Section 1, we want to extract a community as a structure that lasts for a longer time period whereas the blogosphere consists of a number of short-term subgraphs.

We define a community as a set of blogs that communicates with each other in a synchronized manner, i.e., communication among the community members is triggered by some events (such as a news article). Such communication is observed as a number of dense subgraphs, each of which is a short-term thread of discussion.

Finding a community is to identify its structure and temporal dynamics: a *community graph* that represents how much one blog communicates with another, and a *community intensity* that represents the activity level of the community varying over time.

Since the communication in a community is observed as dense subgraphs, the structure of such subgraphs should reflect the community graph structure. However, the structure of a single dense subgraph does not necessarily reflect the entire structure of a community: Since a member does not always participate in a thread of discussion, a community may appear as smaller pieces of disconnected subgraphs at a particular time. Moreover, it is possible that members of different communities participate in a single subgraph.

Our idea here is to represent a community structure as a combination of the observed subgraphs. Then the problem is how to find coefficients for such combination as well as the values of the community intensity over time.

Community factorization is a technique to find such parameters that give the best explanation of the observed data: i.e., if community graphs are combined together with their intensities as weighting factors, the combined graph should approximate the observed data. In the rest of this section, we formalize this community factorization as a factorization problem in the framework of constrained optimization.

3.2 Problem Formulation

In this section we formalize the above idea on community factorization.

Assume that there are n blogs b_i $(i = 1, \ldots, n)$ in the blogosphere. The linking activity in the blogosphere is aggregated as a graph structure $A_s \in \mathcal{R}_+^{n \times n}$ for each time window s $(s = 1, \ldots, t)$. These graphs are then stacked as a tensor $\mathcal{A}$. From the observed graph A_s, dense subgraphs are extracted. These graphs are also stacked as another tensor $\mathcal{B}$. Given $\mathcal{A}$ and $\mathcal{B}$, we will define a set of *community graphs* $\{C_l\}$ $(l = 1, \ldots, k)$ and *community intensities* $\{v_{sl}\}$ $(s = 1, \ldots, t)$. Each graph $C_l \in \mathcal{R}_+^{n \times n}$ represents how blogs communicate to each other within the community. The intensity $\{v_{sl}\}$ indicates how much the l-th community contributes at time s. We want to find k communities such that their community graphs and intensities best explain the observed data $\mathcal{A}$.

3.2.1 Data Tensor

In our analysis, the data representing the blogosphere over time is given as a tensor $\mathcal{A}$, which we call *data tensor*.

We define a link from b_i to b_j at time s $(i \neq j)$ when b_i publishes an entry at time s that has a hyperlink pointing to any content of b_j. These links are counted for each time window (say a day) and represented as an adjacency matrix A_s where $(A_s)_{ij}$ is the count of links from b_i to b_j in the s-th time window. The adjacency matrix A_s can be seen as a *snapshot* of activity in the blogosphere at time s, which we call a *snapshot graph*.

Stacking the adjacency matrices for the n time windows together, we have a 3-dimensional tensor

$$\mathcal{A} = [A_1, \ldots, A_t] \in \mathcal{R}_+^{n \times n \times t},$$

where the first two dimensions of $\mathcal{A}$ are blog indices (of citing blogs and cited blogs), and the third dimension of $\mathcal{A}$ is time window index.

3.2.2 Basis Tensor

For each snapshot graph A_s, we want to identify dense subgraphs where blogs communicate with each other. We can apply a graph partitioning algorithm, such as Shi's *normalized cut* [20] or Newman's *optimal modularity* [18, 24]. After removing insignificant subgraphs (e.g., a subgraph with only a couple of nodes), we have m_s graphs $B_{s1}, \ldots, B_{sm_s}$. We call them *basis subgraphs* since we will define a community as a linear combination of these subgraphs.

For the t time windows, we have in total $m = \sum_{s=1}^{t} m_s$ basis subgraphs. Stacking these basis subgraphs together, we get another 3-dimensional tensor

$$\mathcal{B} = [B_{11}, \ldots, B_{1m_1}, \ldots, B_{t1}, \ldots, B_{tm_t}] \in \mathcal{R}_+^{n \times n \times m},$$

and we call $\mathcal{B}$ the *basis tensor*. Essentially, the basis in the basis tensor capture all significant subgraphs at all time windows.

3.2.3 Community Graphs and Intensities

Since a community is established through communication, i.e., basis subgraphs, let us define a community graph C_l ($l = 1, \ldots, k$) as a linear combination of the basis subgraphs:

$$C_l = \sum_{p=1}^{m} u_{pl} B_p, \tag{1}$$

where u_{pl} is a weight that indicates how important the p-th basis subgraph is to the l-th community. In other words, u_{pl} indicates to what level a basis subgraph B_p (at some time window) belongs to the l-th community. The coefficients $\{u_{pl}\}$ are parameters we will need to estimate.

At time s, we see a snapshot graph A_s in the data. Note that multiple communities may have their communications at the same time since communities behave concurrently. Thus, multiple community graphs can affect the structure of A_s. We introduce the community intensity v_{sl} of the l-th community at time s such that

$$\sum_{l=1}^{k} v_{sl} C_l \tag{2}$$

represents the observed data A_s. Then our problem is formulated as minimization of the following error

$$\frac{1}{2} ||\mathcal{A} - \sum_{l=1}^{k} \mathcal{C}_l||^2 \tag{3}$$

where a tensor $\mathcal{C}_l$ is given as

$$\mathcal{C}_l = [v_{1l} C_l, \ldots, v_{tl} C_l] \in \mathcal{R}_+^{n \times n \times t} \tag{4}$$

Plugging Equations (1) and (4) into Equation (3), we can show that this formulation can be posed as the optimization problem to minimize the following objective function

$$J_1 = \frac{1}{2} ||\mathcal{A} - (\mathcal{B} \times_3 U) \times_3 V^T)||^2 \tag{5}$$

subjected to $U \in R_+^{m \times k}, V \in R_+^{t \times k}$. In the above formula, $\times_3$ represents the 3-mode multiplication of a tensor by a matrix.

The solution of this optimization is given next.

3.3 Solution

3.3.1 Solution by Non-negative Matrix Factorization

For the objective function (5), because only the 3-mode tensor product is involved, it can be equivalently written in a matrix form as

$$J_1 = \frac{1}{2} ||A - BUV^T||^2 \tag{6}$$

for $A \in \mathcal{R}_+^{n^2 \times t}$ and $B \in \mathcal{R}_+^{n^2 \times m}$. In the above objective function, A is obtained from the data tensor $\mathcal{A}$ in the following way: each column $A(s)$ of A is obtained by stacking the columns of the blog graph A_s into an $n^2 \times 1$ vector. B is obtained in a similar way from the basis tensor $\mathcal{B}$. Therefore, A and B in Equation (6) are given while our task is to search for non-negative matrices U and V that minimize J_1.

Once we have the solutions U and V to Equation (6), we can derive the j-th community from columns $U(j)$ and $V(j)$. Recall that a column $U(j)$ of U is a vector of weights on the basis subgraphs. Therefore, the community graph of the j-th community is given as the 3-mode multiplication of $\mathcal{B}$ by $U(j)$. At the same time, $V(j)$ represents the change of interaction intensity within the j-th community over time.

With U and V constrained to be non-negative matrices, this optimization problem falls into the category of *non-negative matrix factorization* (NMF) [15].

NMF has many advantages over general matrix factorization like principal component analysis (PCA). First, NMF decomposes data into the *addition* of several non-negative components. Such a nature of additive only (no subtractive) makes results of NMF easy to interpret. For example, Lee et al. [15] showed that, whereas traditional PCA decomposes the pictures of human faces into eigen-faces whose physical meaning is not clear, NMF is able to decompose the pictures into individual human parts such as eyes, nose, and mouth. Second, outputs of NMF can be *directly* used as the results for tasks such as document clustering. In comparison, one usually has to apply further post-processing steps, such as k-means, to the outputs of general matrix factorization. As demonstrated by Xu et al. [26], such post-processing steps are nonintuitive and sometimes do not give as good performance as directly using the outputs of NMF. Third, another feature of NMF is that it does not require orthogonality on outcomes. In comparison, for PCA, U and V must both have orthogonal columns. This feature of NMF is very appealing in our case because in our solution, the columns of V represent the temporal trends of different communities and it will be unreasonable to force these temporal trends to be orthogonal to each other. Similarly, a blog may have membership in multiple communities and it will be unreasonable to force the columns of U to be orthogonal.

3.3.2 Relation to Standard and Convex NMFs

Let us contrast our formulation against direct application of traditional NMF formula. The original NMF minimizes the objective function $||A - UV^T||^2$ as described in [15, 26] (which we call the *standard* NMF). Another form of NMF (which we call the *convex* NMF) with a different objective function $||A - AUV^T||^2$ has been proposed independently by Xu et al. [25] and Ding et al. [6]. By re-writing the objective function for the standard NMF as $||A - IUV^T||^2$, where I represents the identity matrix, we can generalize a family of NMF as $||A - XUV^T||^2$ for various X. Our community factorization chooses B instead of A or I.

For an application in information retrieval, the standard NMF and the convex NMF are meaningful in the following ways. The data is a *term-document* matrix A where the j-th column of A represents the terms occurred in the j-th document and the i-th row of A represents the documents that contain the i-th term. In the standard NMF, a concept is represented by a column of U, i.e., by a term vector. In the convex NMF, a concept is represented by a column of AU, i.e., by an additive combination of the documents themselves.

Because our data A represents graph structure, neither the standard NMF nor the convex NMF is directly applicable to our application. If we use standard NMF, then a community comprises arbitrary links and this ignores the relationship among links (e.g., e_1 and e_2 point to the same node); if we use the convex NMF, then a community must be a combination of snapshot graphs, which is obviously incorrect — a snapshot graph most likely contains communications of different communities at the same time. In contrast, in our setting, the links belonging to the same basis graph must be assigned together to a community graph.

3.3.3 Smoothing by Regularization

It is a common technique to incorporate prior knowledge into the objective function by introducing regularization terms. For this, we generalize Equation (6) by introducing Tikhonov regularization terms (*cf* [22]) as

$$J_2 = \frac{1}{2}||A - BUV^T||^2 + \frac{1}{2}\gamma_1||R_1U||^2 + \frac{1}{2}\gamma_2||R_2V||^2 \quad (7)$$

where γ_1 and γ_2 are user defined parameters. In this paper, we set γ_1 to be 1 and R_1 to be the identity matrix to regularize U. For V, we apply a simple piece of intuitive prior knowledge—the temporal trends, i.e., the column of V, should be smooth. That is, the value difference between two consecutive (in temporal order) elements in the same column of V should be small. For this purpose, we set R_2 to be a *difference* matrix

$$R_2 = \begin{pmatrix} 1 & -1 & 0 & 0 & \cdots & \cdots & \cdots & \cdots \\ -1 & 2 & -1 & 0 & \cdots & \cdots & \cdots & \cdots \\ 0 & -1 & 2 & -1 & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & 0 & -1 & 2 & -1 \\ \cdots & \cdots & \cdots & \cdots & 0 & 0 & -1 & 1 \end{pmatrix}$$

We will later use experimental studies to demonstrate the effect of tuning γ_2 on the smoothness of the results. Of course, we can also choose more sophisticate spline functions [11] instead of the above R_2. As can be seen, in general the matrices R_1 and R_2 (or any other regularization matrices) are not necessarily non-negative.

In general, regularization would be useful to incorporate other prior knowledge in our framework. For example, from the profile of a blogger we may know if she is a Democrat or Republican and this information can help determine the community preference of the blogger. As another example, we may be interested in communities that are very active during a specific period of time. That is, we have some preference for certain temporal trends. However, it is out of scope of this paper to demonstrate such general regularization.

3.4 Some Practical Issues

3.4.1 Size of Time Windows and Basis Subgraphs

In our algorithm, we cut data into snapshots according time windows and then apply graph partition algorithms on each snapshot to extract the basis subgraphs. Two practical issues arise: how to choose the size of time window and how to choose the size of basis subgraphs. As a matter of fact, given data over enoughly long period, our algorithm is not very sensitive to these two sizes, as long as they are not overly large. For example, when extracting communities from the blogosphere, we can aggregate blog linkage data by days or by weeks and for data in each time window, we can choose different numbers of basis subgraphs. No matter what window size we choose (the window size is not even necessarily uniform), as long as it is not too long, the fact that members of the same community behave in synchronization will be detected by our algorithm. For each snapshot, even if we cut a true community into small pieces (of course, not as small as a pair of nodes) and therefore put them as different basis subgraphs in the basis tensor, as long as these small pieces frequently co-occur in the data tensor, they will be picked up together by our algorithm to reconstruct the original true community. However, we should be cautious about overly long time window sizes and basis subgraphs sizes because they aggregate out the details about community behavior.

3.4.2 Number of Communities

Determining the right cluster number k is a difficult problem in clustering research that in many cases has no clear answers. In spectral clustering, one commonly used method is to sort the eigenvalues of the corresponding eigen-decomposition in an decreasing order and then pick the cluster number k where there is a large gap between the k-th and the *(k+1)*-th eigenvalues. The reasoning is that the *(k+1)*-th eigenvalues reflect the error introduced by keeping the top-k components in the eigen-decomposition [8]. In our case, because the non-linearity of NMF, we are not able to use this eigen-decomposition argument. However, we still can try different k's to compare the reconstruction error and then choose one that is reasonably small and at the same time explains data reasonably well.

4. COMPUTATION AND COMPLEXITY

We now give an algorithm to compute a solution to the optimization problem whose objective function is given by Equation (7), analyze the complexity of the algorithm, and prove the correctness of the algorithm.

4.1 Iterative Updating Rules

To solve for U and V in Equation (7), we start by setting U and V to some random non-negative matrices and then iteratively update U and V:

THEOREM 1. *The following multiplicative updating rules will converge to non-negative solutions to the optimization problem whose objective function is given by Equation (7)*

$$u_{ij} \leftarrow u_{ij}\sqrt{\frac{[B^TAV + \gamma_1(R_1^TR_1)_-U]_{ij}}{[B^TBUV^TV + \gamma_1(R_1^TR_1)_+U]_{ij}}} \quad (8)$$

$$v_{ij} \leftarrow v_{ij}\sqrt{\frac{[A^TBU + \gamma_2(R_2^TR_2)_-V]_{ij}}{[VU^TB^TBU + \gamma_2(R_2^TR_2)_+V]_{ij}}} \quad (9)$$

where $(R_1^TR_1)_+$, $(R_2^TR_2)_+$, $(R_1^TR_1)_-$, and $(R_2^TR_2)_-$ are the non-negative matrices that represent the positive and negative parts of $R_1^TR_1$ and $R_2^TR_2$, respectively.

4.2 Proof for the Updating Rules

Now we prove that the updating rules given in Equations (8) and (9) converge to the solutions to U and V in Equation (7). We start with the general optimization on quadratic form. Assuming we want to find the non-negative solution $\vec{y}$ to minimize the following quadratic form

$$J_3(\vec{y}) = \frac{1}{2}\vec{y}^TC\vec{y} + \vec{d}^T\vec{y} \quad (10)$$

Ding et al. [6] showed the following lemma.

LEMMA 1 (DING). *The following iterative update procedure will converge to the solution $\vec{y}$*

$$y_i \leftarrow y_i\sqrt{\frac{(C_-\vec{y} + \vec{d}_-)_i}{(C_+\vec{y} + \vec{d}_+)_i}} \quad (11)$$

Equipped with this lemma, now we prove Theorem 1.

PROOF OF THEOREM 1. We first rewrite the objective function as

$$\begin{aligned}J_2 =& \frac{1}{2}||A - BUV^T||^2 + \frac{1}{2}\gamma_1 \cdot ||R_1U||^2 + \frac{1}{2}\gamma_2 \cdot ||R_2V||^2 \\ =& \frac{1}{2}Tr(A^T - VU^TB^T)(A - BUV^T) \\ &+ \frac{1}{2}\gamma_1 \cdot Tr(U^TR_1^TR_1U) + \frac{1}{2}\gamma_2 \cdot Tr(V^TR_2^TR_2V) \\ =& \frac{1}{2}Tr(A^TA - 2A^TBUV^T + VU^TB^TBUV^T) \\ &+ \frac{1}{2}\gamma_1 \cdot Tr(U^TR_1^TR_1U) + \frac{1}{2}\gamma_2 \cdot Tr(V^TR_2^TR_2V)\end{aligned}$$

Defining the vectorization, $\vec{u}, \vec{v}$, of U, V by stacking the columns of U and V, and ignoring the constant term $\frac{1}{2}Tr(A^TA)$, we can rewrite J_2 as the quadratic form for $\vec{u}$ and for $\vec{v}$, respectively. Then it can be easily shown that for $\vec{u}$

$$\begin{aligned}C &= \gamma_1 \cdot I_k \otimes (R_1^TR_1) + (V^TV) \otimes (B^TB) \\ \vec{d} &= vec(-B^TAV)\end{aligned}$$

and for $\vec{v}$

$$\begin{aligned}C &= \gamma_2 \cdot I_k \otimes (R_2^TR_2) + (U^TB^TBU) \otimes I_n \\ \vec{d} &= vec(-A^TBU)\end{aligned}$$

where $\otimes$ is the Kronecker product and vec is the vectorization operation (see, e.g., [10]). Plugging the above forms into Equation (11), we can get the updating rules in Equations (8) and (9). □

It is worth noting that in Equations (8) and (9) we assume that A and B are non-negative, which is true in our application. However, even if this assumption is not true, because the updating rule (11) applies to general quadratic forms (i.e., when C and $\vec{d}$ are not necessarily non-negative), we still can derive from Equation (11) multiplicative updating rules that are similar to (only slightly different from) Equations (8) and (9), which converge to non-negative solutions U and V.

4.3 Complexity Analysis

For an algorithm to be applicable to real problems in the blogosphere, it should be scalable to hundreds of millions of blogs. We now discuss some aspects of our proposed algorithm.

First, our algorithm requires the basis tensor $\mathcal{B}$ to be computed. This should not be a bottleneck, for there exist very efficient algorithms for graph partitioning. As an example, there exists a greedy algorithm for optimal modularity [4] that is essentially linear in the size of the graph. Second, in the updating equations (8) and (9), the most time and memory consuming parts are B^TA and B^TB. However, this should not be a serious problem either. First, both A and B are extremely sparse matrices—the number of non-zero elements in A is the same as the number of links in the original data and the number of non-zero elements in B is less than that in A (because B is a partition of A). In addition, although A and B are relatively large, B^TA and B^TB are relatively small. Furthermore, we only have to compute B^TA and B^TB once and then use the results throughout all the iterations.

The algorithm is an iterative one. The overall complexity is the number of iteration times the complexity of one iteration. The number of iterations largely depends on the numerical precision. The complexity of one iteration can be analyzed as follows. Let l be the number of links in the graph. Based on the model, we have $l \geq m \geq t \geq k$. Because of the sparsity of matrices A and B, the complexity of A^TB and B^TA is $O(lmt)$ and the complexity of B^TB is $O(lm^2)$. After computing these terms, the rest of the matrix multiplication are all of lower dimensionality, therefore the complexity is covered by $O(lm^2)$. Thus, the total computation complexity of one iteration is $O(lm^2)$. Therefore, our algorithm should be scalable to large number of blogs in the blogosphere.

5. EXPERIMENTAL STUDIES

In this section, we perform experimental studies by applying our technique to discovering communities from three sets of data. We first use a synthetic data and controlled studies to illustrate some good properties of our technique. The second data set is a blog data set obtained by an in-house crawler developed at NEC Laboratories America. This data set contains a small number of blogs among which there have been intensive interactions over a very long period of time. We will show that our technique discovers many interesting communities that are not detectable by traditional methods. Finally, we study our technique by using a large scale benchmark blog data set. On one hand we demonstrate that our technique is scalable to such a large data set, and on the other hand, we reveal a weak point of our technique when the time period is extremely short.

5.1 Synthetic Data Set

We design this experiment to demonstrate that our algorithm can separate two overlapped communities that have different temporal trends. The data set contains 150 blogs that belong to two communities. Each community contains 100 blogs and therefore there are 50 blogs that participate in both of the two communities. We let the intensities of interaction within the two communities vary following sinusoid trends with different phases (as will be revealed in Figure 2), over 100 consecutive time windows. Figures 1(a) and 1(b) show the adjacency matrix of the blog graph aggregated over all the 100 time windows in a 3D plot and a 2D plot. Figures 1(c) and 1(d) show the adjacency matrices of two communities discovered by applying Shi's Normalized Cut algorithm to the aggregated blog graph. As can be seen, the community result is incorrect.

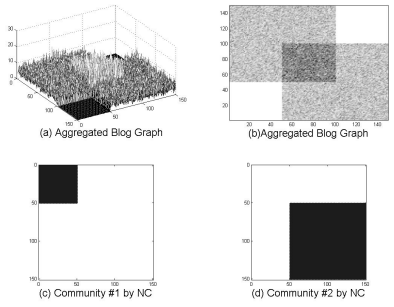


Figure 1: Synthetic data: (a) and (b), the aggregated blog graph; (c) and (d), two communities discovered by the Normalized Cut algorithm

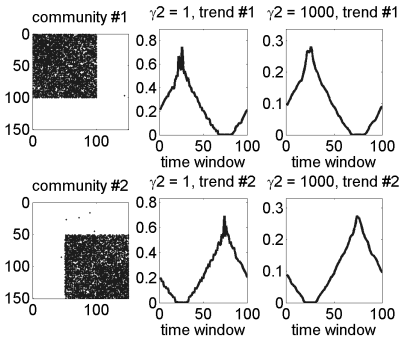


Figure 2: Synthetic data: two communities together with their trends extracted by using our algorithm

Figure 2 shows the two communities discovered by our algorithm. The communities are given as weighted graphs that have overlap in the membership (see Figure 3 for the actual weights in one of the discovered communities). Figure 2 also shows the temporal trends obtained by using $\gamma_2 = 1$ and $\gamma_2 = 1000$, respectively. As can be seen, both the community structure and the temporal trends are discovered accurately. Also can be seen from the figure, when γ_2 is set to 1000 and therefore the regularization term has more weight, the obtained temporal trends become smoother as expected. When γ_2 is increased from 1 to 1000, the factorization error, $||A - BUV^T||$, did not increase much (from 252 to 259), meaning that the introduction of the regularization term did not cause negative effects on the factorization quality.

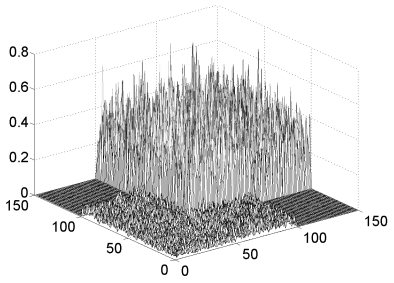


Figure 3: Synthetic data: soft community membership

Next, we perform an experiment to demonstrate that our algorithm is not too sensitive to the variation of window size. We aggregate data in the previous study in the following way — in each step, we generate a random number p between 1 and 3 and aggregate the next p time windows into a single one; we repeat this process until all 100 time windows are used. Then we apply our algorithm on this aggregated data. Figure 4 shows the results (with trends V unwrapped according to the aggregation). Compared with Figure 2, we can see that the main communities and trends are both successfully recovered, although with a little worse quality.

5.2 NEC Data Set

At NEC Laboratories America, we have built a focused crawler on blogs. Given seeds of technology-related blogs, the crawler discovered blogs that are densely connected with the seeds, resulting in an expanded set of blogs that com-

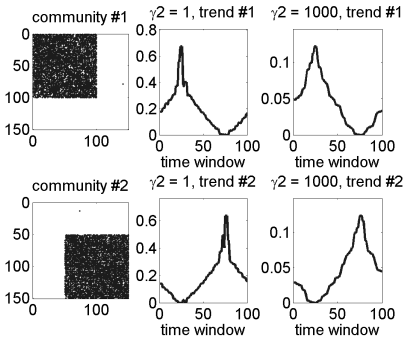


Figure 4: Synthetic data: two communities together with their trends extracted by using our algorithm, with varying time window sizes

municate with each other. The crawler then continued monitoring for new entries over a long time period. From this data set, we use 407 English blogs that have 274,679 entries in 441 days (63 weeks) between July 10th in 2005 and September 23rd in 2006. These entries are connected with 148,681 links. Expanded from the seed set on technology, the data set actually contain roughly two groups of blogs — one with technology focus and another with politics focus. Figure 5 shows the blog graph for this NEC data set,

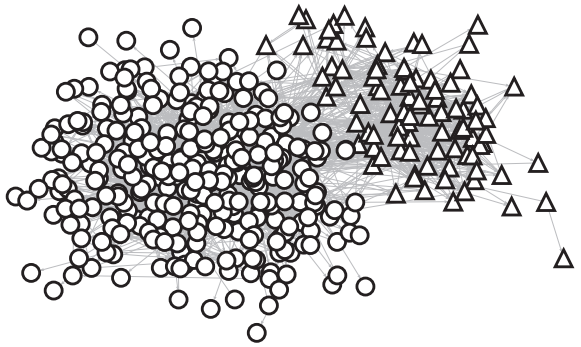


Figure 5: The blog graph for the NEC data

where the layout is determined by the Pajek software using the Kamada-Kawai algorithm. Here we can see two groups: blogs with technology focus are placed as circles in the left region of the graph, and blogs with politics focus are placed as triangles in the right region of the graph.

We first try extracting communities from the aggregated blog graph by using traditional algorithms. Given the graph in Figure 5, we could imagine two large static communities on technology and politics. However, it is hard to see more details as well as the temporal dynamics. By using the Normalized Cut algorithm, we split the aggregated blog graph into 50 clusters (communities). Then for the temporal trends of the communities, we report the number of links among each community every day. Figure 6 shows some of the representative results. We did not find any useful information in this result.

We now apply our algorithm on this data set to discover the same number of 50 communities. The number 50 has been set rather arbitrarily, although we have tested other numbers such as 20 and 100 and obtained similar results.

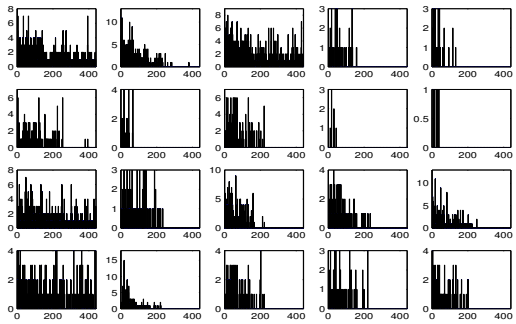


Figure 6: Temporal trends of communities extracted by a traditional method

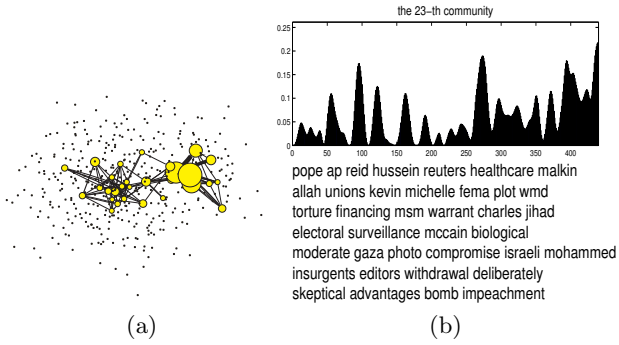


Figure 7: Community related to politics in general

In Figures 7-10, we show some representative communities that have been discovered. To visualize each community, we display the community graph on the left side and the community intensity over the 441 days on the right side.

In the community graph, the size of a node and the width of a link reflect how important that blog and the interaction between that pair of blogs are in the community. The size of a node is determined by the corresponding row sum in the community graph C_l as defined in Equation (1) and the width of a link is determined by the corresponding entry in C_l. In addition, to give readers a high-level idea on the types of community members, we fix the coordinates of all the blogs in the graphs to the same position as they are in Figure 5 (i.e., technology blogs in the left region and politics blogs in the right region).

In addition, we use the content of the blog entries to validate our discovered communities. Note that our algorithm depends purely on structural (hyperlink) information, not on content of blogs. However, intuitively, a valid community should have coherent topics discussed and consistent vocabulary used among community members. Therefore in this experiment, we extract top keywords from each community to see if they form a coherent set. Keywords are ranked with the frequency within the community divided by the one within the entire data set.

Now we illustrate several sample communities that are discovered by our algorithm. Please note that in almost all the 50 communities discovered by our algorithm, we are able to detect meaningful trends and coherent topics. In the following discussion, we only show several more well-known communities.

In the first community, which is shown in Figure 7, the main community members include several well-known political blogs[3] that interacted heavily. This community does not have any major spike. Instead, the intensity fluctuates as political events happen over the whole period of study. Note that *Michelle Malkin* is the name of a famous political blogger.

The second community, which is shown in Figure 8, is formed around an authoritative blog by David Sifry, the CEO of one of the top blog search engines — Technorati. The peak in the temporal trend happened on October 17, 2005. On that day, David Sifry posted in his blog site a comprehensive study on the current status of the blogosphere[4] which includes the number of blogs and entries tracked by Technorati, the speed of growth of the blogosphere, the spam blogs in the blogosphere, the comparison of blogs with mainstream media, and so on. This report was one of the most authoritative studies on the blogosphere and it has been cited and discussed extensively immediately after it had been released.

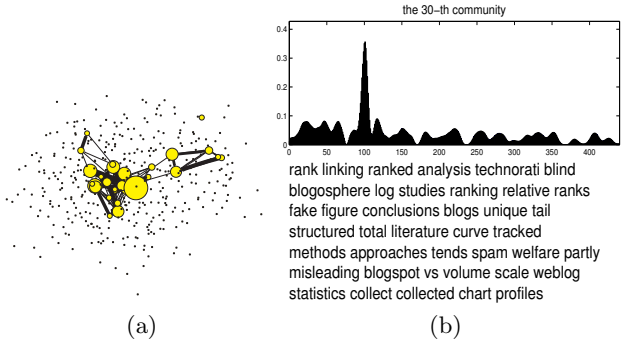


Figure 8: Community related to the status of the blogosphere

The peak in the temporal trend of the third community, as shown in Figure 9, happened on February 1, 2006, when the newspapers in some European countries republished the

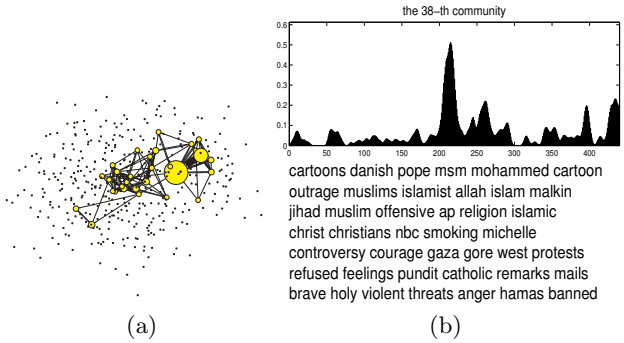


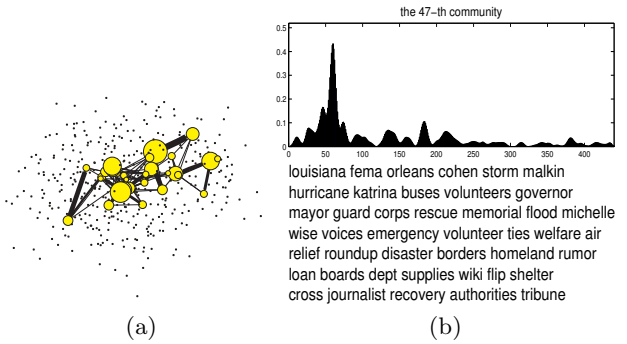
Figure 9: Community related to the Muhammad cartoons controversy

[3]http://www.washingtonmonthly.com/
http://michellemalkin.com/
http://ezraklein.typepad.com/blog/

[4]http://www.sifry.com/alerts/archives/2005_10.html

controversial Muhammad cartoons and therefore triggered widespread protests all over the world. As can be seen, the main members of this community belong to the political group of blogs.

The forth community, as shown in Figure 10, is related to the hurricane Katrina, which happened at the end of August 2005. As can be seen, due to the severe aftermath of the hurricane and due to its high political impact, the community members are more diversified — they are distributed all over the blogosphere.



(a) (b)

Figure 10: Community related to hurricane Katrina

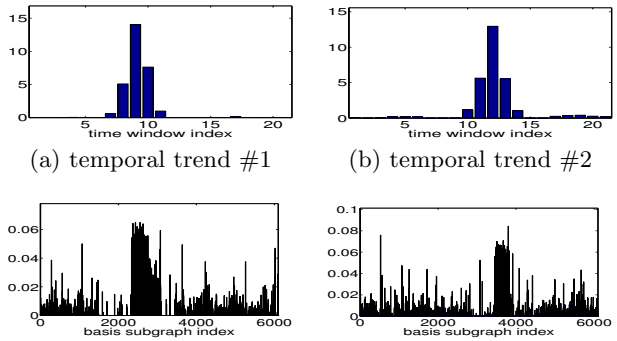
In summary, for this NEC data set, our algorithm is able to detect meaningful communities together with their temporal trends. Top representative keywords in each of the discovered community reveal coherent topics.

5.3 Benchmark Data Set

We apply our community extraction algorithm on the benchmark data set provided by the organizers of the WWW 2006 Workshop on the Weblogging Ecosystem. Compared with the NEC data set, this workshop dataset is of much larger scale. It contains 8.37 million entries from 1.43 million different blog sites during a 3-week period between July 4th and July 24th, 2005. Because our community extraction algorithm is link-based, instead of the whole data set, we studied the subset of blogs that contain at least one link. By this restriction, we are able to narrow down the number of blogs to around 141K, where the number of links among this set of blogs are 1.62 million.

We use this data set to illustrate two points about our algorithm. First, we show that our algorithm is scalable to data set with large number of blogs and links. As will be reported in the next subsection, our algorithm can handle this large data set very efficiently by using a modest desktop PC running Matlab codes.

More importantly, we want to use this data set to demonstrate a weak point of our algorithm. That is, in order to discover communities that show consistency over time, we need data for a long period. For this benchmark data, there are hundreds of thousands of blogs, but there are only 21 time windows. As a result the community number cannot be too large. We set the number of community to be 10 and run our algorithm on the benchmark data. In Figure 11 we show two sample communities that are discovered whereas all other eight communities follow similar patterns. As can be seen from the figure, the temporal trends for the communities are just local spikes at different time windows. In the lower panels of the figure, we show the U vectors for the two communities. As can be seen, each community consists most of local basis subgraphs in the time window where the community peaks. In other words, each community essentially picks the whole snapshot graph of one time window. The temporal trends also tell us that the snapshot graphs at consecutive time windows have some correlation. However, other than this trivial information, our algorithm is not able to discover meaningful communities.



(a) temporal trend #1 (b) temporal trend #2

(c) column of U, #1 (d) column of U, #2

Figure 11: Two sample communities extracted from the benchmark data: (a) and (b), temporal trends of the two communities; (c) and (d), the columns of U corresponding to the two communities

When the number of blogs n is large and the number of time windows t is small, it is difficult for our algorithm to extract meaningful communities. Similar weak points exist in applying NMFs to traditional information retrieval area. That is, when the number of documents is too few, a better explanation to all the documents is to take each individual document as a concept. In our case, instead of individual groups of blogs that form interactive communities, a better explanation to the observed data (in terms of low reconstruction error) is to take some snapshot graphs as communities. In conclusion, our method should be applied to data over a longer time period (i.e., t/n is large), and the traditional approach with aggregated graphs should be applied to data within a short time period (i.e., t/n is small).

5.4 Running Time

In Table 1 we report the running time of our algorithm on the three data sets. Our algorithm is implemented in Matlab and runs on a PC of Pentium IV processor with 2G Hz CPU and 2GB memory. The codes are implemented in Matlab. The reported running time is in *second per iteration*. When the criterion for convergence is set to be that $|J_2(t) - J_2(t-1)| < 10^{-6}$, the algorithm can converge with in 1000 iterations for all the three cases. As can be seen from the running time, our algorithm scales nicely to the size of the data both in terms of the number of blogs and the number of links.

Table 1: Running time for the 3 data sets

Data Set Name	Blog Count	Link Count	Window Count	Running Time (sec)
Synthetic	150	207,709	100	1.23
NEC	407	148,681	441	0.86
Benchmark	141,046	1,622,428	21	1.66

6. CONCLUSION AND FUTURE WORK

The blogosphere has a unique structure: a series of short-term subgraphs representing the communication among blogs. In this paper, we proposed a novel technique that capture the structure and temporal dynamics of communities from the blogosphere. In our framework, a community is a set of blogs that communicates with each other in a synchronized manner, triggered by some events (such as a news article). The community is represented by a *community graph*, which indicates how often one blog communicates with another, and *community intensities*, which indicate the activity level of the community over time. Our method, *community factorization*, extracts such communities from the communication among blogs that is observed as subgraphs (i.e., threads of discussion). We formalized this as a factorization problem in the framework of constrained optimization.

Experimental studies were conducted on both synthetic and real blog data. With synthetic data, we demonstrated that our technique is able to identify two overlapping communities that have different temporal dynamics. In NEC data set, our technique was able to discover meaningful communities that are not detectable by traditional methods. Through the WWW Workshop benchmark data set, we pointed out that our technique is not appropriate for data with a very short time period. Finally, we demonstrated that our algorithm scales nicely to handle a large number of blogs.

We plan to extend our research in several directions. First, the technique proposed in this paper is based on link analysis only and not combined with content analysis. We plan to incorporate content information into our framework by extending the tensors with keywords as one more dimension so that it can capture multiple topics of interaction among blogs. Second, our technique adopted Shi et al.'s normalized cut algorithm for partitioning local blog graphs as *undirected* graphs. We are also interested in using *directed* graph partitioning algorithms since the direction of links in the blogosphere is important.

7. REFERENCES

[1] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: membership, growth, and evolution. In *Proc. of the 12th ACM SIGKDD Conference*, 2006.

[2] D. Chakrabarti, R. Kumar, and A. Tomkins. Evolutionary clustering. In *Proc. of the 12th ACM SIGKDD Conference*, 2006.

[3] Y. Chi, B. L. Tseng, and J. Tatemura. Eigen-trend: Trend analysis in the blogosphere based on singular value decompositions. In *Proc. of the 15th CIKM Conference*, 2006.

[4] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70:066111, 2004.

[5] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM J. on Matrix Analysis and Applications*, 21(4), 2000.

[6] C. Ding, T. Li, and M. Jordan. Convex and semi-nonnegative matrix factorizations for clustering and low-dimension representation. Technical Report LBNL-60428, Lawrence Berkeley National Laboratory, 2006.

[7] G. Flake, S. Lawrence, and C. Giles. Efficient identification of web communities. In *Proc. of the 6th ACM SIGKDD Conference*, 2000.

[8] G. Golub and C. V. Loan. *Matrix Computations*. Johns Hopkins University Press, third edition, 1996.

[9] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *Proc. of the 13th WWW Conference*, 2004.

[10] D. A. Harville. *Matrix Algebra From a Statistician's Perspective*. Springer, first edition, 2000.

[11] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, first edition, 2003.

[12] H. Ino, M. Kudo, and A. Nakamura. Partitioning of web graphs by community topology. In *Proc. of the 14th WWW Conference*, 2005.

[13] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins. On the bursty evolution of blogspace. In *Proc. of the 12th WWW Conference*, 2003.

[14] R. Kumar, J. Novak, and A. Tomkins. Structure and evolution of online social networks. In *Proc. of the 12th ACM SIGKDD Conference*, 2006.

[15] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401, 1999.

[16] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proc. of the 11th ACM SIGKDD Conference*, 2005.

[17] Q. Mei, C. Liu, H. Su, and C. Zhai. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In *Proc. of the 15th WWW Conference*, 2006.

[18] M. E. J. Newman. Modularity and community structure in networks. *Proc. Natl. Acad. Sci.*, 2006.

[19] A. Qamra, B. L. Tseng, and E. Y. Chang. Mining blog stories using community-based and temporal clustering. In *Proc. of the 15th CIKM Conference*, 2006.

[20] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8), 2000.

[21] J. B. T. Falkowski and M. Spiliopoulou. Mining and visualizing the evolution of subgroups in social networks. In *Proc. of the IEEE WI Conference*, 2006.

[22] G. Wahba. *Spline Models for Observational Data*. Society for Industrial and Applied Mathematics, 1990.

[23] X. Wang and A. McCallum. Topics over time: A non-markov continuous-time model of topical trends. In *Proc. of the 12th ACM SIGKDD Conference*, 2006.

[24] S. White and P. Smyth. A spectral clustering approach to finding communities in graph. In *SDM*, 2005.

[25] W. Xu and Y. Gong. Document clustering by concept factorization. In *Proceedings of the 27th Annual International ACM SIGIR Conference*, 2004.

[26] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th Annual International ACM SIGIR Conference*, 2003.