

Incremental Spectral Clustering With Application to Monitoring of Evolving Blog Communities

Huazhong Ning* Wei Xu[†] Yun Chi[†] Yihong Gong[†] Thomas Huang*

Abstract

In recent years, spectral clustering method has gained attentions because of its superior performance compared to other traditional clustering algorithms such as K-means algorithm. The existing spectral clustering algorithms are all off-line algorithms, i.e., they can not incrementally update the clustering result given a small change of the data set. However, the capability of incrementally updating is essential to some applications such as real time monitoring of the evolving communities of websphere or blogsphere. Unlike traditional stream data, these applications require incremental algorithms to handle not only insertion/deletion of data points but also similarity changes between existing items. This paper extends the standard spectral clustering to such evolving data by introducing the *incidence vector/matrix* to represent two kinds of dynamics in the same framework and by incrementally updating the eigenvalue system. Our incremental algorithm, initialized by a standard spectral clustering, continuously and efficiently updates the eigenvalue system and generates instant cluster labels, as the data set is evolving. The algorithm is applied to a blog data set. Compared with recomputation of the solution by standard spectral clustering, it achieves similar accuracy but with much lower computational cost. Close inspection into the blog content shows that the incremental approach can discover not only the stable blog communities but also the evolution of the individual multi-topic blogs.

Keywords: *Incremental clustering, Spectral Clustering, Incidence Vector/Matrix, Web-blogs*

1 Introduction

Spectral clustering is notable both for its theoretical basis of graph theory and for its practical success. It recently has many applications in data clustering, image segmentation, Web ranking analysis and dimension reduction. Spectral clustering can handle very complex

and unknown cluster shapes in which cases the commonly used methods such as K -means and learning a mixture model using EM may fail. It relies on analyzing the eigen-structure of an affinity matrix, rather than on estimating an explicit model of the data distribution [14, 16]. In other words, the top eigenvectors of the graph Laplacian can unfold the data manifold to form meaningful clusters [21].

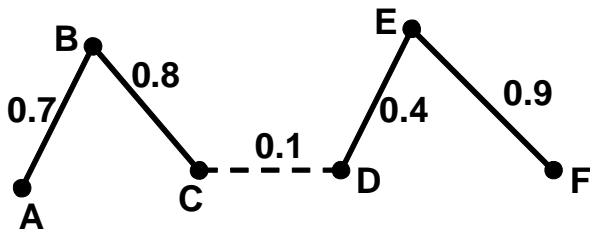
However, existing spectral algorithms are all off-line algorithms, hence they cannot be directly applied to dynamic data set. Therefore, to handle evolving data set, e.g., web data, there is a need to develop efficient algorithms for inductive spectral clustering to avoid expensive recomputation of the solution from the scratch. An intuitive approach is fixing the graph on the training data and assigning new test points to their corresponding clusters by the nearest neighbor in the training data [21]. However, the error will be accumulated quickly when more test points that are close to the cluster boundary are added. This paper extends the spectral clustering to handle evolving data by incrementally updating the eigenvalue system, which achieves more accurate results while requires low computational cost.

There exist many previous incremental clustering algorithms [9, 8, 2] that are designed to handle only insertion of new data points. However, data sets such as web pages and blogs, require the incremental algorithms to handle not only insertion/deletion of nodes but also similarity changes between existing items. Figure 1 gives a toy example where a graph evolves from (a) to (b), with a similarity change of 0.5 added to the edge CD and a new node G connected to node F . In Figure 1(a), the graph should be cut at the edge CD ; while in Figure 1(b) the cut edge is DE due to the similarity change on edge CD .

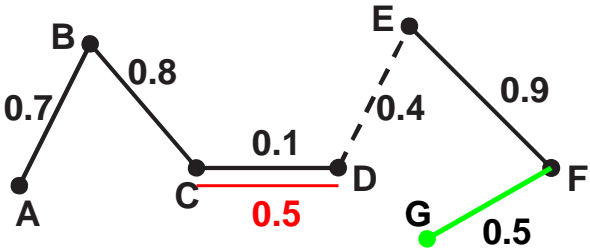
We handle the two kinds of dynamics in the same framework by representing them with the *incidence vector/matrix* [1]. The Laplacian matrix can be decomposed into the production of two incidence matrixes and a similarity change can be regarded as an incidence vector appended to the original incidence matrix and an insertion/deletion of a data point is divided into a se-

*Dept. of ECE, U. of Illinois at Urbana-Champaign, Urbana, IL 61801. {hning2,huang}@ifp.uiuc.edu.

[†]NEC Laboratories America, Inc., 10080 N. Wolfe Road, Cupertino, CA 95070. {xw,ychi,ygong}@sv.nec-labs.com.



(a)



(b)

Figure 1: A toy example of evolving data. (a) Before evolution; (b) After evolution. The dash lines are the cut edges.

quence of similarity changes. Each newly added incidence vector (similarity change) may induce increment to the Laplacian and degree matrixes, and we approximate the corresponding increments of the eigenvalues and eigenvectors by omitting the influence of the data points outside the spatial neighborhood of the updating data points. In this way, the eigen-system and the cluster labels are incrementally updated as the new test points or new incidence vectors are added.

This approach is very useful to the applications in which the similarity matrix is sparse while both the data points and their similarities are dynamically updated. An example is the community discovery of the web-blogs. The key observation is that a link reference from an entry of a source blog to an entry of a destination blog serves as an endorsement of the similarity between the two blogs. A graph can be constructed based on the similarities of the web-blogs and communities (clusters) can be discovered by spectral clustering. However, web-blogs are evolving, and new blogs and new links are added or removed every day. Therefore standard spectral clustering cannot be used to online monitor the web-blogs because of the huge number of blogs and, in turn, of the high computational cost. For sparse similarity matrix, Lanczos method [7] may save much

cost to solve the eigenvalue problem. But it is still impractical to recompute the solution from the scratch at each time instance the data set is updated, especially when the web-blogs are huge. On the contrast, our approach applied on the web-blogs data achieves similar accuracy but with much lower computational cost, compared with recomputation by the standard spectral clustering.

2 Related Work

Spectral clustering evolved from the theory of spectral graph partitioning, an effective algorithm in high performance computing [5]. Recently there is a huge volume of literature on this work. Ratio cut objective function [17, 10] naturally captures both mincut and equipartition, the two traditional goals of partitioning. This function leads to eigenvalue decomposition of the Laplacian matrix. Shi and Malik [16] proposed a normalized cut criterion that measures both the total dissimilarity between the different groups as well as the total similarity within the groups. The criterion is equivalent to a generalized eigenvalue problem. Ding *et al.* presented a min-max cut [6] and claimed that this criteria always leads to a more balanced cut than the ratio cut and the normalized cut. Unlike the above approaches, Ng *et al.* [14] proposed a multi-way clustering method. The data points are mapped into a new space spanned by the first k eigenvectors of the normalized Laplacian. Clustering is then performed with traditional methods (like k -means) in this new space. However, none of the above algorithms are designed to incrementally cluster dynamic data.

However, there exists a large research literature on incremental clustering using techniques other than spectral methods. Most of such algorithms are mainly designed to cluster streaming data where each data record can be examined only once. They usually dynamically update the cluster centers [9], medoids [8], or a hierarchical tree [2] when new data points are inserted. However, they did not consider the scenario of similarity updating, like the evolving web data. Therefore, recent incremental algorithms on web data gain more and more attention, of which the PageRank metric has gained enormous popularity with the success of Google [4]. For example, Desikan *et al.* [4] and Langville *et al.* [11] exploit the underlying principle of the first order markov model on which PageRank is based, to incrementally compute PageRank for the evolving Web graph. Instead of PageRank, this paper presents an efficient updating algorithm to cluster evolving data, like web-blogs, with dynamic similarities as well as insertion/deletion of data points.

3 Basic Formulation

Given a weighted graph $G = G(E, V)$ with node set V , edge set E and similarity matrix W where w_{ij} indicates the similarity of node v_i and v_j , spectral clustering partitions the graph into two or more disjoint subgraphs. Since our approach is based on, but not limited to, the normalized cut [16], this algorithm is briefly reviewed using the notation in the tutorial [5].

3.1 Normalized Cut In this paper, the similarity matrix W is assumed to be symmetric with $w_{ii} = 1$ and $0 \leq w_{ij} \leq 1$ when $i \neq j$. Denote the degree matrix as $D = \text{diag}\{d_1, d_2, \dots, d_n\}$ where

$$d_i = \sum_j w_{ij},$$

and Laplacian matrix as $L = D - W$. Let the similarity between the subgraphs A and B be

$$s(A, B) = \sum_{i \in A} \sum_{j \in B} w_{ij}.$$

The normalized cut aims to minimize the criteria function

$$J_{Ncut}(A, B) = \frac{s(A, B)}{d_A} + \frac{s(A, B)}{d_B}$$

where

$$d_A = \sum_{i \in A} d_i, d_B = \sum_{i \in B} d_i.$$

After some manipulations, the criteria function can be rewritten as

$$(3.1) \quad \frac{\mathbf{q}^T L \mathbf{q}}{\mathbf{q}^T D \mathbf{q}}$$

where \mathbf{q} is the cluster indicator and satisfies

$$(3.2) \quad \mathbf{q}^T D \mathbf{q} = 1, \mathbf{q}^T D \mathbf{1} = 0,$$

and ideally

$$(3.3) \quad q_i = \begin{cases} \sqrt{d_B/d_A d} & \text{if } i \in A \\ -\sqrt{d_A/d_B d} & \text{if } i \in B \end{cases}$$

where $d = \sum_{i \in V} d_i$. If \mathbf{q} is relaxed to take real value, we can minimize Eqn. 3.1 by solving the generalized eigenvalue system

$$(3.4) \quad L \mathbf{q} = \lambda D \mathbf{q}.$$

3.2 Incidence vector/matrix As we mentioned before, in many real applications both the data points and their similarities are dynamic. Then a question arises: how to represent two kinds of dynamics in the same

framework and how to feed them into the eigenvalue system without violating the original representation? We solve this problem by introducing the *incidence vector/matrix*.

DEFINITION 3.1. An **incidence vector** $\mathbf{r}_{ij}(w)$ is a row vector with only two nonzero elements: i -th element equal to \sqrt{w} and j -th element $-\sqrt{w}$, indicating data point i and j having a similarity w .

DEFINITION 3.2. An **incidence matrix** R is a matrix with each row is an incidence vector.

An incidence vector can be rewritten as $\mathbf{r}_{ij} = \sqrt{w} \mathbf{u}_{ij}$ where \mathbf{u}_{ij} (adopted from [19]) is a row vector with only two nonzero elements: i -th element equal to 1 and j -th element -1 . w can be negative if $\sqrt{-1}$ is allowed. The definition of incidence vector/matrix in this paper is partly different from the traditional definition [1] because our emphasis is to extend the incidence matrix to incorporate similarity changes by appending incidence vectors.

PROPOSITION 3.1. If $L = D - W \in \mathbb{R}^{n \times n}$ is a Laplacian matrix, then there exists an incidence matrix R such that $L = R^T R$ [3]. In addition, R is a stack of all the incidence vectors $\mathbf{r}_{ij}(w_{ij}), i < j$, in any order, i.e., $R \in \mathbb{R}^{\frac{n(n-1)}{2} \times n}$.

Proof. Regardless of the order in which the incidence vectors stacked in R , the product

$$(3.5) \quad \begin{aligned} R^T R &= \sum_{1 \leq i < j \leq n} \mathbf{r}_{ij}(w_{ij})^T \mathbf{r}_{ij}(w_{ij}) \\ &= \sum_{1 \leq i < j \leq n} w_{ij} \mathbf{u}_{ij}^T \mathbf{u}_{ij}. \end{aligned}$$

But,

$$\mathbf{u}_{ij}^T \mathbf{u}_{ij} = \begin{pmatrix} \vdots & & \vdots & & \\ \cdots & 1 & \cdots & -1 & \cdots \\ \vdots & & \vdots & & \\ \cdots & -1 & \cdots & 1 & \cdots \\ \vdots & & \vdots & & \end{pmatrix}$$

Add all of the matrixes $w_{ij} \mathbf{u}_{ij}^T \mathbf{u}_{ij}, i < j$ together and it follows that,

$$\sum_{1 \leq i < j \leq n} w_{ij} \mathbf{u}_{ij}^T \mathbf{u}_{ij} = L$$

Firstly we consider a single similarity change. With the decomposition in Proposition 3.1, the increments

of D and L , induced by a similarity change (i.e., an incidence vector), can be deduced as follows. Suppose there is a similarity change Δw_{ij} between data points i and j , then the new incidence matrix \tilde{R} can be the old R appended by the incidence vector $\mathbf{r}_{ij}(\Delta w_{ij})$,

$$\tilde{R} = \begin{pmatrix} R \\ \mathbf{r}_{ij}(\Delta w_{ij}) \end{pmatrix} = \begin{pmatrix} R \\ \sqrt{\Delta w_{ij}} \mathbf{u}_{ij} \end{pmatrix},$$

so the new Laplacian matrix \tilde{L} can be written as,

$$\tilde{L} = \tilde{R}^T \tilde{R} = R^T R + \Delta w_{ij} \mathbf{u}_{ij}^T \mathbf{u}_{ij}$$

or the increment of L as

$$(3.6) \quad \Delta L = \Delta w_{ij} \mathbf{u}_{ij}^T \mathbf{u}_{ij},$$

and the increment of the degree matrix D is

$$(3.7) \quad \Delta D = \Delta w_{ij} \text{diag}\{\mathbf{v}_{ij}\}.$$

where \mathbf{v}_{ij} is a row vector with i -th and j -th elements equal to 1 and other elements equal to 0.

As to insertion/deletion of a data point, it can be regarded as a sequence of similarity changes appended to the original incidence matrix. For instance, when a data point i , which has similarities $w_{ij_1}, w_{ij_2}, \dots, w_{ij_k}$, is added, it is equivalent to a sequence of similarity changes of $w_{ij_1}, w_{ij_2}, \dots, w_{ij_k}$ occurring in any order, corresponding to a sequence of incidence vectors $\mathbf{r}_{ij_1}(w_{ij_1}), \mathbf{r}_{ij_2}(w_{ij_2}), \dots, \mathbf{r}_{ij_k}(w_{ij_k})$.

4 Incremental Spectral Clustering

In Section 3.2, any updating of the dynamic data is equivalent to a (or a sequence of) incidence vector(s) $\mathbf{r}_{ij}(w)$ appended to the original incidence matrix R . Here we approximate the increments of the eigenvalues and eigenvectors in the spectral clustering, induced by the updating $\mathbf{r}_{ij}(w)$. The approximation is carried on, but not limited to, the generalized eigenvalue system of the normalized cut, $L\mathbf{q} = \lambda D\mathbf{q}$ [16].

4.1 Increment of Eigenvalue $\Delta\lambda$ There is a closed-form solution to the eigenvalue increment of a symmetric generalized eigenvalue system.

PROPOSITION 4.1. *Suppose $A\mathbf{x} = \lambda B\mathbf{x}$ is a generalized eigenvalue system where both $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times n}$ are symmetric, then the perturbation of λ in terms of the perturbations of A and B is*

$$(4.8) \quad \Delta\lambda = \frac{\mathbf{x}^T (\Delta A - \lambda \Delta B) \mathbf{x}}{\mathbf{x}^T B \mathbf{x}}$$

Proof. Differentiate both sides of the generalized eigenvalue system $A\mathbf{x} = \lambda B\mathbf{x}$,

$$(4.9) \quad \Delta A \mathbf{x} + A \Delta \mathbf{x} = \Delta \lambda B \mathbf{x} + \lambda \Delta B \mathbf{x} + \lambda B \Delta \mathbf{x}.$$

Left multiply both sides by \mathbf{x}^T and obtain

$$(4.10) \quad \mathbf{x}^T \Delta A \mathbf{x} + \mathbf{x}^T A \Delta \mathbf{x} = \Delta \lambda \mathbf{x}^T B \mathbf{x} + \lambda \mathbf{x}^T \Delta B \mathbf{x} + \lambda \mathbf{x}^T B \Delta \mathbf{x}.$$

Since

$$\mathbf{x}^T A = \lambda \mathbf{x}^T B$$

because A and B are symmetric, Eqn. 4.10 can be rewritten as

$$\mathbf{x}^T \Delta A \mathbf{x} = \Delta \lambda \mathbf{x}^T B \mathbf{x} + \lambda \mathbf{x}^T \Delta B \mathbf{x}.$$

After a few manipulations, we obtain Eqn. 4.8.

Suppose the updating is the incidence vector $\mathbf{r}_{ij}(\Delta w_{ij}) = \sqrt{\Delta w_{ij}} \mathbf{u}_{ij}$. Substitute ΔL in Eqn. 3.6, ΔD in Eqn. 3.7 and \mathbf{q} for ΔA , ΔB and \mathbf{x} in Eqn. 4.8 respectively, then $\Delta\lambda$ of the generalized eigenvalue system $L\mathbf{q} = \lambda D\mathbf{q}$ is

$$\Delta\lambda = \frac{\mathbf{q}^T (\Delta w_{ij} \mathbf{u}_{ij}^T \mathbf{u}_{ij} - \lambda \Delta w_{ij} \text{diag}\{\mathbf{v}_{ij}\}) \mathbf{q}}{\mathbf{q}^T D \mathbf{q}}.$$

But

$$\begin{aligned} \mathbf{q}^T \mathbf{u}_{ij}^T \mathbf{u}_{ij} \mathbf{q} &= (q_i - q_j)^2, \\ \mathbf{q}^T \text{diag}\{\mathbf{v}_{ij}\} \mathbf{q} &= q_i^2 + q_j^2, \end{aligned}$$

and

$$\mathbf{q}^T D \mathbf{q} = 1$$

because of the normalization in Eqn. 3.2, $\Delta\lambda$ can be expressed as

$$(4.11) \quad \Delta\lambda = \Delta w_{ij} ((q_i - q_j)^2 - \lambda(q_i^2 + q_j^2))$$

$\Delta\lambda$ can be further simplified if we assume that \mathbf{q} is ideal as in Eqn. 3.3, two clusters has nearly the same degrees ($d_A \approx d_B$ in Section 3.1), and $\lambda \ll 1$ (it usually holds for the top smallest eigenvalues),

$$(4.12) \quad \Delta\lambda \approx \begin{cases} -2\lambda \frac{\Delta w_{ij}}{d} & i, j \text{ in the same cluster} \\ 4 \frac{\Delta w_{ij}}{d} & i, j \text{ in different clusters} \end{cases}$$

From the approximation, increasing the similarity of two points in the same cluster decreases the eigenvalue λ , and λ increases if they are in different clusters and in addition the increase is much greater in magnitude than the decrease because usually the top smallest eigenvalues $\lambda \ll 1$.

4.2 Increment of Eigenvector $\Delta\mathbf{q}$ Generally the increment of a eigenvector $\Delta\mathbf{q}$ can be solved by Power Iteration or Lanczos method [7]. These methods run very fast on sparse matrixes, but it is still impractical for huge matrixes such as web data (see Section 5). Fortunately, clustering requires only discrete cluster labels and precise eigenvectors are not necessary. Thus we adopt an approximate approach to compute $\Delta\mathbf{q}$ as fast as it can be applied to huge and evolving data, partly at the expense of accuracy.

Substitute ΔL in Eqn. 3.6, ΔD in Eqn. 3.7 and \mathbf{q} for ΔA , ΔB and \mathbf{x} in Eqn. 4.9 respectively and after some manipulations, we obtain a linear equation for $\Delta\mathbf{q}$

$$(4.13) \quad K\Delta\mathbf{q} = \mathbf{h}$$

where

$$(4.14) \quad K = L - \lambda D,$$

$$(4.15) \quad \mathbf{h} = (\Delta\lambda D + \lambda\Delta D - \Delta L)\mathbf{q}$$

Since ΔL , ΔD and $\Delta\lambda$ are known according to Eqn. 3.6, Eqn. 3.7 and Eqn. 4.11, $\Delta\mathbf{q}$ can be solved by

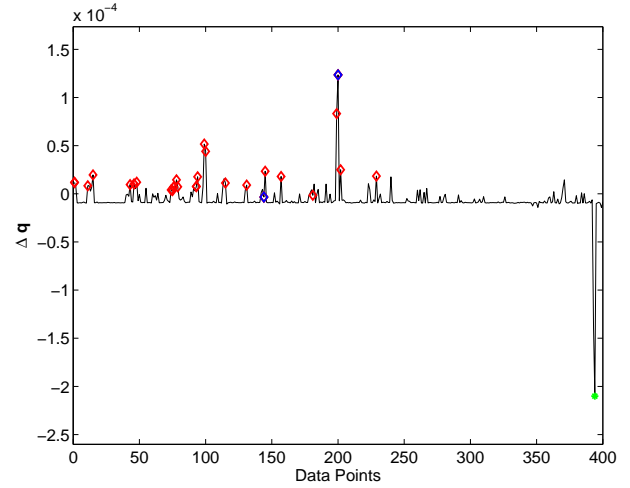
$$(4.16) \quad \Delta\mathbf{q} = (K^T K)^{-1} K^T \mathbf{h}$$

if $K^T K$ is nonsingular.

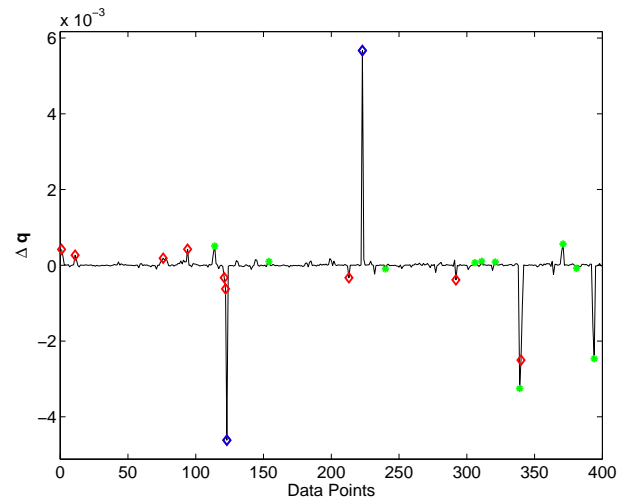
However, it is impractical to directly apply Eqn. 4.13 or 4.16 to solve $\Delta\mathbf{q}$. Firstly, K and $K^T K$ are singular because $L\mathbf{q} = \lambda D\mathbf{q}$, i.e., $K\mathbf{q} = 0$. Secondly, and more importantly, the size of K and $K^T K$ are huge for large data sets which means that solving Eqn. 4.13 requires very high computational cost, even higher than that of the Lanczos method [7].

Therefore, we adopt an approximate approach by fully exploiting the properties of the application, and even partly at the expense of the accuracy of $\Delta\mathbf{q}$ since only discrete cluster labels are needed. As we know, a similarity change of two data points i and j has little influence on the data points far from i and j . In other words, to maintain the cluster structure of the graph, only i and j and a few points close to them need to be considered. Figure 2 gives two examples of $\Delta\mathbf{q}$ induced by a similarity change on the data set of Web-Blogs (see Section 6). Here the eigenvectors are solved by Lanczos method [7] and $\Delta\mathbf{q}$ by Eqn. A-2 in *Appendix A*. In the figure, points i and j are marked blue, a red mark indicates that the point is directly adjacent to i or j , and a green mark means a leaf in the graph. It shows that most of the spikes are either adjacent to i or j or leaves. The issue of leaves will be discussed in Section 5.

Given a similarity change $r_{ij}(\Delta w_{ij})$, let $N_{ij} = \{k | w_{ik} > \tau \text{ or } w_{jk} > \tau\}$ be the spatial neighborhood



(a)



(b)

Figure 2: $\Delta\mathbf{q}$ induced by a similarity change of two points (marked blue). A red mark indicates that the point is directly adjacent to the two points; and a green mark means a leaf in the graph. (a) Two points are in the same cluster. (b) Two points are in different clusters.

of both i and j , where τ is a predefined threshold which can take 0 for sparse data set. In accordance with the above observations, we assume $\Delta q_k = 0$ if $k \notin N_{ij}$ and eliminate them from $\Delta\mathbf{q}$, and accordingly the corresponding columns in K are also omitted. The left form $\Delta\mathbf{q}_{ij}$ and $K_{N_{ij}}$. Thus we obtain

$$(4.17) \quad \Delta\mathbf{q}_{ij} = (K_{N_{ij}}^T K_{N_{ij}})^{-1} K_{N_{ij}}^T \mathbf{h}$$

Since the number of the columns of $K_{N_{ij}}$ is very small compared with the size of the data set, computational

cost of the above equation is very low.

4.3 The Incremental Algorithm Summarize Section 4.1 and 4.2, and the incremental spectral clustering consists of the following steps.

ALGORITHM 4.1. (INCREMENTAL SPECTRAL CLUSTERING)

1. Suppose at time t , the data set grows large enough and the similarity matrix W , degree matrix D and Laplacian matrix L are available.
2. Solve Eqn. 3.4 as standard spectral clustering does for eigenvectors with the smallest eigenvalues. This solution and the matrixes (D and L) serve as the initialization.
3. From then on, when a similarity change occurs, use Eqn. 4.11 and 4.17 to update the eigenvalues and eigenvectors and Eqn. 3.6 and 3.7 to update D and L .
4. If a data point is added or deleted, it is regarded as a sequence of similarity changes and step 3 is repeatedly conducted.
5. After T similarity changes occur, re-initialize the spectral clustering by repeating from step 2, so as to avoid the big accumulated errors.

5 Discussions on the Algorithm

Firstly, the time complexity is discussed. The generalized eigenvalue system can be transformed into a standard eigenvalue problem of $D^{-\frac{1}{2}}LD^{-\frac{1}{2}}\mathbf{q} = \lambda\mathbf{q}$, and solving a standard eigenvalue problem takes $O(n^3)$ operations [16], where n is the number of data points. When n is as large as in the Web applications, this is impractical. Fortunately, Lanczos method [7] can greatly reduce the computational cost if the similarity matrix W is sparse. And Web application is such an example. The time complexity of the Lanczos algorithm is $O(n^{\frac{3}{2}})$ if the special properties of W is fully exploited [16]. This is also the running time of the baseline system in Section 6.

However, the computational cost is still very high if the data set is large and undergoes frequent evolution as the Web-blogs be. In this case, it is hard for a standard spectral clustering to update the cluster labels immediately after the data set is updated. On the contrast, our incremental approach may success. It needs constant running time to compute $\Delta\lambda$ and $O(\bar{N}^2n) + O(\bar{N}^3) + O(\bar{N}n) + O(\bar{N}^2)$ to compute $\Delta\mathbf{q}$, where \bar{N} is the average size of the spatial neighborhood of a node, $O(\bar{N}^2n)$ is needed to compute $A = K_{N_{ij}}^T K_{N_{ij}}$

in Eqn. 4.17, $O(\bar{N}^3)$ for inversion of A , $O(\bar{N}n)$ for $b = K_{N_{ij}}^T \mathbf{h}$, and $O(\bar{N}^2)$ for $A^{-1}b$. In the Web applications, \bar{N} is usually constant, so the running time of the incremental approach is $O(n)$. The \bar{N} can be adjusted by tuning the threshold τ , so the time complexity is tunable to some extent at the expense of the accuracy.

Secondly, the influence of the leaves on the eigenvectors need to be explained. Suppose node i is a leaf that is connected to node j and j is only connected to node k and i . As discussed in Appendix B, q_i , q_j , Δq_i and Δq_j may be spikes (see Figure 2). However, their influence on other nodes may decay greatly because they have only one edge connected the other part of the graph. Furthermore, since the leaves usually do not lie on the boundaries, i.e., they are usually not supporting vectors, they should be very trivial in graph partitioning. Therefore, we ignore their influence in Eqn. 4.17.

It is important to point out the limitations of our incremental approach and their possible solutions.

- 1) The error is accumulating though growing slowly. This may be a critical problem to many incremental algorithms. We use re-initialization to avoid a collapse.
- 2) When the data set grows larger or evolves more frequently, e.g., if all the blogs on the internet are considered, the $O(n)$ complexity probably still makes the algorithm fail. One possible $O(1)$ solution is to propagate the influence of a similarity change to its spatial neighborhood and then update the eigenvector according to the received influence.

6 Experiments

Our algorithm is designed to handle the sparse (or nearly sparse) data with two kinds of dynamics in the same framework. Web-blogs are such kind of data. Recently web-blogs have become a prominent media on the internet that enables the users (bloggers) to publish, update and access the personalized content. Bloggers frequently interact with each other by trackback, adding comments to, or by referring to the entries in other blogs. The reference links, comments, and trackback grow continuously, new blogs may be created, and old blogs may be deleted. Therefore web-blogs are evolving with the time. Simultaneously, the virtual blog communities maybe emerge through the bloggers' interactions and evolve as the web-blogs updating.

There is a volume of literature on the research of *virtual community* [15, 12]. Rheingold [15] defined *virtual community* as "social aggregations that emerge from the Net when enough people carry on those public discussions long enough, with sufficient human feeling, to form webs of personal relationship in cyberspace". It should involve long term and meaningful conversations

in cyberspace which suggests sustained membership [12]. In other words, many factors, such as time decaying, direction of reference, awareness of trackback, and so on, should be considered when measuring the relationship between two blogs. To save our approach from being overwhelmed by the details of blogs, we simply measure the similarity between two blogs i and j by

$$(6.18) \quad w_{ij} = e^{-\frac{1}{\beta l_{ij}}}$$

where any interaction between i and j serves as an endorsement to the similarity, l_{ij} measures the number of interactions or links, and β controls the marginal effect of the endorsement when more interactions occur. When further interactions occur, the similarity increases more slowly, and finally approaching 1. We aim to discover the blog communities and their evolution through the incremental spectral clustering.

6.1 Data description The blog data were collected by the NEC laboratories American, using a blog crawler. Starting from some seed blogs, the crawler continuously crawls their RSS feeds and then the corresponding entries [12]. The blog entries are analyzed and the hyperlinks embedded in the content are extracted. Some “hot” blogs become new seed blogs when they meet some criteria.

The data were crawled from July 10th 2005 to April 30th 2006, for 42 consecutive weeks. We use the subset written in English, that consists of 489 blogs and totally 75,614 links. The self-reference links are removed since they do not contribute to the interactions in the communities. And 14,510 links remain effective, averagely 30 effective links for each blog. Figure 3 shows the number of remaining links created in each week.

6.2 Comparison with the baseline We use the standard normalized cut [16] as a baseline that is implemented using the Lanczos method [7]. We start from the 23th week and the similarity matrix W , degree matrix D and Laplacian matrix L are built on the links of the previous 22 weeks (10,246 links in total). Then the Algorithm 4.1 is applied to the data set and the clusters are continuously updated as more links are added from Week 23 to 42 (4,264 links in total). The cluster number is chosen as 3 after close inspection of the initial eigenvalues and eigenvectors. Automatic selection of cluster number can be referred to [20]. To make comparison, the baseline – standard normalized cut re-solves the generalized eigenvalue system at each time instance when any link is added (L and D is updated by Eqn. 3.6 and 3.7).

We use Eqn. A-2 and A-3 in *Appendix A* to compute

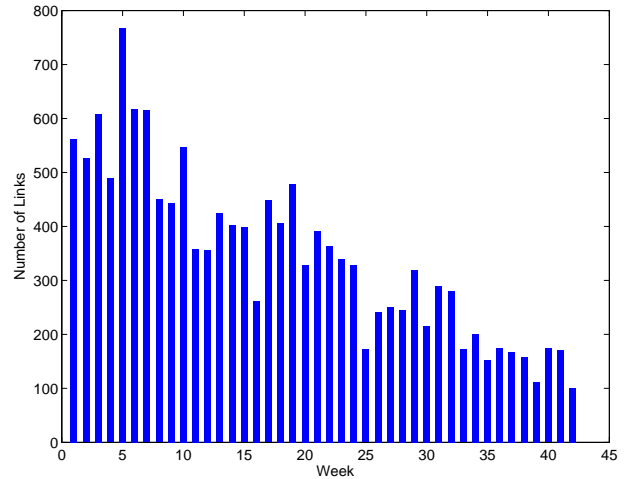


Figure 3: The number of effective links in each week in NEC Blog Data.

the error $E_{dif}(\mathbf{q}_{inc}, \mathbf{q}_{bas})$ and difference $dif(\mathbf{q}_{inc}, \mathbf{q}_{bas})$ between the two sets of eigenvectors generated by the incremental approach (\mathbf{q}_{inc}) and the baseline (\mathbf{q}_{bas}) respectively, after each link is added. Figure 4 shows the differences of the eigenvectors with the second smallest eigenvalue (the smallest is 0), right after Week 27, 32, 37 and 42. The points marked green are leaves which cover almost all of the salient spikes. Figure 5 shows the error E_{dif} of the corresponding eigenvectors (two biggest leaf-node spikes are removed because otherwise the two nodes contribute nearly half of the error). The error is accumulating to 0.0666 as more links are added, and averagely equal to 0.0223. Considering that the eigenvectors are normalized to one before the errors are computed, these errors are actually very small. The relative error of the eigenvalue λ by the incremental approach with the baseline as “true measurement” is illustrated in Figure 6. The relative error is 2.77% on average and accumulating to reach maximum 5.20%.

The above comparison only reveals that our incremental solution to the generalized eigenvalue system closely approximates the solution by the baseline. However, it is important to compare the clustering results. Cluster labels are obtained by discretizing the eigenvectors as [16] did. Given a data point (blog) i , let l_i and \hat{l}_i be the labels generated by the baseline and the incremental approach respectively. The following metric [18] is defined to measure the accuracy,

$$AC = \max_{map} \frac{\sum_i^n \delta(l_i, map(\hat{l}_i))}{n}$$

where n denotes the total number of data points, $\delta(x, y)$

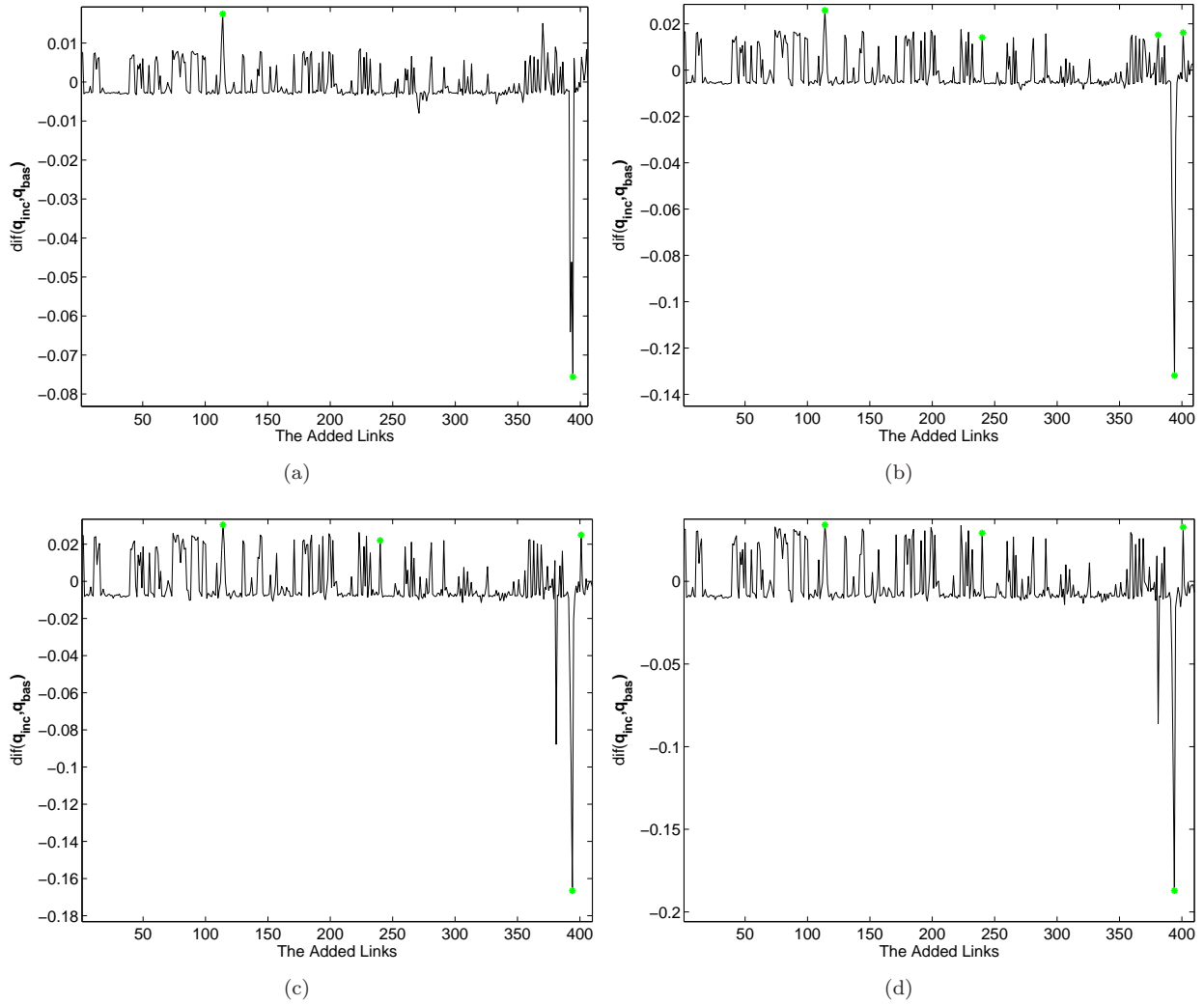


Figure 4: The difference $dif(\mathbf{q}_{inc}, \mathbf{q}_{bas})$ of the eigenvectors with the second smallest eigenvalue after (a) 27, (b) 32, (c) 37, (d) 42 weeks. Green marks indicate leaf nodes.

is the delta function that gives one when $x = y$ otherwise zero, and $map(\hat{l}_i)$ is a mapping function of labels. The optimized mapping function can be found by the Kuhn-Munkres algorithm [13] in polynomial time. Figure 7 shows the accuracy corresponding to each added link which is 98.68% on average. The accuracy drops slowly as more links are added, and reaches the minimum 96.83%, i.e., about 10 blogs are clustered differently from the baseline.

Besides the accuracy, the computational cost is also compared. Both the incremental approach and the baseline are implemented in MATLAB. The time cost of adding each link is recorded for both systems, which is plotted in Figure 8. It shows that the computational cost for the baseline is much higher than that of the

incremental approach, and averagely it is 0.7328 seconds for the former and 0.0415 seconds for the latter. It is expected that the difference is bigger for larger data sets, because the computational cost for the incremental approach is $O(n)$ while it is $O(n^{\frac{3}{2}})$ for the baseline.

6.3 Blog Communities The incremental spectral clustering is applied to the NEC web-blog data set after Week 22, with the output of the baseline as initialization. The method discovers three main communities. The communities are basically stable from Week 23 to Week 42, i.e., both the membership and topic are roughly sustained during this period. However, some individual blogs may jump among the communities as the data evolving. The incremental algorithm can capture

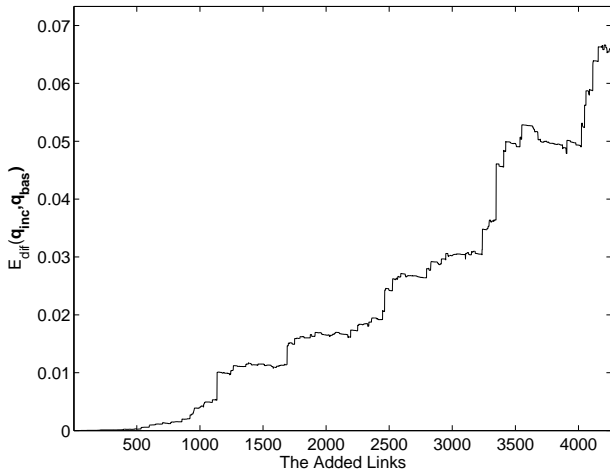


Figure 5: The error $E_{diff}(\mathbf{q}_{inc}, \mathbf{q}_{bas})$ of the eigenvectors with the second smallest eigenvalue.

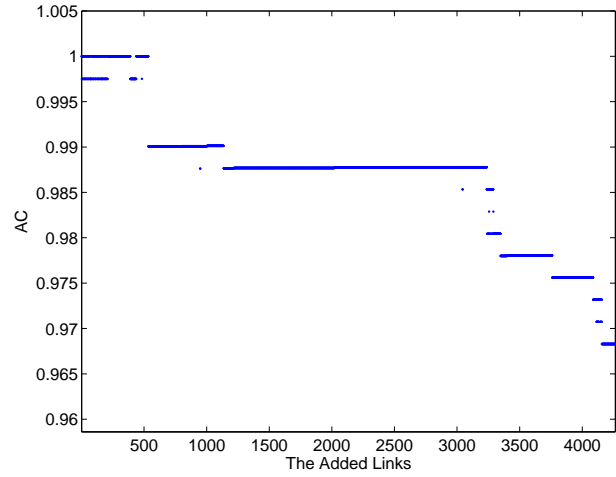


Figure 7: The accuracy corresponding to each added link.

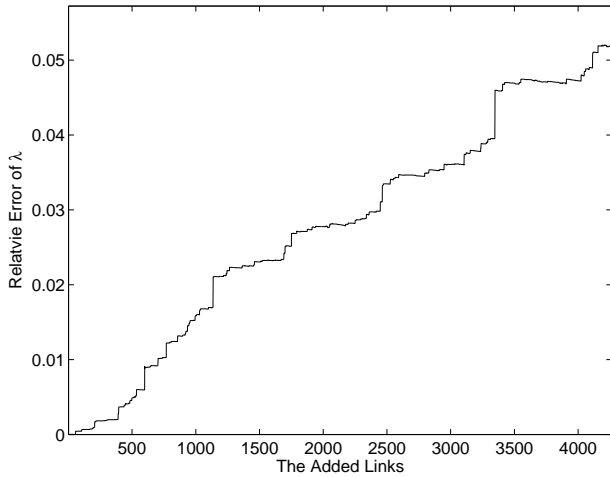


Figure 6: The relative error of the eigenvalue λ by the incremental approach.

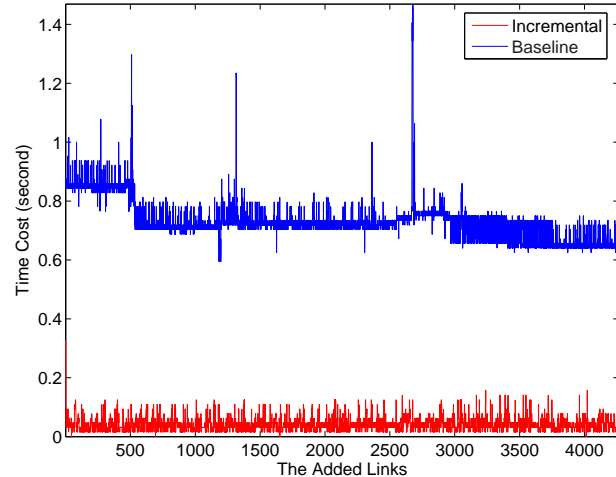


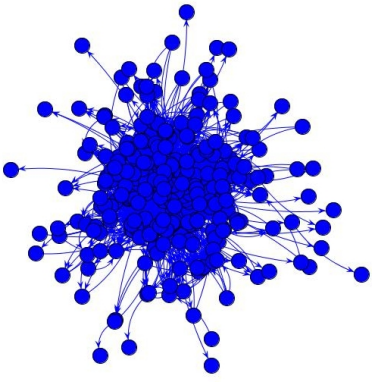
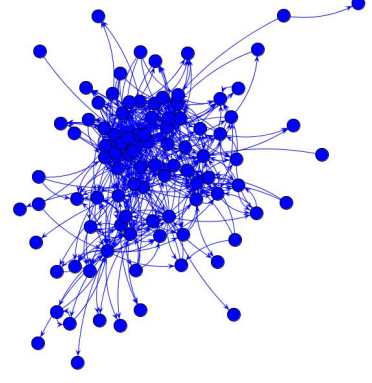
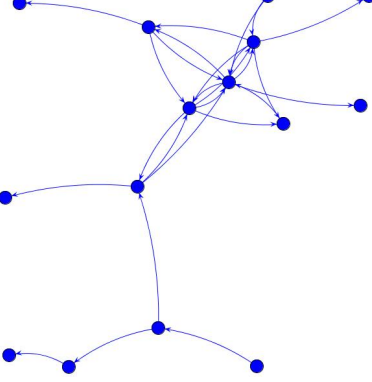
Figure 8: Time cost of adding each link for both the incremental approach and the baseline.

the evolution.

First we use the content of the blog entries to validate the discovered communities. Since the algorithm simply depends on the structural (hyperlink) information, the content validation is meaningful. We extract the keywords with top relative frequency from each community to validate whether they form a consistent cluster [12]. Here the relative frequency $f_r(w)$ of a keyword w is the ratio of its frequency $f'(w)$ in the content of its community to its frequency $f(w)$ in the entire data set, i.e., $f_r(w) = f'(w)/f(w)$. Unlike the absolute frequency $f'(w)$ in a community, relative frequency can reduce the influence of the internal language frequency

of a keyword. The validation is done every three weeks and the top keywords basically remain stable. Table 1 shows the subgraph of each community and its corresponding keywords at Week 28. The topics can be clearly identified from the keywords. The three communities focus on technology&business, politics and culture respectively. The technology&business community is mainly discussing high technologies and their commercials which are often talked at the same time on the internet and it is hard to separate them. More interestingly, the third community is mainly discussing a small topic of culture – library, after carefully checking the content of its blogs.

Table 1: Discovered communities.

Subgraph	Keywords
 <p>Tech&business community</p>	<p>fleshbot, betanews, sfist, torrone, lifehacker, threaded, middot, vonage, cingular, adwords, username, engadget, sprint, psp, bluetooth, nokia, phillip, powerbook, macromedia, verizon, usb, adsense, lcd, widget, tivo, tuaw, sms, voip, cellphones, businessweek, myspace, samsung, aol, feedburner, plasma, wifi, wireless, logged, hd, ndash, skype, xbox, apis, api, ipod, shuffle, nano, yahoo, os, gps, newsgator, cellphone, mozilla, sf, mobile, flash, dept, inch, blackberry, apps, mac, icio, gadget, linking, keyboard, ads, phones, beta, subscribers, interface, bookmarks, apple, motorola, startup, ebay, itunes, opml, advertisers, google, enhanced, newest, ajax, porn, firefox, messenger, messaging, offerings, desktop, hp, portable, acquisition, publishers, password, networking, xp, seth, robot, silicon, gadgets, broadband</p>
 <p>Politics community</p>	<p>shanghaiist, ebffed, atta, uzbekistan, uranium, irs, niger, shanghai, liberals, islam, saudi, iranian, loan, sheehan, tax, gop, beijing, opposition, fitzgerald, elections, reform, cia, arab, valerie, plame, muslims, arabia, danger, muslim, islamic, hong, novak, democrats, pakistan, iran, partisan, palestinian, liberal, immigration, conservatives, abortion, democrat, corruption, cindy, saddam, democratic, rove, msm, indictment, wilson, libby, constitutional, terror, ambassador, democracy, syria, withdrawal, regime, republicans, presidential, congressional, propaganda, corps, sunni, hussein, voters, israeli, deputy, kong, taiwan, russia, hearings, iraq, terrorism, miers, republican, election, constitution, afghanistan, pentagon, clinton, scandal, taxes, iraqis, troops, liberty, senate, israel, fema, conservative, minister, fbi, civil, parliament, nuclear, foreign, kerry, nomination, administration, china</p>
 <p>Culture community</p>	<p>libraries, library, gaming, copyright, stephen, academic, wiki, spyware, lawsuit, circuit, chicago, learning, brands, games, teach, skills, cultural, classes, complaint, teaching, staff, dance, presentation, vendors, brand, anybody, commons, amp, conferences, marketers, culture, colleagues, corporation, print, survey, students, eric, distance, game, consumers, desk, contract, access, computing, collection, books, titles, flickr, knowledge, conference, searching, student, authors, tech, county, trends, permission, registration, activities, java, learned, fair, buttons, letters, cases, deeply, amendment, engines, keyword, drm, privacy, copies, collaboration, practices, speakers, school, celebrity, taught, resources, practical, audience, seth, marketing, context, training, motion, xml, websites, boss, courts, define, studies, job, communities, database, fiction, community, association, chat, players</p>

Although the topics of the three communities are roughly stable, some individual blogs may bounce among them. We select out such blogs and carefully read their contents. We found that they usually cover more than one topics or change their topics at some time. So call them “multi-topic blogs”. The URLs and the discovered topics of some multi-topic blogs are listed in Table 2. Among them, The first blog in the Table 2

was created by Rebecca MacKinnon, a research Fellow at the Law School’s Berkman Center for Internet and Society, who focuses on three main subjects: the future of media in the internet age, freedom of speech online, and the internet in China. These subjects can be labelled as high technology and politics. The 4th blog in Table 2, created by David Mattison, focuses on digital libraries, digital collections and digital preser-

Table 2: List of multi-topic blogs. Topic 1: tech&business; Topic 2: politics; Topic 3: culture

No.	URL	Topic
1	http://rconversation.blogs.com/rconversation/	1, 2
2	http://blog.ericgoldman.org/	1, 3
3	http://www.josalmon.co.uk	1, 2
4	http://www.davidmattison.ca/wordpress	1, 3
5	http://www.cultureby.com/trilogy/	1, 3
6	http://www.joegratz.net	1, 3

vations. So it is assigned to the culture community (library subtopic) or the high technology community. When during a specific period the multi-topic blogs refer to or are referred by more blogs in one community, they are prone to be clustered into that community. The incremental approach can basically capture this evolution. The baseline may also obtain it by recomputing the eigenvalue system for each updating, but the former is much more efficient. Note that the identification of multi-topic blogs can be regarded as a byproduct of our approach.

7 Conclusions

This paper presented an incremental approach for spectral clustering to handle dynamic data. Two kinds of dynamics, insertion/deletion of data points and similarity change of existing data points, are incorporated in the same framework by representing them with *incidence vector/matrix*. The incremental algorithm, initialized by a standard spectral clustering, continuously and efficiently updates the eigenvalue system and generates instant cluster labels, as the data set is evolving. The algorithm is applied to the Web-blogs data. Compared with recomputation by standard spectral clustering, it achieves similar accuracy but with much lower computational cost. Close inspection into the blog content shows that the incremental approach can discover not only the stable blog communities but also the evolution of the individual multi-topic blogs.

Appendix A. Difference of Two Eigenvectors

An eigenvector is subject to a scalar, i.e., if \mathbf{q} is a eigenvector, then $c\mathbf{q}$ is also a eigenvector for any constant $c \neq 0$. Therefore, $\mathbf{q}_1 - \mathbf{q}_2$ cannot serve as a proper difference. We define an *error* that is invariant to scale and measures the difference of the two eigenvectors,

$$(A-1) \quad E_{dif}(\mathbf{q}_1, \mathbf{q}_2) = \min_{\gamma} \left\| \gamma \frac{\mathbf{q}_1}{\|\mathbf{q}_1\|} - \frac{\mathbf{q}_2}{\|\mathbf{q}_2\|} \right\|^2,$$

and define the difference as

$$dif(\mathbf{q}_1, \mathbf{q}_2) = \gamma \frac{\mathbf{q}_1}{\|\mathbf{q}_1\|} - \frac{\mathbf{q}_2}{\|\mathbf{q}_2\|}.$$

Table 3: Influence of λ on η .

λ	0	0.020	0.040	0.060	0.080	0.100
η_i	1.000	1.309	1.845	2.995	7.1483	-24.972
η_j	1.000	1.211	1.571	2.327	5.022	-15.686

Differentiate the right side of Eqn. A-1 and set the differentiation to 0, then γ can be solved,

$$\gamma = \frac{\mathbf{q}_1^T \mathbf{q}_2}{\|\mathbf{q}_1\| \|\mathbf{q}_2\|}.$$

Then, the difference and the *error* can be rewritten as,

$$(A-2) \quad dif(\mathbf{q}_1, \mathbf{q}_2) = \frac{(\mathbf{q}_1^T \mathbf{q}_2) \mathbf{q}_1}{\|\mathbf{q}_1\|^2 \|\mathbf{q}_2\|} - \frac{\mathbf{q}_2}{\|\mathbf{q}_2\|}$$

$$(A-3) \quad E_{dif}(\mathbf{q}_1, \mathbf{q}_2) = 1 - \frac{(\mathbf{q}_1^T \mathbf{q}_2)^2}{\|\mathbf{q}_1\|^2 \|\mathbf{q}_2\|^2}$$

The error is symmetric, i.e., $E_{dif}(\mathbf{q}_1, \mathbf{q}_2) = E_{dif}(\mathbf{q}_2, \mathbf{q}_1)$, but the difference is not.

Appendix B. Influence of the Leaves

Suppose node i is a leaf that is connected to node j and j is connected to only node k and i . The i - and j -th rows of Eqn. 3.4 give the relation of q_i , q_j and q_k ,

$$A \begin{bmatrix} q_i \\ q_j \end{bmatrix} = q_k b$$

where

$$A = \begin{bmatrix} w_{ij} - \lambda(1 + w_{ij}) & -w_{ij} \\ -w_{ij} & (w_{ij} + w_{jk})(1 - \lambda) - \lambda \end{bmatrix}$$

and

$$b = \begin{bmatrix} 0 \\ w_{jk} \end{bmatrix}$$

If A is non-singular, the above equation is solved,

$$\begin{cases} q_i = \eta_i q_k \\ q_j = \eta_j q_k \end{cases}$$

For some combinations of w_{ij} , w_{jk} and λ , η_i and η_j may be very large. Table 3 is an example where we assume $w_{ij} = w_{jk} = e^{-1}$. Consequently, a small change of Δq_k may induce a large Δq_i and Δq_j .

References

- [1] B. Bollobas. *Modern Graph Theory*. Springer, New York, 1998.

- [2] M. Charikar, C. Chekuri, T. Feder, and R. Motwani. Incremental clustering and dynamic information retrieval. *Annual ACM Symposium on Theory of Computing*, 1997.
- [3] F. R. K. Chung. *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92) (Cbms Regional Conference Series in Mathematics)*. American Mathematical Society, February 1997.
- [4] P. Desikan, N. Pathak, J. Srivastava, and V. Kumar. Incremental page rank computation on evolving graphs. *WWW*, 2005.
- [5] C. Ding. A tutorial on spectral clustering. *ICML*, 2004.
- [6] C. Ding, X. He, H. Zha, M. Gu, and H. Simon. A min-max cut algorithm for graph partitioning and data clustering. *Proceedings of ICDM*, 2001.
- [7] G. Golub and C. V. Loan. *Matrix Computations*. John Hopkins Press, 1989.
- [8] S. Guha, N. Mishra, R. Motwani, and L. OCallaghan. Clustering data streams. *the Annual Symposium on Foundations of Computer Science, IEEE*, 2000.
- [9] C. Gupta and R. Grossman. Genic: A single pass generalized incremental algorithm for clustering. *2004 SIAM International Conference on Data Mining*, 2004.
- [10] L. Hagen and A. B. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design*, 11(9):1074–1085, September 1992.
- [11] A. N. Langville and C. D. Meyer. Updating pagerank with iterative aggregation. *WWW*, 2004.
- [12] Y.-R. Lin, H. Sundaram, Y. Chi, J. Tatemura, and B. Tseng. Discovery of blog communities based on mutual awareness. *3rd Annual Workshop on the Weblogging Ecosystem*, 2006.
- [13] L. Lovasz and M. Plummer. *Matching Theory*. Akademiai Kiado, North Holland, Budapest, 1986.
- [14] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *NIPS*, 2002.
- [15] H. RHEINGOLD. The virtual community: Homesteading on the electronic frontier. *The MIT Press*, 2000.
- [16] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2000.
- [17] Y. Wei and C. Cheng. Towards efficient hierarchical designs by ratio cut partitioning. *Proc. International Conference on Computer Aided Design*, pages 298–301, 1989.
- [18] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. *SIGIR*, 2003.
- [19] S. X. Yu and J. Shi. Segmentation given partial grouping constraints. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(2):173–183, 2004.
- [20] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. *NIPS*, 2004.
- [21] X. Zhu. Semi-supervised learning literature survey. (1530), 2005. http://www.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf.