Blog Community Discovery and Evolution Based on Mutual Awareness Expansion

Yu-Ru Lin Hari Sundaram Yun Chi Junichi Tatemura Belle L. Tseng
Arts Media and Engineering Program
Arizona State University
NEC Laboratories America
Cupertino, CA 95014
Email: {yu-ru.lin, hari.sundaram}@asu.edu, {ychi, tatemura, belle}@sv.nec-labs.com

Abstract

There are information needs involving costly decisions that cannot be efficiently satisfied through conventional web search engines. Alternately, community centric search can provide multiple viewpoints to facilitate decision making. We propose to discover and model the temporal dynamics of thematic communities based on mutual awareness, where the awareness arises due to observable blogger actions and the expansion of mutual awareness leads to community formation. Given a query, we construct a directed action graph that is time-dependent, and weighted with respect to the query. We model the process of mutual awareness expansion using a random walk process and extract communities based on the model. We propose an interaction space based representation to quantify community dynamics. Each community is represented as a vector in the interaction space and its evolution is determined by a novel interaction correlation method. We have conducted experiments with a real-world blog dataset and have promising results for detection as well as insightful results for community evolution.

1. Introduction

Navigational queries such as "take me to the CNN website" can be addressed through conventional web search engines. However, there are information needs that cannot be satisfied by simple answer, or there might be no "correct" answer to the query. Such queries are usually referred as *informational* queries. Examples include costly product search, e.g. "Nikon D200", "the best school district in Manhattan" and "Global warming." The first query has monetary implications, the second impacts the family, and the third is important to someone who is passionate about that topic and may want to join a group.

Revealing communities that discuss these topics would be of great value to the user. Furthermore, tracking how the discussion change in communities is important as in each of these example queries, the user is unlikely to make quick decisions. Instead, they may like to follow the discussions over time before they make up their mind. In fact users are actively seeking multiple viewpoints to inform their decision making.

We believe that community centric web search can play an important role in informational queries. The challenge lies in extracting and tracking active communities with a content related theme to provide meaningful information. In this paper we develop a computational approach to discover and model the temporal dynamics of thematic communities in the Blogosphere.

Our Approach. A key idea in this paper is that human communities emerge through observable actions by the community members. According to Dourish [4], members share the common sense understandings through the reciprocal actions in a so called *action community*. For example if Alice leaves a comment on Bob's blog, then Bob is aware of Alice. If he reciprocates by leaving a comment on Alice's blog, then the awareness is mutual. On the other hand, if Alice creates a citation link to a post by Bob, then Bob cannot observe this action. In this case, while Alice is aware of Bob, he is unaware of Alice and the awareness is *not* mutual. For two people to be mutually aware of each other, there needs to be evidence of bi-directional action.

The role of mutual awareness in human community has been discussed in [8]. In this paper we further examine the process where mutual awareness leads to community formation. We are interested in detecting communities that have a strong content related theme. We first construct a directed graph that is time-dependent, and weighted with respect to a specific query. Each node is a blogger, and the edges are the observable actions. We use a "symmetric social distance" which is estimated using a random walk process, to capture mutual awareness expanding among community. We then extract communities by maximizing the distance between two sets of bloggers.

We develop an interaction space based representation to quantify community dynamics, where each dimension represents a pair-wise interaction between two people, and each community is a vector in the interaction space. Two communities are close if they are close in the interaction space. We compute the *interaction correlation* of two communities by using histogram intersection of their vectors. Then, given a community at a certain time slice, the community evolution is determined by maximizing the interaction correlation over communities in the previous time slice. We have conducted experiments with a real-world blog dataset and have promising results for detection as well as insightful results for community evolution.

Related work. There has been prior work on analyzing online social network dynamics. Kumar *et al*. [7] studied the bursty evolution of links among different communities in the blogosphere. Backstrom *et al*. [1] analyzed the dynamic community formation. These studies focus on high-level structural changes while our analysis focuses on the micro structural and thematic changes down to the level of individual communities.

Community extraction has been studied as a graph problem. Shi *et al*. [11] proposed using normalized cut criteria for graph partitioning which resulted in an efficient spectral clustering algorithm. Dhillon *et al*. [3] showed the close connection between spectral clustering and weighted kernel k-means. Kanna *et al*. [6] proposed an iterative spectral clustering algorithm using minimum conductance cut criteria. We propose the symmetric social distance as a clustering criterion, to detect communities based on the key insight of human community formation. Our work is closely related to [13] which proposes a random walk based distance measure and apply kernel k-means clustering.

There has been recent work on community evolution. Palla *et al*. [9] and Falkowski *et al*. [5] seek to capture the behavior of human groups by quantifying the social group evolution. Our analysis differs from theirs in that their method is only based on community membership difference and focuses on structural change, which ignores the evolution of interactions and topical interests among community. Our approach explores community dynamics from structural and thematic aspects based on members' interaction.

The rest of the paper is organized as follows. We give formal problem statement in section 2. In section 3 we present the proposed approach to detect human communities in the blogosphere. In section 4 we discuss a framework to analyze community evolution. We present the experimental results in section 5 and finally conclude in section 6.

2. Problem statement

We first introduce notions used in this paper. We consider each unique blog as a *blogger* who writes blog posts (or entries) to communicate personal interests, thoughts, experience, etc. Some bloggers *interact* with other bloggers by contributing comments in response to specific blog posts. In this paper, we consider one of the most frequently observable actions in the blogosphere – creating links to other blogs' post in their own posts, i.e. entry-entry link.

To capture human interacting behavior with specific topical interests, we construct a time-dependent weighted directed graph $G_Q(t) = (V,E)$ with respect to a specific query Q, where V is a set of n bloggers (as nodes) and E is a subset of all possible actions i.e. $E \subseteq V \times V$ of m direct links (as edges). G_Q represents the degree of interests in a query topic Q for a set of users and their interactions. Assume we have edge weight $w(e)$: $E \rightarrow \Re_+$ for all $e \in E$, where the edge weight $w(e)$ represents the degree of interest in interactions between two bloggers with respect to Q. Let $\mathbf{A}$ be the matrix corresponding to a direct graph $G_Q(t)$, and let $[\mathbf{A}]_{ij} = w(e)$ correspond to the edge weight of the edge $e = (i, j)$ from node i to j.

In order to detect an action community, it is important that the actions are mutually observable by community members. Given bloggers' query-sensitive actions as an action matrix $\mathbf{A}$, we then construct a mutual awareness matrix $\mathbf{W} = \min(\mathbf{A}, \mathbf{A}^T)$, where $\mathbf{A}^T$ is the transpose of $\mathbf{A}$. The mutual awareness matrix $\mathbf{W}$ represents the degree of interests in mutually observable actions between any pair of bloggers for a certain time.

Let C be a set of communities relevant to the user query Q. We assume that each of these communities has a finite temporal duration. At the time t of the query, $C(t) \in C$ communities will exist. Without loss of generality, we assume there are k such communities. The query sensitive community detection and evolution problem is stated as follows:

1. *Community detection*: Given a query Q at time t, a time-dependent graph $G_Q(t)$, find k query-sensitive communities $C(t) = \{C_1, C_2, \ldots, C_k\}$.
2. *Community dynamics*: Given a set of communities $C(t) = \{C_i\}$ for $i = 1...k$, determine for each C_i, $C^*_i(t+1)$. $C^*_i(t+1)$ represents the evolution of $C_i(t)$.

Note that in general the set of communities *detected* at given time $t+1$, $C(t+1)$, will be different from the set of evolved communities $\{C^*_i(t+1)\}$. For example, there might be a new community *born* at time $t+1$.

3. Detection

In this section we propose a community detection method based on the concept of mutual awareness expansion.

3.1 Mutual awareness expansion

A community emerges from direct interactions (i.e., mutual awareness) between individuals since they gain a sense of community through observation of other members' interaction [4,8]. In [8] the authors propose mutual awareness as a key property in community. Here we further discuss how mutual awareness plays a key role in community formation. We highlight three elements in this process: (1) transitivity, (2) reciprocity, and (3) frequency. First, one could become aware of a member without direct interaction since he or she can observe his or her direct peers interacting with other people. Thus awareness is transitive. Second, such transitive awareness must be reciprocal. If expansion of awareness is only one directional, one might not feel belonging to the community. Third, the amount of observed interaction must be sufficient for members to feel connected to each other. We refer to this process as *mutual awareness expansion*.

We propose a model of mutual awareness expansion that uses a random walk process to estimate the probability that two bloggers are aware of each other through mutual awareness expansion.

The process is similar to Travers and Milgram's well-known small world experiment [12]. Given two people A and B, we ask A to forward a letter to B. If A does not know B, she forwards the letter to one of her friend who is more likely to know B. Unlike Milgram's experiment, though, we do this process for both directions: we also ask B to forward the letter *back* to A. The expected path length from A to B and B to A should capture the three elements of mutual awareness expansion, i.e., transitivity, reciprocity and frequency. We refer to this length as "symmetric social distance" in contrast with Milgram's one-way "social distance."

The model is formally stated as follows. Given a direct graph $G = (V,E)$ and the mutual awareness matrix $\mathbf{W}$ associated with G, the random walk on G is defined to be the Markov chain with state space V and the transition matrix $\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$, where $\mathbf{D}$ is a diagonal matrix with element $d_{ii} = \sum_j w_{ij}$. A random walker at a node i on G will follow the transition probability $p_{ij} = [\mathbf{P}]_{ij}$ to visit the next node j. Note that by construction $w_{ii} = \sum_j w_{ij}$ (i.e. $[\mathbf{P}]_{ii} = 1/2$) for $i \neq j$[1].

Let the social distance from node u to v, denoted by $\tau_{u\to v}$, the expected number of steps to reach node v from node u. Let the symmetric social distance $\tau_{u\leftrightarrow v} = \tau_{u\to v} + \tau_{u\leftarrow v}$. In random walk literature, these two quantities are usually referred to as expected *hitting time* ($\tau_{u\to v}$ and $\tau_{u\leftarrow v}$) and expected *commute time* ($\tau_{u\leftrightarrow v}$), respectively. We now briefly summarize prior work [2] in computing commute time in a random walk process.

The social distance from u to v satisfies:

$$\tau_{u\to v} = \begin{cases} 1 + \sum_{(u,x)\in E} p_{ux}\tau_{x\to v} & \text{if } u \neq v \\ 0 & \text{if } u = v \end{cases} \quad <1>$$

where p_{ux} is the transition probability from u to x. To reach v from u, the random walker takes one step to get to the next node x with transition probability p_{ux}, and then calculates the rest expected distance to v.

Let $\mathbf{H}$ be the social distance matrix with element $[\mathbf{H}]_{uv} = \tau_{u\to v}$, and define $\Delta = \mathbf{I} - \mathbf{P}$. We can rewrite eq.<1> as:

$$\Delta\mathbf{H} = \mathbf{J} - vol\mathbf{D}^{-1} \quad <2>$$

where $\mathbf{J}$ denote the all 1's matrix and $vol = \sum_{u,v\in V} w_{uv}$. The right hand side subtract $vol\mathbf{D}^{-1}$ because $[\mathbf{H}]_{uu} = \tau_{u\to u} = 0$, and the expected number of steps for a random walker to return to his starting node is $1/\pi_v$, where $\pi_v = d_{vv}/vol$ is the stationary distribution of visiting node v. Define the Green's function $\mathbf{G}$ so that $\mathbf{G}\Delta = \mathbf{I}$. It has been shown in [2] that the social distance (hitting time) from u to v can be computed as:

$$\tau_{u\to v} = \frac{vol}{d_{vv}}\mathbf{G}(v,v) - \frac{vol}{d_{uu}}\mathbf{G}(u,v) \quad <3>$$

Further define $\mathbf{G}' = \mathbf{G}\mathbf{D}^{-1}$ and $\mathbf{L} = \mathbf{D}\Delta = \mathbf{D} - \mathbf{W}$. It can be checked that both $\mathbf{G}'$ and $\mathbf{L}$ are symmetric. Because $\mathbf{G}\Delta = \mathbf{I} = \mathbf{G}'\mathbf{L}$, $\mathbf{G}'$ can be solved as the pseudo-inverse of $\mathbf{L}$, as:

$$\mathbf{G}' = \mathbf{L}^{+} = \sum_{i=2}^{|V|} \frac{1}{\lambda_i}\phi_i\phi_i^T \approx \sum_{i=2}^{k} \frac{1}{\lambda_i}\phi_i\phi_i^T \quad <4>$$

where $0 = \lambda_1 < \lambda_2 \leq \ldots \leq \lambda_n$ are the eigenvalues, ϕ_i's are the corresponding eigenvectors of $\mathbf{L}$. $\mathbf{G}'$ can be approximately computed by truncating the eigenvalues and eigenvectors after the k^{th} smallest eigen-pair in eq.<4> for $k < n$.

The symmetric social distance $\tau_{u\leftrightarrow v}$ is computed by:

$$\begin{aligned} \tau_{u\leftrightarrow v} &= vol(\mathbf{G}'(v,v) - \mathbf{G}'(u,v) + \mathbf{G}'(u,u) - \mathbf{G}'(v,u)) \\ &\approx vol\sum_{i=2}^{k} \frac{1}{\lambda_i}(\phi_i(u) - \phi_i(v))^2 \end{aligned} \quad <5>$$

[1] By this construction, we create a lazy transition matrix $\mathbf{P} = (\mathbf{I}+\mathbf{P}')/2$ which is aperiodic if $\mathbf{P}'$ is an irreducible transition matrix. $\mathbf{P}'$ is defined by $\mathbf{P}' = \mathbf{D}^{-1}\mathbf{W}$ where $[\mathbf{W}]_{ii} = 0$.

3.2 Detection algorithm

We use the symmetric social distance as a criterion to extract a community. Given a set of bloggers V, a subset S of V can be seen as a community if the symmetric social distance between members of S is short whereas distance between members and non-members is longer. Therefore, we split the set V into two sets S and $V\backslash S$ by maximizing the symmetric social distance. The objective is given as follows:

$$S = \arg\max_{S \subset V} \omega(S,V) \sum_{\substack{u \in S \\ v \in V \backslash S}} \tau_{u \leftrightarrow v}, \qquad <6>$$

$$\omega(S,V) = \left| |S| - \frac{|V|}{2} \right| + \frac{|V|}{2},$$

where $\omega(S,V)$ is a weighted function used to let the objective function favor balanced splits.

Table 1: The detection algorithm

Community Detection Algorithm

Input: $G_Q(t) = (V, E)$, the mutual awareness matrix $\mathbf{W}$ associated with G_Q, and the number of communities K
Output: a set of communities C

1 $C \leftarrow \{V\}$
2 while $|C| < K$
3 $V' \leftarrow$ select the largest set from C
4 Compute $\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}_{V'}$, where $\mathbf{W}_{V'}$ is the matrix from $\mathbf{W}$ with rows and columns indexed by elements in $V' \subseteq V$
5 $\{\phi_2, \ldots, \phi_k\} \leftarrow$ the 2^{nd} to the k^{th} largest eigenvectors of $\mathbf{P}$
6 $S = \arg\max_{S \subset V'} \omega(S,V') \sum_{\substack{u \in S \\ v \in V' \backslash S}} \tau_{u \leftrightarrow v}$
7 $C \leftarrow (C \backslash V') \cup \{S, V' \backslash S\}$
8 end-while

We give our detection algorithm in Table 1. Note that we compute the i^{th} largest eigenvectors of $\mathbf{P}$ since they are equivalent to the i^{th} smallest eigenvectors of $\mathbf{L}$, given that $\mathbf{L} = \mathbf{D}-\mathbf{W}$ and $\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$. The algorithm initially take the entire set V (line 1) and find a set S in V, where the members in S are most far apart from $V\backslash S$ in terms of symmetric social distance (line 6). It then splits S from V (line 7) and put two sets S and $V\backslash S$ in C. From C, it iteratively selects the largest set and repeats the same steps until K communities are found. The step at line 6 can be computed efficiently given the elements of eigenvector sorted in a decreasing order. We split the set V' by cutting two consecutive elements in the eigenvector that have the largest difference.

4. Evolution

In this section we present an interaction space framework to quantify community dynamics. We propose a method to represent a community using the interaction of its members (section 4.1) and identify the community evolution based on the interaction based representation (section 4.2).

4.1 Interaction is central to a community

The state of a community is determined by the actions of individual members. In traditional analysis, a community is considered to be a group of people, and correlation between communities are quantified by relative overlapped of community membership, i.e. ratio of common members [5,9]. While this approach is computationally inexpensive, it has several weaknesses. First, the analysis ignores members' interaction. A human community could change due to interaction within the same group. Second, the analysis implicitly assumes that all group members play an equal role in community evolution, however this is a simplification. In a human community different members can play different roles, an important member's leaving or joining the community can have significant impact.

Consider a toy example in Figure 1, at time t, there are two communities A and B. Community A has core members $\{a, b, c, d\}$ and peripheral members $\{e, f, g, h\}$, and community B has members $\{i, j, k, l\}$. In this case, it is more reasonable to consider community A is closer to community A' than B', as A' retains the core members in A, while B' has the peripheral members of A. However, the ratio of overlapped membership is the same, and thus fails to distinguish the evolutional correlation.

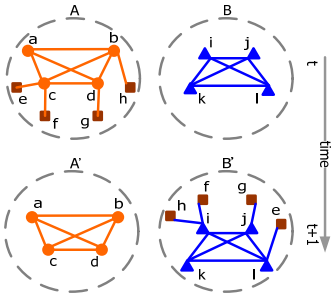


Figure 1: An example of community evolution.

To address these issues, we propose an interaction based representation for each community. The idea is to represent community in the *interaction space*. In this representation, each dimension represents a pair-wise interaction between two people. Since there are N people in the entire set (the universe of people), we will have an $N \times N$ dimensional interaction space.

Members of every community are a subset of this universal set.

Let (i, j) be the index of the dimension that represents how possible user i will interact with user j. The interaction space can be represented using the transition matrix $\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$, where $\mathbf{D}$ is a diagonal matrix with element $d_{ii} = \sum_j w_{ij}$ and $\mathbf{W}$ is the mutual awareness matrix at a time t. $\mathbf{W}$ represents the mutually observable actions occurring before time t. A user i's action at time t is defined as a distribution of his/her interaction probability with other users, as $\mathbf{x}(i) = [\mathbf{P}]_{ij}$ for $j = 1 \ldots N$.

The state of a community is determined by interactions of its members. Thus each community can be defined as a vector in the interaction space. The interaction vector of a community C_k at time t, denoted by $\mathbf{x}(C_k)$, is defined by:

$$\mathbf{x}(C_k; i, j) = \begin{cases} [\mathbf{P}]_{ij}, \text{ if } i \in C_k \\ 0, \text{ otherwise} \end{cases} \quad <7>$$

where $\mathbf{x}(C_k; i, j)$ is the element in the (i, j)-dimension of $\mathbf{x}(C_k)$ and C_k is a community existing at certain time t.

This representation captures richer human activities than membership representation. The membership correlation can be regarded as an operation on the subspace defined by the (i, i)-dimension for $i = 1 \ldots N$ in the entire interaction space.

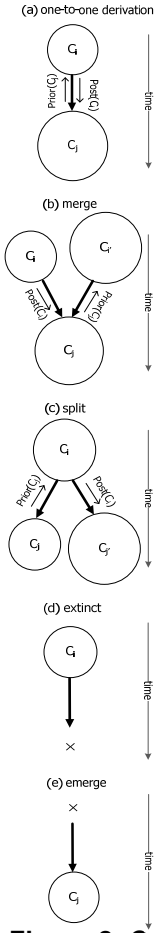
4.2 Interaction evolution

We now present our approach for determining community dynamics through an interaction vector intersection. The key idea is to find the closest community in the $(t+1)^{\text{th}}$ time slice given a community in the current time slice. This approach is based on the assumption that the community dynamics are smooth given the temporal unit (e.g. 1 week) of interaction.

Given a set of communities $C(t) = \{C_k\}$ for $k = 1 \ldots K$, we determine the evolution $C^*_k(t+1)$ for each community C_k by finding the closest community from the set of communities $C(t+1)$. Two communities are close if they are close in the interaction space. We compute the *interaction correlation* of two communities C_i and C_j, denoted by $s(C_i, C_j)$, using histogram intersection of their interaction vectors:

$$s(C_i, C_j) = \frac{\sum_{k=1}^{N^2} \min\left(\mathbf{x}(C_i; k), \mathbf{x}(C_j; k)\right)}{\sum_{k=1}^{N^2} \max\left(\mathbf{x}(C_i; k), \mathbf{x}(C_j; k)\right)} \quad <8>$$

where $\mathbf{x}(C_i; k)$, $\mathbf{x}(C_j; k)$ are the k^{th} dimension of the interaction vectors for community C_i and C_j respectively. This measure compares the all pairs of interaction initiated by members in C_i or C_j. The value of $s(C_i, C_j)$ increases if there are changes in membership or members' interactions. The histogram intersection similarity measure considers only relevant elements in matching vectors and is thus less sensitive to the entire vector length.



(a) One-to-one derivation: If $C_i = Prior(Post(C_i))$ or $C_j = Post(Prior(C_j))$, there is a one-to-one correspondence between C_i and C_j. In this case, we can further distinguish whether C_i grows or shrink simply by comparing the size of C_i and C_j.

(b) Merge: If $C_i \neq Prior(Post(C_i))$, there exists at least one community $C_{i'} = Prior(Post(C_i))$ distinct from C_i that merges with C_i into C_j.

(c) Split: If $C_j \neq Post(Prior(C_j))$, there exists at least one community $C_{j'} = Post(Prior(C_j))$ distinct form C_j so that C_i splits into C_j and $C_{j'}$.

(d) Extinct: If $s(C_i, C_{j'}) < \varepsilon$ for all $C_{j'} \in C(t+1)$, the community C_i is extinct because there exist no communities at time $t+1$ that close to C_i.

(e) Emerge: If $s(C_{i'}, C_j) < \varepsilon$ for all $C_{i'} \in C(t)$, the community C_j emerges at time $t+1$ because there exist no communities at time t that close to C_j.

Figure 2: Community evolution patterns.

The interaction correlation is then used to identify the community evolution. First, we determine the community in the past that is closest to the current community C_i. We refer such community as a *prior* community, denoted by $Prior(C_i)$. $Prior(C_i)$ is computed as argmax $s(C_i, C_j)$ where the maximization is over all communities C_j in the previous time slice. We say C_i *evolves from* $Prior(C_i)$. Similarly, we identify which community in the future is closest to the current community C_i. We refer to such community as a *post* community, denoted by $Post(C_i)$.

The evolutional history, i.e. evolutional graph, of a community can be determined by the post/prior relationship. There are five cases of evolution that can be detected in our analysis: (a) one-to-one derivation, (b) merge, (c) split, (d) extinct and (e) emerge. Figure 2 shows the evolution taxonomy of these five cases. A community evolutional graph is constructed based on the taxonomy.

5. Experimental results

In this section, we present the experimental results based on a real blog dataset.

5.1 Experimental dataset

The real-world blog dataset used in our experiments has been collected by our blog crawler. This dataset contains 274,679 entries and 148,681 entry-entry links crawled from 407 blogs during 63 consecutive weeks, between July 10th in 2005 and September 23rd in 2006. An entry-entry link is considered to be a response by a blogger, towards another blogger's post. We construct a series of time dependent blog graphs sampled at a weekly interval.

To detect query-sensitive communities, we construct the action matrix with respect to a query Q. We compute the relevance score of blog posts to the query Q and extract entry-entry links to estimate the degree of interactions between two bloggers with respect to Q. We picked keywords related to four significant events: "katrina", "london bomb", "ipod nano" and "zotob worm. These keywords serve as query topics to detect the corresponding communities. Because these query keywords are relatively short, we compute the query relevancy by employing a web-based similarity function [10]. We summarize the procedure to estimate the query relevancy in the following.

1. A blog contains a sequence of posts in a chronological order, denoted as $\{p_t\}$. Given a query Q, the query relevancy r_t of a post p_t is defined as the content similarity between Q and p_t, i.e. $r_t = s(Q, p_t)$, where s is a similarity measure.
2. Based on [10], $s(Q, p)$ is computed using cosine similarity between the context vector of the query Q and the content vector of the post p. The context vector of Q is constructed from a set of "snippets" retrieved using a Web search engine [10].
3. The query relevancy of an entry-entry link from blogger u's post $p_{u,t}$ to blogger v's post $p_{v,t'}$, denoted by $r_{uv,t}$, is defined by $r_{uv,t} = \alpha r_{u,t} + (1-\alpha) r_{v,t'}$, where $r_{u,t}$ and $r_{v,t'}$ are the query relevancy of post $p_{u,t}$ and $p_{v,t'}$ respectively. The cited and citing posts can be created at different time and in general $t' \le t$. An entry-entry link is associated with a citing post at time t.
4. The degree of interaction between two bloggers with respect to Q is calculated by aggregating the query relevancy of entry-entry links. Link created later has higher weight. Specifically, we compute $a_{uv,t}$, the query relevancy of a blogger u's actions in response to another blogger v with respect to a query Q, as $a_{uv,t} = \Sigma_t \, r_{uv,t} \, e^{-\lambda(t^*-t)}$, where $r_{uv,t}$ is defined as above, t^* is the query time and $t<t^*$.

The query relevancy of interaction is used to construct the query-sensitive graph G_Q. G_Q is used to determine the query-sensitive communities.

5.2 Evaluation metrics

We use three well-known metrics, conductance, coverage and entropy, for evaluating the performance of different community detection methods. The *conductance* is defined for each cluster as follows:

$$\Phi(C) = \frac{\sum_{u \in C, v \notin C} w_{uv}}{\min\left(\sum_{u \in C, v \in V} w_{uv}, \sum_{u \notin C, v \in V} w_{uv} \right)} \quad <9>$$

where $\Phi(C)$ is the conductance of community C, $w_{uv} = [\mathbf{W}]_{uv}$ and $\mathbf{W}$ is the mutual awareness matrix. A community with a small conductance is relatively cohesive because the amount of interaction to non-members is small with respect to the density of either community members or non-community members.

The *coverage* is defined by:

$$P(C) = \frac{\sum_{u,v \in C} w_{uv}}{\sum_{u,v \in V} w_{uv}} \quad <10>$$

where $P(C)$ is the coverage of community C. Coverage measures the fraction of interactions that are within community. Communities with higher coverage are preferable because a larger value of coverage implies more interaction occurring within communities than between community members and non-members.

Entropy is used to measure the clustering performance with respect to a given class/clustering distribution. For each cluster j, we compute p_{ij}, the "probability" that a member of current cluster j belongs to cluster i at previous time, as follows: $p_{ij} = m_{ij}/m_j$, where m_j is the number of samples in cluster j (at t) and m_{ij} is the number of samples from cluster i (at t-1) into cluster j (at t). The entropy for a cluster j is defined as:

$$H(j) = -\frac{1}{\log L} \sum_{i=1}^{L} p_{ij} \log p_{ij} \quad <11>$$

where L is the number of clusters. The total entropy is computed by taking the weighted average over all clusters.

The first two metrics, conductance and coverage, measure the cohesiveness of communities at a given time (snapshot). The entropy measure compares the clustering distributions from two consecutive snapshots of the blog graph in order to measure the temporal changes of the detected communities.

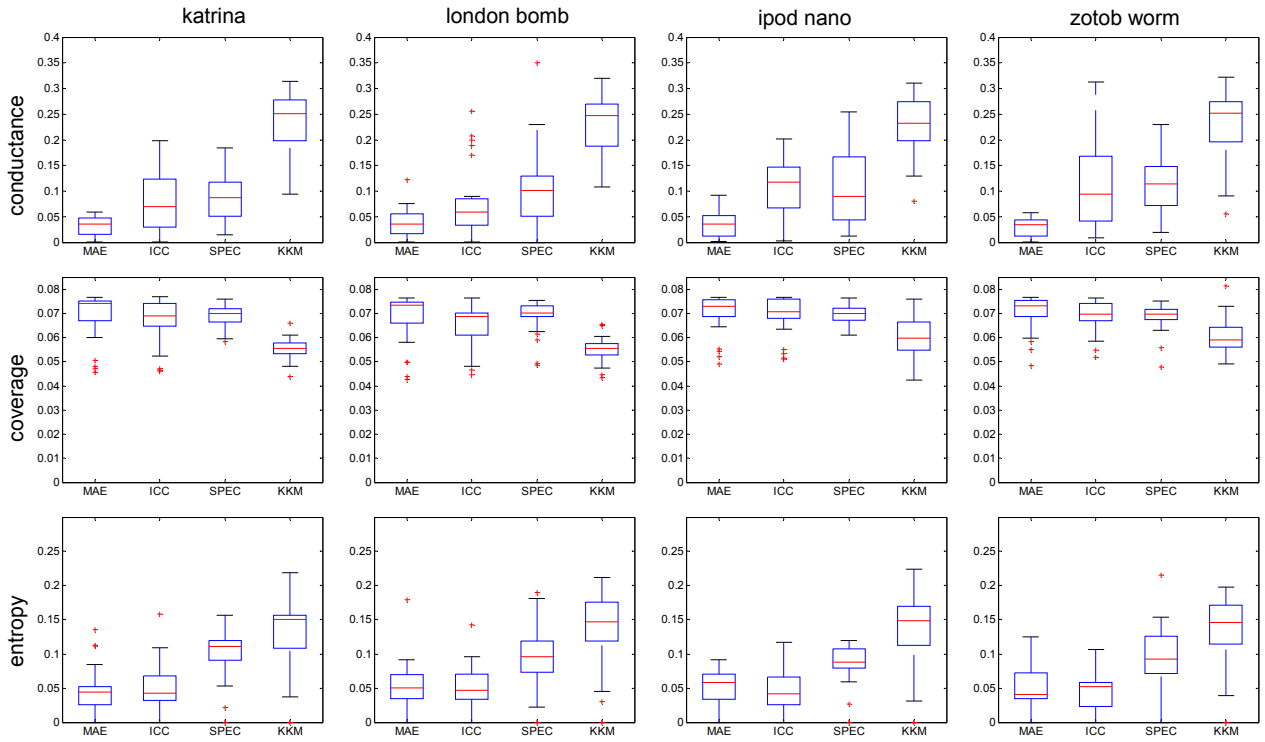


Figure 3: Clustering performance: Our methods (MAE) compared to three baseline algorithms, kernel k-means (KKM), normalized cut (SPEC) and iterative conductance cutting (ICC).

5.3 Clustering performance

We compare our community detection method with three well-known baseline clustering algorithms, including the kernel k-means [3], normalized cut [11] and iterative conductance cutting [6]. We apply our method and the baseline algorithms to detect K communities w.r.t. the four above mentioned queries at each snapshot graph using four different methods. Specifically all four clustering algorithms take the query-sensitive mutual awareness matrices as input and are evaluated on the same matrices. The clustering performance of all snapshots for each method is summarized in Figure 3. We denote our proposed method as "MAE", kernel k-means as "KKM", normalized cut as "SPEC" and the iterative conductance cutting as "ICC". In Figure 3, each column shows the community detection results per query, with query keywords listed on the top. Each row shows the results measured by the three metrics. In each plot we show four boxes with whiskers for the four clustering algorithms.

As can be seen from the figure, our method, MAE, outperforms KKM and SPEC in terms of low conductance, high coverage, and low entropy, while it performs as well as ICC. Also MAE is relatively stable compared with other baseline algorithms, which can be seen from the smaller height of the boxes. The comparison in Figure 3 suggests our proposed method detects good quality communities in terms of clustering performance.

5.4 Community stories

In this section we demonstrate the evolution of the communities with respect to four important events.

Hurricane Katrina. The first query keyword is "katrina", which is about a natural disaster caused by the hurricane Katrina in August 2005. By using the proposed approach, we find relevant communities as shown in Figure 4.

In Figure 4, each node represents a detected community where the communities detected during the same week are aligned horizontally and the communities at different time snapshots are connected by arrows through our evolution analysis. The grayscale of an arrow is proportional to the interaction similarity between the two communities. The node size reflects the number of community members and the reddish shade of node is proportional to the query relevancy for the keyword "katrina". More saturated node represents more relevant community.

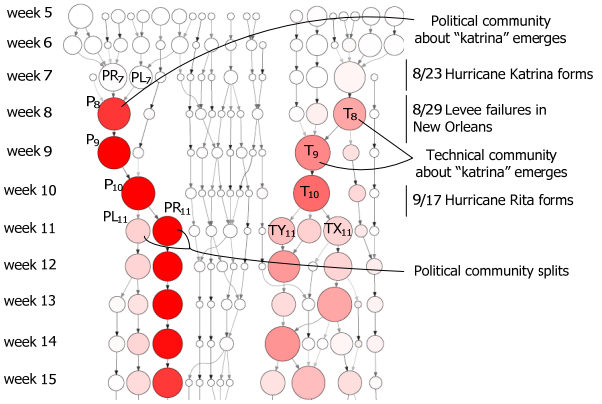


Figure 4: community evolution graph for the query "katrina".

We observed emergence of two types of communities in our dataset, one with a focus on politics (shown on the left) and the other with a focus on technology (shown on the right). When the event Katrina occurred in week 7, we found debates about the government response between members in communities PR_7 and PL_7. The two communities merged into a community P_8 where the discussion focused more about the relief events such as fund-raising. The communities P_9 and P_{10} followed by P_8 have high temporal coherency in terms of interaction similarity. At week 11, P_{10} splits into community PR_{11} and PL_{11} which have members common to PR_7 and PL_7 respectively. The communities following PR_7 still have follow-up discussion about the event, e.g. post-Katrina crime and rumor through media coverage, and a related new event about another hurricane formed at week 10, while communities followed by PL_{11} become less active in posting about Katrina. On the right side, technological communities T_8 and T_9 emerged during week 8 and 9. Discussions in these communities included, e.g. posting how technology can help in post-Katrina relief efforts (at week 9) and a new lunched site for searching missing people from the disaster.

London bombing. We use the keywords "london bomb" to detect communities that related to the London bombing event. This was a series of terrorist attacks on London public transportation system on July 2005. The first attacks occurred on July 7, which is a week ahead of the duration of our dataset. The second attempted attacks occurred in week 2. The detected communities with respect to the query "london bomb" are shown in Figure 5. The most relevant communities are those with political interests. As shown in Figure 5, the PR community initiated related discussion due to the London bombing events, and followed up by posting about terrorist attacks including the September 11, 2001 attack in New York and the Bangladesh bomb blasts on August 17, 2005. Another political community, denoted by PL, was less interested in posting about the London bombing event in the beginning but later during week 8, they joined the discussion with PR community on terrorist related issues, e.g. investigation on suspects, etc. The technical communities related to "london bomb" have two branches. The one denoted by TX is interested in mobile telecommunications and is initially related to the query because of, e.g. some security issue about introducing cellphone service in New York city's underground subway stations. The other denoted by TY was related to the query because of, e.g. some discussions on citizen journalists (meaning that the public journalism is enabled by networking technologies such as blog, mobile, etc.) in the London bombing events.

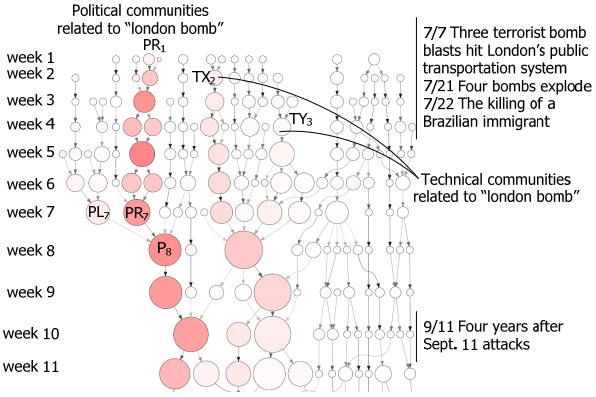


Figure 5: community evolution graph for the query "london bomb".

iPod nano. We use the keywords "ipod nano" to detect communities relevant to the release of a digital audio player "iPod nano". The product was introduced at week 9 and had an overwhelming impact on consumers. As shown in Figure 6, the relevant communities detected from our dataset are rendered on the left. The communities initiated relevant discussion several weeks before the iPod nano was officially introduced and their posts become more and more relevant when closing to the release date. The iPod fans community grew at week 6. The fans continued posting about iPod, e.g. wishlist of iPod, etc. The query relevancy became higher when the time corresponding to other iPod products or news. Unlike the query for "katrina", the communities with political interests seemed to be indifferent with this event.

Computer worm. The last query keyword "zotob worm" is about computer worms that infected computers with Microsoft OS. Several companies such as CNN are affected by the worms during week 5. The detected communities relevant to this query are shown

in Figure 7. In our dataset, this event does not spur as much discussion as above mentioned events. The reason could be that "zotob warm" is more a factual query – bloggers care more about the facts related to the event, such as patch updating or virus removal, than subjective discussion on the event.

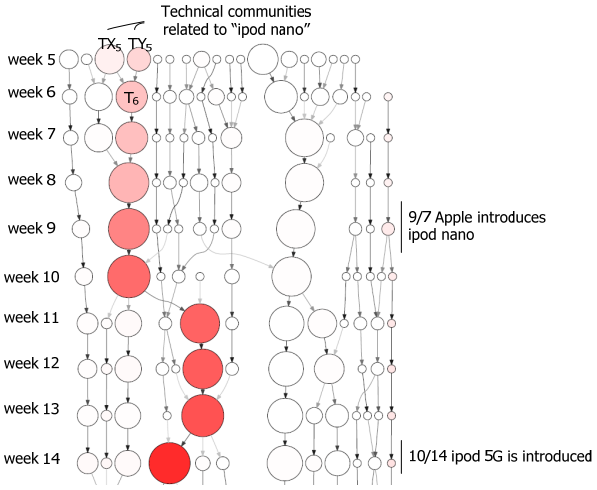


Figure 6: community evolution graph for the query "ipod nano".

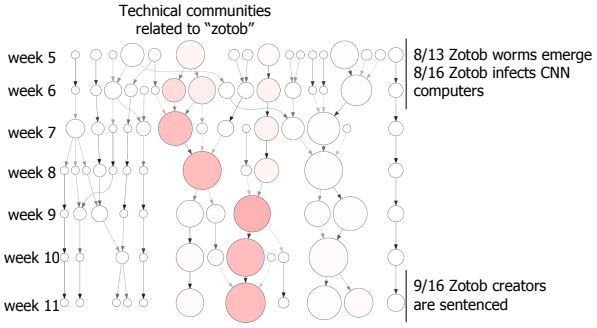


Figure 7: community evolution graph for the query "zotob warm".

The experimental results illustrate our approach to extracting community evolution with real world data. We observed that for queries such as "katrina," "London bomb" and "ipod nano," real world events that generate a diversity of opinion, the community evolution patterns are meaningful.

6. Conclusion and future work

A key idea in this paper was that observable actions lead to the emergence of human communities, and awareness expansion was critical to community formation. We showed how to detect blog communities based on mutual awareness expansion, given a specific query. We proposed a symmetric social distance measure that captures the expansion process and use it to detect communities. We developed an interaction space based representation to quantify community dynamics. Each community was a vector in the interaction space. Community evolution was established by maximizing the interaction correlation between communities across two time slices.

We have excellent experimental results for community detection using standard evaluation metrics. The community evolution with respect to a query reveals interesting community dynamics. As part of future work, we plan to (a) study the community dynamics at different time granularities and (b) develop ranking schemes to characterize the extracted communities.

7. References

[1] L. BACKSTROM et al. (2006). *Group formation in large social networks: membership, growth, and evolution*, Proc. of the 12th ACM SIGKDD, ACM Press, 44-54

[2] F. CHUNG and S. YAU (2000). *Discrete Green's functions.* J. of Combinatorial Theory (A) **91**(1-2): 191-214.

[3] I. DHILLON et al. *A unified view of kernel k-means, spectral clustering and graph partitioning.* Technical report, University of Texas at Austin, 2005.

[4] P. DOURISH (2001). *Where the Action Is:: the Foundations of Embodied Interaction*, Mit Pr.

[5] T. FALKOWSKI et al. (2006). *Mining and Visualizing the Evolution of Subgroups in Social Networks*, International Conference on Web Intelligence, 2006., 52-58

[6] R. KANNAN et al. (2004). *On Clusterings: Good, Bad and Spectral.* J. of the ACM **51**(3): 497-515.

[7] R. KUMAR et al. (2005). *On the Bursty Evolution of Blogspace.* World Wide Web **8**(2): 159-78.

[8] Y. LIN et al. (2006). *Discovery of Blog communities based on Mutual Awareness*, Proc. of the 3rd Annual Workshop on the Weblogging Ecosystems: Aggregation, Analysis and Dynamics,

[9] G. PALLA et al. (2007). *Quantifying social group evolution.* eprint arXiv: 0704.0744.

[10] M. SAHAMI and T. HEILMAN (2006). *A web-based kernel function for measuring the similarity of short text snippets*, Proc. of the 15th international conference on World Wide Web, ACM Press, 377-86

[11] J. SHI and J. MALIK (2000). *Normalized Cuts and Image Segmentation.* IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(8): 888-905.

[12] J. TRAVERS and S. MILGRAM (1969). *An Experimental Study of the Small World Problem.* Sociometry **32**(4): 425-43.

[13] L. YEN et al. (2005). *Clustering using a random walk based distance measure*, Proc. of the 13th Symposium on Artificial Neural Networks (ESANN 2005), 317-24