PET: A Statistical Model for Popular Events Tracking in Social Communities

Cindy Xide Lin *
University of Illinois at Urbana-Champaign
201 N Goodwin Ave., Urbana, IL 61801
xidelin2@uiuc.edu

Bo Zhao
University of Illinois at Urbana-Champaign
201 N Goodwin Ave., Urbana, IL 61801
bozhao3@uiuc.edu

Qiaozhu Mei
University of Michigan
1085 S University Ave., Ann Arbor, MI 4810
bozhao3@uiuc.edu

Jiawei Han
University of Illinois at Urbana-Champaign
201 N Goodwin Ave., Urbana, IL 61801
hanj@cs.uiuc.edu

ABSTRACT

User generated information in online communities has been characterized with the mixture of a text stream and a network structure both changing over time. A good example is a web-blogging community with the daily blog posts and a social network of bloggers.

An important task of analyzing an online community is to observe and track the popular events, or topics that evolve over time in the community. Existing approaches usually focus on either the burstiness of topics or the evolution of networks, but ignoring the interplay between textual topics and network structures.

In this paper, we formally define the problem of popular event tracking (PET) in online communities, focusing on the interplay between texts and networks. We propose a novel statistical method that models the popularity of events over time, taking into consideration the burstiness of user interest, information diffusion on the network structure, and the evolution of textual topics. Specifically, a Gibbs Random Field is defined to model the influence of historical status and the dependency relationships in the graph; thereafter a topic model generates the words in text content of the event, regularized by the Gibbs Random Field. We prove that two classic models in information diffusion and text burstiness are special cases of our model under certain situations. Empirical experiments with two different communities and datasets (*i.e.*, Twitter and DBLP) show that our approach is effective and outperforms existing approaches.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining

General Terms

Algorithms

Keywords

PET, popular events tracking, social communities, topic modeling

1. INTRODUCTION

The prevailing of Web 2.0 techniques has led to the boom of various online communities. Good examples are social communities such as Facebook [1], Blogger [2] and Twitter [3], which successfully facilitate the information creation, sharing, and diffusion among the web users. As a result, a popular topic or event can spread much faster than in the Web 1.0 age. Indeed, when searching for a recent popular event (*e.g.*, "Toyota recall") on Twitter, all the results returned on the first page are created within the past five minutes.

In many scenarios, it is appealing to have a system that tracks the diffusion and evolution of a popular event in a social community. Who are still interested in watching Avatar 50 days after its release date? What do people say about Tiger Woods before and after the scandal? Hot topics emerge, prevail and die. It is desirable to monitor whether people like, what they like, and how their interests change over time.

Tracking the evolution of a popular topic is challenging. The diffusion of an event is vague. You don't know whether I am interest in an event; and even if you do, from whom did I get this interest?

Fortunately, a large volume of text data is generated from the social communities. Besides communicating with friends, a web user also constantly generates text contents such as blogs, tweets, and comments. Both the communications and the contents are changing along time, resulting in a network structure and a text collection which evolve simultaneously and interrelatedly. When we read

*This work was supported in part by NASA grant NNX08AC35A, the U.S. NSF grant IIS-09-05215, an HP Research grant, and by the Army Research Laboratory accomplished under Cooperative Agreement Number W911NF-09-2-0053. The first author was supported by the Microsoft Women's Scholarship. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

KDD'10, July 25–28, 2010, Washington, DC, USA.

[1] http://www.facebook.com
[2] http://www.blogger.com
[3] http://www.twitter.com

what you've written, we can infer your interest in an event; and when we glimpse your communications, we can guess where the interest comes from. When we track the communications and contents over time, we can find out the burstiness, the evolution, and the spread of an event in a social community.

Taking another example, researchers regularly publish papers and also collaborate with other researchers. By analyzing the evolution of publications and collaborations, we can track how a research topic initializes, evolves, and diffuses over the research community, in terms of both content and impact. In all these scenarios, there is an urgent need for a principled method that couples a stream of text and a stream of networks in order to track popular events.

In this paper, we propose a novel and principled probabilistic model (called PET) for tracking popular events in a time-variant social community that consists of both a stream of text information and a stream of network structures. Specifically, PET leverages a Gibbs Random Field to model the interest of users, depending on their historical status as well as the influence from their social connections. A topic model is designed to explain the generation of text data given the interest of a user in an event. The Gibbs Random Field and the topic model thus interplay by regularizing each other. The tasks of tracking popular events are thus cast as an optimization problem aiming at the inference of a joint distribution that consolidates all of historical, textual, and structural features.

We show that PET is motivated by and well reflects the existing observations and findings about information diffusion in social networks and the topic burstiness in text. PET is well connected to two classic models [14, 21], which are proven to be special cases of PET under certain situations. Empirical experiments on two different online communities show that our approach is effective and outperforms various baselines.

The rest of this paper is organized as follows. Section 2 formally defines the problem of PET, as the solution of which a unified probabilistic model is proposed in Section 3. Section 4 discusses the connection of PET to two classic models in literature. We present experiments and results in Section 5, discuss the related work in Section 4, and conclude in Section 7.

2. PROBLEM FORMULATION

In this section, we formally define the related concepts and the task of popular event tracking in social communities. Let us begin with defining a few key concepts as follows.

Definition 2.1. **Network Stream**. Let $G = \{G_1, G_2, \cdots, G_T\}$ be a stream of network structures, where G_k is a snapshot of a general network G at time t_k, ($k \in [1...T]$). And $G_k = \{V_k, E_k\}$, where V_k is a set of vertices and E_k is a set of edges. In a social network, a vertex corresponds to a person. An edge $e = (i, j) \in E_k$ stands for a connection (or a tie) between vertices i and j. We define $g_k(i, j)$ as the strength of the tie (i, j) at time t_k. *W.l.o.g*, we define G_k as a complete graph but allow $g_k(i, j)$ to be any non-negative real value, *i.e.*, $g_k(i, j) = 0$ if there is no tie between vertices i and j. Note G_k can be either undirected or directed.

Definition 2.2. **Document Stream**. Let $D = \{D_1, D_2, ..., D_T\}$ be a stream of document collections, where D_k is the set of documents published between time t_{k-1} and t_k. We further denote $D_k = \{d_{k,1}, d_{k,2}, \cdots, d_{k,N}\}$, where $d_{k,i}$ is the text document(s) associated with the node $v_{k,i}$ in G_k. Document $d_{k,i}$ is represented by a bag of words from a fixed vocabulary $W = \{w_1, w_2, \cdots, w_M\}$. That is, $d_{k,i} = \{c(d_{k,i}, w_1), c(d_{k,i}, w_2), \cdots, c(d_{k,i}, w_M)\}$, where $c(d_{k,i}, w)$ denotes the number of occurrences of word w in $d_{k,i}$.

Definition 2.3. **Topic**. We present a semantically coherent topic θ as a multinomial distribution of words $\{p(w|\theta)\}_{w \in W}$ with the constraint $\sum_{w \in W} p(w|\theta) = 1$. We allow a topic to have different versions over time, denoting the version at time t_k as θ_k ($k \in [1...T]$).

Definition 2.4. **Event**. We define a general event as a stream of topics $\Theta^E = \{\theta_0^E, \theta_1^E, \theta_2^E, \cdots, \theta_T^E\}$. We call θ_0^E the primitive topic of the event, which is independent of the network. θ_0^E can either be specified by the users or be automatically discovered by an event detection algorithm [10]. θ_k^E corresponds to the version of θ_0^E at time t_k. θ_k^E is dependent of the network, which indicates the major aspects of the event in network G_k. Altogether Θ^E represents the origin and evolution of the contents of the event over time. We use $\Theta, \theta_0, \theta_k$ to denote $\Theta^E, \theta_0^E, \theta_k^E$ when there is no ambiguity.

Definition 2.5. **Interest**. For a particular event, at each time point t_k, we assume each node v_i in G_k has a certain level of interest in the event. We model such level of interest as a real value $h_k(i) \in [0, 1]$, and denote the set of interest values for all vertices in G_k as H_k, *i.e.*, $H_k = \{h_k(1), h_k(2), \cdots, h_k(N)\}$. Note that one can also define $h_k(i)$ with a set of discrete levels.

Based on the definitions above, we can define the **event-related information** in a social community as 1) an *observed* stream of network structures; 2) an *observed* stream of text documents; 3) a *latent* stream of topics about the event; and 4) a *latent* stream of interests. We illustrate these concepts with two real world social communities: Twitter and DBLP.

Example 2.1. *For Twitter* [4] *(a micro-blogging network), we extract a collection of N users and all posts published by these users in a range of T days. A time point t_k is defined as the k^{th} day in the time range. $d_{k,i}$ is the document obtained by concatenating all tweets published by user i on day k. The edge weight $g_k(i, j)$ is an estimation of how much user i is influenced by user j on day k, e.g., $g_k(i, j)$ could be defined as the number of i's tweets that follow j in the past 30 days before day k. Here, G_k is a directed graph.*

Example 2.2. *For DBLP* [5] *(a bibliographic network), we retrieve N authors and all publications of these authors in T years. A time point t_k corresponds to the k^{th} year. $d_{k,i}$ is the concatenation of titles of author i's papers in year k. The network G_k is created among these authors according to their co-author relationship. $g_k(i, j)$ is defined as the number of papers co-authored by author i and j in year k, so here G_k is an undirected graph.*

With the definitions of related concepts, we can now formally define the major tasks in the problem of **popular event tracking on networks**. Given the input of network stream G, document stream D and the primitive topic of an event, θ_0, the tasks include:

Task 1: Popularity Tracking. Formally, we want to infer the latent stream of interests, *i.e.*, H_k at each time point t_k during the tracking period. The H_k values can not only indicate the overall popularity trend of the event, but also provide much richer information about how the interest develops, evolves, and spreads on the network.

Task 2: Topic Tracking. Formally, we want to infer the latent stream of topics about the event Θ^E over time. An event starts from its primitive form θ_0^E, and while it is developing, the major aspects of the event may shift substantially over time. By inferring the stream of topics, we expect to keep track of the new development about the event, understand its evolution, and identify the most attentive aspect of the event to the community over time, *etc.*

[4] http://www.twitter.com

[5] http://www.informatik.uni-trier.de/∼ley/db/

Tracking the popular events in a social community is important and challenging in many ways. To track the popularity of events on the network, we should figure out how the interest of each individual is influenced by its social connections, and then develop reasonable models to simulate the formulation and diffusion of the interest on the networks. To track the content evolution of the event, we should make sure the topics we track should be always relevant to the event, and more importantly, reflect the current interest of individuals on the network. This requires us to propose a unified model that takes interest diffusion, network structure and textual contents into consideration at the same time.

It is also worth mentioning that in this work we focus on event tracking, not detection, since the primitive event topic θ_0^E is considered as input to our system. We have observed that in general an event could be well described by a small number of keywords, *e.g.*, "avata", "tiger woods affair", so it is feasible for users to provide the primitive event topic. Indeed, our approach could be combined with any existing event detection algorithms that can automatically discover the bursty keywords or topics either from the same network or other sources, *e.g.*, news articles or the web, then our system will track the events on the focused network.

In next section, we present a novel probabilistic model to achieve the tasks of popular event tracking.

3. EVENT TRACKING MODELS

In this section, we present a novel probabilistic model, **PET**, for tracking popular events in social communities. By considering both the evolution of textual documents and the evolution of network structures, our model can capture the popularity and topic evolution of events in a unified process.

3.1 Intuitions

As discussed in Section 2, a reasonable model of popular events in a social community should not only capture the diffusion of information on the network, but also the burstiness of interests and the generation of contents. What factors should PET consider? What existing observations in social networks and text mining could PET utilize? Before formally introducing the model, we first explain several key observations that motivate the model:

Observation 1. **Interest and Connections.** It has been shown in the study of *social influence* [9] that the behavior of a social actor, *e.g.*, v_i, is usually influenced by its friends [16], especially friends that have stronger ties with v_i [5]. We may expect that the cascade behavior also applies to the interest in an event. On the other hand, the study of *homophily* has shown that people with similar interests are more likely to become connected [1]. Moreover, the $v_i's$ connections have an even stronger influence on the interest of v_i's if v_i's friends have similar interests or v_i's friends with the same interest are strongly connected [3].

Observation 2. **Interest and History.** The behavior of each individual should be generally consistent over time, thus present a strong "personalized" pattern. This also means interest towards certain events should not change dramatically within a short time. When there is a bursting pattern of the interest at time t_k, it's more likely to remain at a high level at time t_{k+1} [14].

Observation 3. **Content and Interest.** When an individual v_i has a higher level of interest in an event, the content she generates should be more likely to be related to the event. On the other hand, when we find v_i writes more about the event, we can assume she is more interested in the event.

We expect these intuitions and observations be helpful in designing the probabilistic model.

3.2 The General Model

Now, at time t_k, we already know the network stream $G_{1...k}$ (short for $\{G_1, G_2, \cdots, G_k\}$) and document stream $D_{1...k}$ (short for $\{D_1, D_2, \cdots, D_k\}$). Let us assume that we've also known the previous interest values $H_{1...(k-1)}$. We want to infer the current interest value H_k and topics Θ_k on the network. We may further make an Markovian simplification that the current interest status only depends on the previous status, *i.e.*, H_{k-1}. So formally, the task is cast as the inference of the posterior of H_k and Θ_k: $P(H_k, \Theta_k | G_k, D_k, H_{k-1})$.

Based on the intuitions and observations, we know H_k depends on the network structure G_k (*i.e.*, Observation 1) as well as the history H_{k-1} (*i.e.*, Observation 2). We also know that the current topic Θ_k and interest status H_k are mutually dependent (*i.e.*, Observation 3) . We can then introduce two reasonable independent assumptions:

(i) Given the current network structure G_k and the previous interest status H_{k-1}, the current interest status H_k is independent of the document collection D_k. The intuition is that people first become interested in the event and therefore generate discussions on it, *i.e.*, D_k should be a result rather than a cause of H_k. Moreover, the interest of an individual is directly determined by her historical status and influential neighbors. Note that the historical documents may still have an impact on H_k, but in an indirect way through H_{k-1}.

(ii) Given the current interest status H_k and the document collection D_k, the current topic model θ_k is independent of the network structure G_k and the previous interest status H_{k-1}. The intuition is that once the author v_i has developed an interest in the event, the contents she writes will only depend on the event itself and the level of the interest.

With the above two assumptions, our object becomes to infer:

$$P(H_k, \Theta_k | G_k, D_k, H_{k-1}) = P(H_k | G_k, H_{k-1}) \cdot P(\Theta_k | H_k, D_k) \quad (1)$$

We denote the first component in Equation 1, $P(H_k|G_k, H_{k-1})$, as the interest model and the second component, $P(\Theta_k|H_k, D_k)$, as the topic model. In the interest model, we propose a multivariate Gibbs Random Field [17] to model the dependency among individuals and the influence of past status (Section 3.3); in the topic model, a mixture model [27] is designed to extract the topic snapshot of the event (Section 3.4). Finally, the inference of the combined model is discussed in Section 3.5.

3.3 The Interest Model

Let us first briefly introduce the Gibbs Random Field [17].

Gibbs Random Field. Given a graph $G = \{V, E\}$, a family of random variables $F = \{F_i\}_{i=1}^N$ is said to be a Gibbs Random Field *w.r.t.* G if and only if its configuration, f, follows a Gibbs distribution that takes the form

$$P(f) = Z^{-1} \times e^{-\frac{1}{\lambda_T} U(f)}$$

where $Z = \sum_{f \in \mathbb{F}} P(f)$ is a normalizing constant called the *partition function*, λ_T is a constant called the *temperature*, and the *energy function* $U(f) = \sum_c V_c(f)$ is a sum of *clique potentials* $V_c(f)$ over all possible cliques c.

In our model, the interest status H_k is a family of random variables defined on graph G_k, and we give a configuration of H_k that follows a Gibbs distribution:

$$P(H_k | G_k, H_{k-1}) = Z^{-1} \times e^{-\frac{1}{\lambda_T} U(H_k)}$$

For the energy function $U(H_k)$, we specifically define two kinds of clique potential functions, while set all other potentials to 0, *i.e.*,

$$U(H_k) = \sum_{i=1}^{N} V_i(h_k(i)) + \sum_{i=1}^{N} V_i'(h_k(i), h_k(-i)) \quad (2)$$

In Equation 2, $-i$ refers to the set of all vertices except i. Note $h_k(i)$ itself is a size-1 clique in G_k, and $\{h_k(i), h_k(-i)\}$ simply equals to G_k, which is also a clique. Hence, Equation 2 is a valid Gibbs Random Field.

We then define $V_i(h_k(i))$ as the transition energy of node i from its last status $h_{k-1}(i)$ to current status $h_k(i)$:

$$V_i(h_k(i)) = (h_k(i) - h_{k-1}(i))^2, \forall i \in [1..N] \quad (3)$$

This definition is mainly motivated by our Observation 2: by minimizing this transition cost we would like the interest values to be generally consistent over time.

The other potential function $V_i'(h_k(i), h_k(-i))$ gives penalty for the difference between the interest of i and its expected value:

$$V_i'(h_k(i), h_k(-i)) = \lambda_{k,i}(h_k(i) - h_k'(i)))^2, \forall i \in [1..N] \quad (4)$$

$h_k'(i)$ is the expectation of $h_k(i)$ estimated from i's neighbors $n(i)$:

$$h_k'(i) = \frac{\sum_{j \in n(i)} g_k(i,j) \cdot h_{k-1}(j)}{\sum_{j \in n(i)} g_k(i,j)} \quad (5)$$

We can see that the design of this cost function is motivated by our Observation 1, which well captures the intuitions in information diffusion: i's current interest is influenced by i's connections, and a stronger tie (*i.e.*, higher $g_k(i,j)$) brings a larger impact.

Moreover, in Equation 4, $\lambda_{k,i}$ is a weight that represents overall how much we trust the "influence from friends", that is,

$$\lambda_{k,i} = \lambda_A \cdot \left(\sum_{j \in n(i)} g_k(i,j) \right) \cdot (1 - \xi(i)), \quad (6)$$

where λ_A is a constant and $\xi(i)$ is the harmonic function [31] defined on the neighbor graph of i:

$$\xi(i) = \frac{\sum_{j_1, j_2 \in n(i), j_1 \neq j_2} g_k(j_1, j_2) \cdot (h_{k-1}(j_1) - h_{k-1}(j_2))^2}{\sum_{j_1, j_2 \in n(i), j_1 \neq j_2} g_k(j_1, j_2)} \quad (7)$$

The definition of $\lambda_{k,i}$ well captures another intuition in our Observation 1: when i's neighbors have a higher agreement on the interest value, the harmonic function becomes smaller, thus results in larger $\lambda_{k,i}$. For special conditions, (i) when $\sum_{j \in n(i)} g_k(i,j) = 0$, we can simply set $h_k'(i)$ to an arbitrary value and set $\lambda_{k,i}$ to zero; and (ii) when $\sum_{j_1, j_2 \in n(i), j_1 \neq j_2} g_k(j_1, j_2) = 0$, we set $\xi(i)$ to 0.5.

To sum up, the posterior of interest status $P(H_k|G_k, H_{k-1})$ is modelled as a Gibbs Random Field on the network G_k. Several potential functions are designed in order to let the interest value of each individual be close to the past status and the "agreement" of the neighbors. The weighting scheme is well motivated by the observations from the real world networks.

3.4 The Topic Model

Now we consider the topic component, $P(\Theta_k|H_k, D_k)$, in Equation 1. In our model, we consider each document $d_{i,k}$ in the collection D_k is generated from a mixture of two multinomial component models. One component model is a background model θ_k^B and the other is the latent event topic model θ_k^E that we want to estimate, *i.e.*, $\Theta_k = \{\theta_k^B, \theta_k^E\}$. The idea is to model the common (non-discriminative) words in D_k with θ_k^B so that the event topic model θ_k^E would attract more discriminative and meaningful words that describe the target event.

The generation process is as follows: to write a word in document $d_{i,k}$, one first choose between the event topic mode (*i.e.*, θ_k^E) and the background model (*i.e.*, θ_k^B), with probability $p(\theta_k^E|d_{k,i})$ and $p(\theta_k^B|d_{k,i})$, respectively. We have $p(\theta_k^E|d_{k,i}) + p(\theta_k^B|d_{k,i}) = 1$. Once the topic is selected, one samples a word from either the event topic model or background model. Different from the traditional mixture language models [12, 27], where the topic distribution of each document is either predefined or solely estimated from the text data, in our model we use the interest value $h_k(i)$, a real value in $[0, 1]$, as the probability of choosing the event topic at node i, *i.e.*, $p(\theta_k^E|d_{k,i}) = h_k(i)$. This is reasonable according to our Observation 3: a higher interest of v_i in the event should result in a higher proportion of the event covered in by v_i. Moreover, as explained in the interest model, $h_k(i)$ could capture the historical interest status and relationships on the network, which implicitly influence the topic model. And modeling the joint distribution with both components would allow the topics and popularity of the events to mutually influence each other over time.

Formally, the probability of generating word w in $d_{k,i}$ is:

$$p(w|d_{k,i}) = h_k(i)p(w|\theta_k^E) + (1 - h_k(i))p(w|\theta_k^B) \quad (8)$$

Then the likelihood of the document collection D_k is given as:

$$P(D_k|H_k, \Theta_k) \propto \prod_{i=1}^{N} \prod_{w \in W} p(w|d_{k,i})^{c(d_{k,i}, w)} \quad (9)$$

where $c(d_{k,i}, w)$ is the number of occurrences of w in $d_{k,i}$.

We further define a conjugate Dirichlet prior of the event topic θ_k^E: $Dir(\{1 + \mu_E p(w|\theta_0^E)\}_{w \in W})$, to incorporate the primitive event topic, which servers as the prior knowledge of the event. By doing this, we regularize the topics so that they do not shift from the event. μ_E is the weight indicating how much we rely on the prior. Formally,

$$P(\Theta_k|H_k) = P(\theta_k^E) \propto \prod_{w \in W} p(w|\theta_k^E)^{\mu_E p(w|\theta_0^E)} \quad (10)$$

We assume $p(w|\theta_k^B)$ does not change over time, which can be simply estimated by the maximum likelihood estimator using the entire document stream.

With the prior defined, the posterior of topics Θ_k is given as:

$$P(\Theta_k|H_k, D_k) \propto P(D_k|H_k, \Theta_k)P(\Theta_k|H_k) \quad (11)$$

3.5 Parameter Estimation

Given our model defined in Equation 1, we can fit the model to the data and estimate the parameters using a Maximum A Posterior estimator. That is:

$$\Lambda^* = \operatorname*{argmax}_{\Lambda} p(C|\Lambda)p(\Lambda) \quad (12)$$

where Λ has the interest values H_k and word distribution in the topic models Θ_k. The hidden variable in our model is $z_{d_{k,i},w}$, indicating which topic (*i.e.*, θ_k^E or θ_k^B) is selected to generate word w in document $d_{k,i}$.

The Expectation Maximization (EM) algorithm [18] can be applied to estimate the parameters efficiently. In the E-step, it computes the expectation of the hidden variables; and in the M-step, it updates parameters Λ to maximize the object function given above.

Specifically, in the E-step we have:

$$p^{(n)}(z_{d_{k,i},w} = \theta_k^E) = \frac{h_k^{(n-1)}(i)p^{(n-1)}(w|\theta_k^E)}{h_k^{(n-1)}(i)p^{(n-1)}(w|\theta_k^E) + (1 - h_k^{(n-1)}(i))p^{(n-1)}(w|\theta_k^B)} \quad (13)$$

In the M-step, given the expectation of the hidden variables, the object function we want to maximize is $E_{\Lambda^{(n-1)}}\{\log p(C|\Lambda)p(\Lambda)\}$, whose concrete form is put in Appendix A.

By integrating a few Lagrange multipliers [18], we can get:

$$p^{(n)}(w|\theta_k^E) = \frac{\sum_{i=1}^{N} c(d_{k,i}, w)p^{(n)}(z_{d_{k,i},w} = \theta_k^E) + \mu_E p(w|\theta_0^E)}{\sum_{w' \in W}\sum_{i=1}^{N} c(d_{k,i}, w')p^{(n)}(z_{d_{k,i},w'} = \theta_k^E) + \mu_E} \quad (14)$$

The inference of $h_k(i)$ boils down to solve:

$$\alpha h_k(i) - \beta - \frac{\gamma}{h_k(i)} - \frac{\delta}{h_k(i) - 1} = 0 \quad (15)$$

where

$$\alpha = \frac{2}{\lambda_T}(1 + \lambda_{k,i}),$$
$$\beta = \frac{2}{\lambda_T}(h_{k-1}(i) + \lambda_{k,i}h'_k(i)),$$
$$\gamma = \sum_{w \in W} c(d_{k,i}, w)p^{(n)}(z_{d_{k,i},w} = \theta_k^E),$$
$$\delta = \sum_{w \in W} c(d_{k,i}, w)p^{(n)}(z_{d_{k,i},w} = \theta_k^B).$$

(i) when $\sum_{w \in W} c(d_{k,i}, w) = 0$, *i.e.*, the document d_i is empty at time point t^k. Then $\gamma = 0$ and $\delta = 0$, so that $h_k(i)$ only depends on the information from the past status and neighbors:

$$h_k(i) = \frac{\beta}{\alpha} = \frac{h_{k-1}(i) + \lambda_{k,i}h'_k(i)}{1 + \lambda_{k,i}} \quad (16)$$

(ii) when $\sum_{w \in W} c(d_{k,i}, w) > 0$, Equation 15 is equivalent to:

$$\alpha h_k(i)^3 - (\alpha + \beta)h_k(i)^2 + (\beta - \gamma - \delta)h_k(i) + \gamma = 0 \quad (17)$$

Any efficient root searching approaches for cubic functions [22] can be applied to find the feasible $h_k(i)$ that satisfies Equation 17. Denote the left of the equation as $f(h_k(i))$. Then $f(-\infty) = -\infty$, $f(+\infty) = +\infty$, $f(0) = \gamma > 0$, $f(1) = -\delta < 0$. It is easy to show there exists exact one root in $(0, 1)$, and therefore the solution for $h_k(i)$ is guaranteed to be found.

4. DISCUSSIONS

We have presented the model and the inference of PET. Although it is a novel probabilistic model, it is well connected to existing models in literature. In this section, we describe two famous existing models of word burstiness and network diffusion, and show that both of them are special cases of PET under certain situations: when the network effect in PET is omitted, it is well connected to the first model (Sec. 4.1); on the other hand, when the topic effect of PET is omitted, it is well connected to the second model (Sec. 4.2). Finally, we analyze the time complexity of PET in Sec. 4.3.

4.1 The State Automation Model

The first is a state automation model proposed by Kleinberg, *et al.* in [14] in the context of detecting bursting activities in an email stream. It is an HMM-like model which assumes the intervals between messages depend on the hidden "bursty" states. We look at a variation of this model which matches our counting data.

Taking a sequence of counting of messages $X = \{x_1, x_2, \cdots, x_T\}$ as the observation, we define a state automation model based on HMM. Instead of the exponential density function in [14], we define the emission probability by a Poisson distribution, since Poisson is much more natural to model word counts [8], *i.e.*,

$$P(x_k|\lambda_k) = \frac{\lambda_k^{x_k} e^{-\lambda_k}}{x_k!},$$

where λ_k is the expected number of messages at time k, which also stands for the hidden state at time t_k. The transition probability from any state to another is defined as a constant related to the number of states. The maximum likelihood estimator gives the λ_k based on:

$$\lambda_k^* = \operatorname*{argmax}_{\lambda_k} \lambda_k^{x_k} e^{-\lambda_k}$$

Now we show this is a special case of PET by setting several constraints and assumptions: (i) we assume all individuals have the same interest level h_k at time k; (ii) we set $\lambda_T = \infty$ in Equation 2, *i.e.*, we ignore the network structures; (iii) we assume there are only two pseudo words in the vocabulary, an *event* word w_1 and a background word w_2, and set $\mu_E = 0$ so that $P(\Theta_k|H_k) = 1$, *i.e.*, the influence of the primitive topic disappears. Thus we have $p(w_1|\theta_k^E) = 1$, and $p(w_2|\theta_k^B) = 1$ for any k. Then our topic model is transformed to a binomial distribution:

$$P(\Theta_k|H_k, D_k) = P(D_k|H_k, \Theta_k)$$
$$\propto h_k^{\sum_{i=1}^{N} c(d_{k,i},w_1)} \times (1 - h_k)^{\sum_{i=1}^{N} c(d_{k,i},w_2)}$$

We know that a Poisson distribution can be described as a limiting case of a binomial distribution. Specifically when total number of words $n = \sum_{w \in W}\sum_{i=1}^{N} c(d_{k,i}, w)$ is sufficiently large, the Poisson distribution in the State Automation model is approximately equivalent to the binomial distribution in our topic model, and we have:

$$\lambda_k \approx n \cdot h_k$$

This well connects PET with the state automation model. The detailed deduction is omitted due to space limitation.

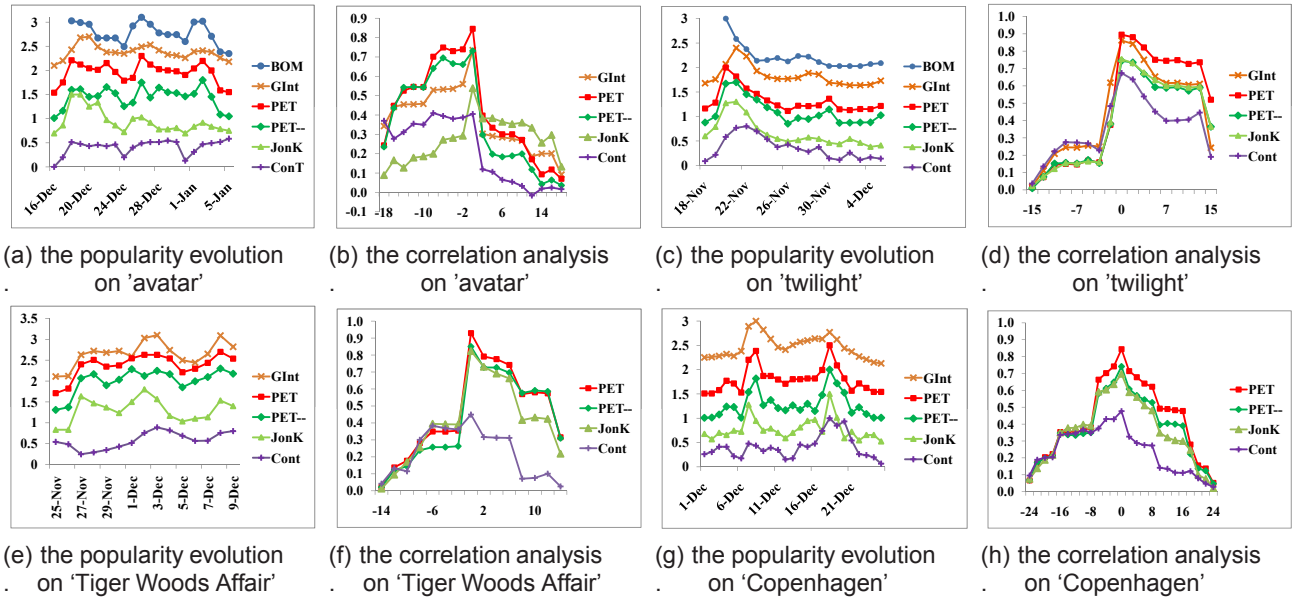
4.2 The Contagion Model

Let us look at another classic model in the context of information diffusion, *i.e.*, the contagion model introduced in [21]. The general idea is that a person becomes infected (corresponding to the case that a person is interested in an event) if the number of its infected friends in the last time point is above a threshold. Let us simplify PET as follows: (i) $c(d_{k,i}, w) = 0$ for any node i and word w, *i.e.*, the influence of the text information is ignored; (ii) $g_k(i, j) = 1$ if v_i is influenced by v_j and otherwise 0; and (iii) $\lambda_A = \infty$ (so that $\lambda_{k,i} = \infty$), *i.e.*, the influence of neighbors becomes dominative, and the history of i is ignored.

According to Equation 16, we have

$$h_k(i) = \operatorname*{arg\,max}_{\lambda_{k,i} \to \infty} \frac{h_{k-1}(i) + \lambda_{k,i}h'_k(i)}{1 + \lambda_{k,i}} = h'_k(i),$$

where $h'_k(i)$ equals to the ratio of infected friends of i at the last time point. If we set $h_k(i)$ to 1 only when $h'_k(i)$ is larger than



(a) the popularity evolution on 'avatar' (b) the correlation analysis on 'avatar' (c) the popularity evolution on 'twilight' (d) the correlation analysis on 'twilight'

(e) the popularity evolution on 'Tiger Woods Affair' (f) the correlation analysis on 'Tiger Woods Affair' (g) the popularity evolution on 'Copenhagen' (h) the correlation analysis on 'Copenhagen'

Figure 1: The Popularity Trend Analysis: PET generates the most consistent trends to the gold standard.

a threshold, otherwise $h_k(i)$ remains 0, this is equivalent to the contagion model.

4.3 Complexity Analysis

Probabilistic Latent Semantic Analysis (PLSA) [12] is a well-known statistical topic model, which and whose variance algorithms are being widely used in practice. Let us analyze the time complexity of PET by comparing it with PLSA.

For a collection $\mathcal{C}$ of N documents that involves t topics and a fixed vocabulary W consisted of M words, the log likelihood to be generated with PLSA is given as follows:

$$L(\mathcal{C}) = \sum_{d \in \mathcal{C}} \sum_{w \in W} c(w,d) \log \sum_{j=1}^{t} p(\theta_j|d)p(w|\theta_j)$$

Estimating the parameters in the above log-likelihood by the Expectation Maximization (EM) algorithm [18] involves the computation for Nt hidden variables and $(N+M)t$ parameters for each EM-iteration. If we expect such EM procedure, in average, terminates after m iterations, it is easy to conclude that the time complexity of PLSA is $O((N+M)mt)$.

Similarly, carrying out a Maximum A Posterior estimator (Sec. 3.5), PET needs $O(NM)$ times computations for both of each E-step and M-step, if the cubic function (Equ. 15) is considered to be solved in constant time [22]. Based on the same assumption that the EM algorithm ends up after m rounds, PET has $O(NMmT)$ run-time complexity for T time points as a whole. Empirically, a popular event in social communities is only able to attract considerable public attention for a short period (*i.e.*, a small value of T), *e.g.*, the discussion of a movie event on Twitter usually becomes trivial after the 90^{th} days of the movie release. Hence, comparing to PLSA, it shows that the time complexity of PET is reasonable and thus affordable in practice.

5. EXPERIMENTS

We have introduced PET, a novel statistical model for Popular Event Tracking in social communities, and discussed its connections with two classic models [14, 21]. In this section, we show the effectiveness of our model with experiments on two different genres of data, *Twitter* and *DBLP*.

5.1 Popular Events Analysis on Twitter

5.1.1 Data Collection

Twitter[6] is a free social networking and microblogging service that enables its users to send and read messages known as *tweets*. Tweets are text-based posts of up to 140 characters displayed on the author's profile page and delivered to the author's *followers*. In this experiment, we create our testing data set (Twitter) by selecting 5,000 users with follower-followee relationships and crawling down 1,438,826 tweets displayed by these users during the period from Oct. 2009 to early Jan. 2010. We consider each day as a time point: for each time point t_k, (i) the document $d_{k,i}$ is obtained by concatenating tweets displayed by user i in day k; and (ii) $g_k(i,j)$ equals to the number of tweets displayed by user i by following user j during the period from t_{k-30} to t_k.

Some simple statistics are presented as follows: (i) for each day, there are only average 37% users who display tweets; (ii) there are 12% days when less than 20% users display tweets; (iii) there are 58% tweets which have at least one followee; (iv) each user has average 10.2 followees. These statistics confirm our hypothesis stated in this paper: *the information of an individual user sometimes is sparse, but individuals are strongly connected by networks*.

5.1.2 Baseline and Gold Standard

JonK. The first baseline is the state automation model stated at Section 4.1, which is a variation of the Kleinberg's model [14]. Concretely, the observation x_k is the total frequency of event-related words in tweets posted by all users, and the hidden state λ_k is selected from a limited set of discrete interest levels $\{\frac{i}{10000}n\}_{i=1}^{100}$. We believe this is a good representative of event tracking methods that do not consider the network effect.

Cont. The second baseline is the contagion model [21] introduced in Section 4.2. Concretely, two users are neighbors in the

[6]www.twitter.com

contagion network at time t_k if they have the follower-followee relationship in the past 30 days. A user becomes newly infected if the number of infected users among her friends in last day is more than a pre-defined threshold. This is a representative of network-based diffusion models that do not consider textual documents.

PET–. To evaluate the effort from network structures in our model, we implement a special version of PET by removing network structures, *i.e.*, we keep every part in the PET model the same. but set $g_k(i, j) = 0$.

BOM. For a movie-related event, the box office earning is a trustworthy criterion to reflect the movie's popularity. Hence, we extract the daily box office at Mojo [7] to be the gold standard for movie-related events.

GInt. For a news-related event, the popularity can be obtained through analysis on the query log of search engines, such as Google [8]. Therefore, we use the interest index supplied by Google Insight [9] as the gold standard for news-related events. Moreover, GInt is a baseline for movie related events.

5.1.3 *Analysis on Popularity Trend*

Experiment Setup. The model PET involves three parameters λ_T, λ_A and μ_E. λ_T and λ_A in the interest model determine the weights for historical and structural information, and μ_E in the topic model is the weight of Dirichlet prior. In our implementation, we set up the parameters empirically as $\lambda_T = 1$, $\lambda_A = 3$ and $\mu_E = 1$. Furthermore, the primitive topic θ_0^E is given as the input for each event E, and H_0 is simply set to all zeros.

Empirical Evaluation. On the testing dataset Twitter, we track the interest levels of events by using PET, PET–, JonK, Cont, BOM (if available) and GInt, respectively. Four popular events are selected for analysis: two movie related events, *i.e.*, 'Avatar' and 'the Twilight Saga: New Moon', and two news related events, *i.e.*,'Tiger Woods Affair' and 'Copenhagen Climate Conference'. We selected these four events because their life cycle well overlaps with the time period of our Twitter data. Furthermore, we select about 30% users and average their interest levels as the overall popularity index, which is curved in Figure 1(a), 1(c), 1(e) and 1(g) for the four events, respectively. To make clearer comparisons, the curves in the same figure are normalized to the same scale [0,1] and are shifted vertically with a certain distance. These modifications do not harm to our experiments, since the trend of each curve is completely reserved. By visual comparisons, in all figures, the curve PET is more similar to the one of the gold standard.

Quantitative Evaluation. We leverage cross-correlation score to quantitatively measure the consistence of the trends to the gold standard. The cross-correlation score is a measure of similarity of two time series as a function of a time-lag between time series [6]. The cross-covariance function between two time series $\mathbf{x} = \{x_i\}_{i=1}^n$ and $\mathbf{y} = \{y_i\}_{i=1}^n$ associated with an event E is defined as

$$c_{\mathbf{xy}}(k) = \frac{1}{n}\sum_{i=1}^{n-k}(x_i - \mu(\mathbf{x}))(y_i - \mu(\mathbf{y})) \;\; k = 0 \cdots n-1$$

$$c_{\mathbf{xy}}(k) = \frac{1}{n}\sum_{i=1-k}^{n}(x_i - \mu(\mathbf{x}))(y_i - \mu(\mathbf{y})) \;\; k = -1 \cdots 1-n,$$

where $\mu(\cdot)$ is the mean. The cross-correlation is the cross-covariance scaled by the variances of the two series:

$$r_{\mathbf{xy}}^E(k) = \frac{c_{\mathbf{xy}}(k)}{\sqrt{c_{\mathbf{xx}}(0) \cdot c_{\mathbf{yy}}(0)}}$$

Figure 1(b), 1(d), 1(f) and 1(h) draw the cross-correlation curves between each method and the gold standard for the four events, respectively [10]. Furthermore, Figure 2 reports the aggregated performance $\mathbf{r}_{xy}$ on a set of twenty events $\{E_i\}_{i=1}^{20}$ by defining $r_{xy}(k) = \frac{1}{20}\sum_{i=1}^{20} r_{xy}^{E_i}(k)$.

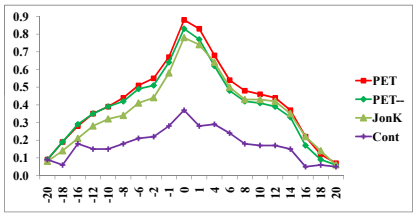


Figure 2: The Aggregated Performance: PET is most consistent with the golden standard

Result Analysis. We can observe several facts:

1. PET always has the best performance (*i.e.*, the highest cross-correlation score), because it estimates the popularity by comprehensively considering historic, textual and structured information in a unified way.

2. Cont always has the worst performance among all comparable methods, since it aims to answer the question in a different scenario: when can a local behavior spread to the whole network? As a contagion model, the behavior of one user can infect another on the network via a long chain and by taking a long transfer time when local interaction is sufficient, so the popularity estimated by Cont at a certain time point could be the mixture of current user behaviors and the ones happened long time ago. However, such 'long chain' rule does not apply to popular events in online social communities. For example, the popularity index of Cont at *Dec 24* in Figure 1(a) is unfavourably higher than the gold standard, because Cont mistakenly transferred some popularity from *Dec 20*. Also, Cont shows a smoother 'valley' at *Dec 5* in Figure 1(e) than the gold standard, because the steep downward slope is neutralized by the 'peak' at *Nov 30*.

3. JonK generally performs well, but is still less accurate than PET at most time points. There are at least two underlying reasons. First, JonK is not able to detect coherent terms that are not given as event-related terms, so JonK may underestimate the popularity due to missed coherent terms. For instance, the popularity index of JonK at *Dec 28* in Figure 1(a) is much lower than the gold standard because people at that time talked about 'avatar' more on 'James Cameron', 'film technology', 'box office', *etc*, rather than directly using the key words 'avatar'. Also, such underestimation happened at *Dec 9* in Figure 1(g), since the event-related terms 'Copenhagen' and 'climate' are insufficient to describe more details of the conference such as 'China', 'global' and 'warming'. Second, similar to many other methods, JonK takes a sequence of aggregated counting data (*e.g.*, the total frequency of terms) as its observation. However, such aggregated

[7] http://boxofficemojo.com/movies
[8] http://www.google.com
[9] http://www.google.com/insights/search

[10] We assume that the popularity of a movie at day k may be reflected on Twitter at day $(k + 1)$.

Dec 14		Dec 18		Dec 26	
trailer	0.21	avatar	0.30	avatar	0.13
avatar	0.10	imax	0.06	imax	0.04
cameron	0.04	trailer	0.05	trailer	0.04
james	0.02	cameron	0.04	technology	0.03
sam	0.01	james	0.04	sam	0.02
director	0.01	alien	0.01	film	0.02
titanic	0.01	titanic	0.01	james	0.02

Table 1: The Content Evolution of 'avatar'

Nov 18		Nov 20		Dec 22	
moon	0.06	moon	0.17	moon	0.11
twilight	0.04	twilight	0.10	twilight	0.04
trailer	0.03	oprah	0.04	fantasy	0.02
chris	0.02	trailer	0.03	chris	0.02
stewart	0.01	vampire	0.03	saga	0.01
premiere	0.01	fantasy	0.02	women	0.01
taylar	0.01	midnight	0.01	milion	0.01

Table 2: The Content Evolution of 'twilight'

Nov 25		Nov 27		Dec 10	
rhapsody	0.07	tiger	0.11	tiger	0.10
muppets	0.06	woods	0.09	woods	0.06
bohemian	0.06	injure	0.04	brown	0.02
lambert	0.01	car	0.03	mistress	0.01
tiger	0.01	accident	0.03	golf	0.01
woods	0.01	championship	0.01	shame	0.01
playlist	0.01	hospital	0.01	divorce	0.01

Table 3: The Content Evolution of 'tiger woods'

Dec 07		Dec 15		Dec 18	
climate	0.02	oral	0.02	climate	0.04
copenhagen	0.01	council	0.02	copenhagen	0.03
conference	0.01	climate	0.01	conference	0.02
china	0.01	trade	0.01	reach	0.01
committee	0.01	copenhagen	0.01	summit	0.01
global	0.01	health	0.01	failure	0.01
warming	0.01	bill	0.01	agreement	0.01

Table 4: The Content Evolution of 'Copenhagen'

data could not precisely stand for the popularity over the network. For example, ten users claiming the same conclusion is definitely more trustworthy than one user repeating the conclusion by ten times. However, JonK treats the two situations indistinguishably.

4. To demonstrate the different performances with and without network structures, we select about 30% users, among which there are strong connections. Compared to PET, PET– shows two weaknesses due to the lack of network structures. On one hand, PET– can not reponse sufficiently to sudden changes. When there is no textual information about a particular user on current time point, PET– will set the new status of the user the same as the previous one, but PET can evaluate the new status more precisely by borrowing information from the user's neighbors. For example, the box office drops a lot at *Nov 21* in Figure 1(c). However, PET– did not recognize such changes until *Nov 23*. On the other hand, PET– is more fragile to reflect local noises. For example, one tweet followed by ten users will be more influential than the same tweet with no follower. However, PET– treats the two situations equivalently, so that the influence of 'isolated' tweets is unfavorably enlarged, *e.g.*, there is an abnormal vibration at *Dec 27* in Figure 1(a).

5.1.4 Analysis on Network Diffusion

In this experiment, we study how events diffuse over networks. There is a burstiness from Dec 16 to Dec 18 in Figure 1(a) since Dec 18 is the release date of 'avatar' in North America. Figure 3 draws the networks on selected 100 users for PET, PET– and Cont, where the color of a vertex represents the interest level of the corresponding user, and an edge stands for the follower-followee relationship between its two ending vertices. View I, II and III correspond to Dec 16, 17 and 18, respectively. For better visual comparisons, View III uses a smaller scale of colors than View I and II, so as to avoid paleness of View I and II.

We can catch several observations: (i) Cont can not tell the difference between interest levels of infected users; (ii) both PET and PET– are able to catch the rising trend of popularity, but vertices in PET are smoothed via edges in the network, which accords better with the situation in real world: *people's interests are inevitably influenced by their friends*.

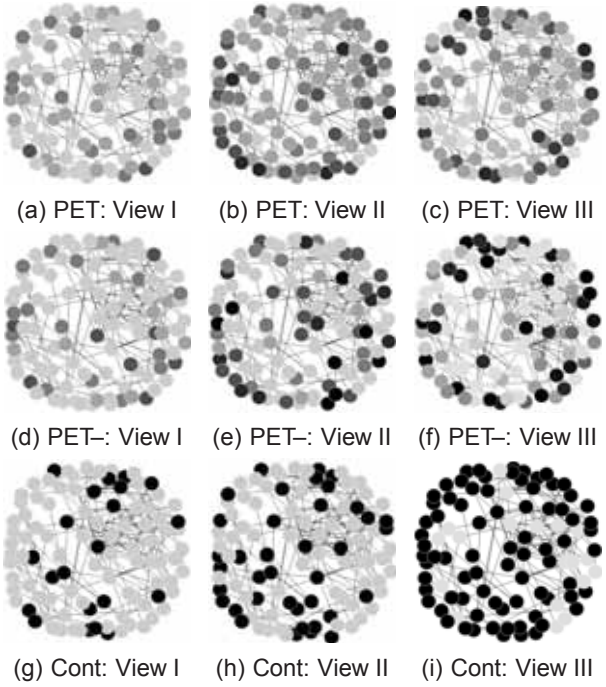


Figure 3: The Network Diffusion Analysis: PET generates the smoothest diffusion.

5.1.5 Analysis on Content Evolution

Table 1-4 shows the topics extracted by PET that evolve along time. These results are interesting and reasonable. For example, in Table 1, users began to talk about 'avatar' by introducing the movie's title, the actor 'Sam Worthington' and the director 'James Cameron' who was also the director of the movie 'Titanic'; in the release day of Dec 18, new terms appeared such as 'aliens' and 'imax', and the term rank of 'trailer' dropped; when this movie became more and more famous, people extended their discussions to the movie's historic significance, *i.e.*, its 3D film technology. Also, Table 3 shows the evolution of gossip on the golf star 'Tiger Woods': before Nov 27, the information about Tiger Woods was

limited and inaccuracy; in Nov 27, the car accident was reported and people worried about his injury condition and golf competitions; after his affair was brought to light, people used words related to the 'scandal' such as 'brown' and 'mistress', blamed his sexual abuse, and felt curious about the possible 'divorce'. Again, by observing Table 4, we can easily find that people kept great attentions on the 'Copenhagen Climate Conference' when it was opened at Dec 07, but thought it was a 'failure' when the conference was closed at Dec 18. In Table 2, the contents do not change much for the movie 'twilight', which reflects the limit aspects of discussions that could be an evidence to explain why its box office earnings kept dropping after its release date.

5.2 Popular Events Analysis on DBLP

Data Collection. The Digital Bibliography and Library Project (DBLP) is a database which contains the basic bibliographic information of computer science publications [11]. In this experiment, we create our testing data set DBLP by selecting 12,949 authors who published at least 10 papers in conferences of data mining and database, and crawling down 500,417 papers published by these authors during the period from 1990 to 2008. Concretely, we consider one year as a time point, and titles and author lists are extracted to form the documents and the co-author networks.

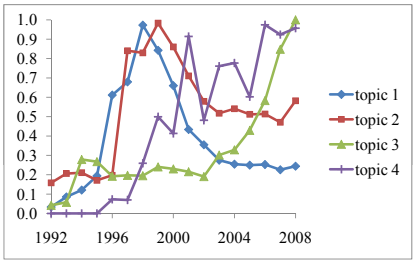


Figure 4: The Popularity Evolution of DBLP Topics.

Result Analysis. We select four research topics (i) 'frequent', 'itemset', 'mining', 'association', 'rule', (ii) 'data', 'cube', 'OLAP', 'aggregation', 'materialization', (iii) 'web', 'mining', 'social', 'network', 'community', and (iv) 'topic', 'modeling', 'PLSA', 'LDA', 'latent'. We then track the popularity evolutions on the four topics. As shown in Figure 4, (i) topic 1 was popular in last decade but was fading out recently; (ii) topic 2 has a burstiness when Jim Gray first introduced 'data cube' in 1996; (iii) topic 3 monotonically increases; and (iv) topic 4 has two rise-ups when PLSA and LDA was introduced in 1999 and 2002, respective.

To sum up, by comprehensively considering historic, textual and structured information into a unified model, PET generates more accurate trends, smoother diffusion, and meaningful content evolution for popular events in social communities.

6. RELATED WORK

PET is a novel model for tracking popular events in social communities. It provides a unified probabilistic model that considers 1) the burstiness of user interest; 2) the evolution of network; 3) the network effect in information diffusion; and 4) the evolution of textual topics. To the best of our knowledge, there is no existing model that considers all these four factors in a unified way. There are, however, several lines of related work.

[11] http://www.informatik.uni-trier.de/∼ley/db

There have been extensive studies on detecting and tracking events, e.g., [14, 15, 7, 24, 26, 13, 32, 2, 29], which facilitate a wide range of tasks such as search [15], clustering [23], classification [10], *etc.*

Event Detection. This line of work has a different goal from our paper: they detect events and our paper aims to track events by observing their popularity and content evolutions. Our event tracking models can be integrated with any existing detection algorithms that can automatically discover the primitive event topic from different sources. These methods typically treat the text collection alone and do not consider the network effect in a social community.

Event Tracking. A state automation model was proposed by Kleinberg, *et al.* [14] to detect bursty activities from an email arrival stream, by assuming the rates of messages are determined by underlying hidden states. [13] models the sequence of counting data by combining two Poisson distributions - one for the normal periodic count data and the other for the rare events. [21] evaluates network diffusion models by considering the question that when a local behavior can spread to the whole population. These methods take either sequences of statistical data (*e.g.*, word frequencies) or interaction systems as the input, but do not simultaneously consider network structures and textual topics in the data stream, which are shown by our study as quite effective in tracking popular events in social communities.

Topic Modeling. Topic modeling approaches [12] [4] have been developed to mine variations of topics in different contexts [28], evolution of topics [20], and correlated patterns in multiple text streams [26]. These methods generally do not consider the network structures. Recently, incorporating network regularization in topic modeling has been proposed [25] [19]. [19] uses a harmonic function to enforce the constraint that topic distribution on neighboring nodes should be similar, and [25] defines a Markov Random Field on the graph to model the influence between nodes in a generative way. However, these methods typically do not consider the burstiness of interests and the evolution of network structures. Thus they can not be directly applied in our problem in order to track popular events. The Gibbs Random Field in our model gives much more flexibility to incorporate various factors so that we are able to model the diffusion of interest and evolution of topics together.

Information Diffusion. Information diffusion is a classic topic in social network analysis, which models the cascade of behaviors on a network structure (e.g., [21, 16, 3, 11]). Some of the potential functions in the Gibbs Random Filed in PET are motivated from the findings in the literature of information diffusion. This line of work, however, do not consider textual topics, and usually do not consider the evolution of ties. It is thus hard to be applied to track popular events.

The most relevant work may be [30], which models the social interactions and topic evolutions in an academic network. However, they treat the social interactions and the evolution of topics in separate procedures, and do not consider the change of social interactions over time.

7. CONCLUSION

In this work, we propose the novel problem of popular events tracking in a social community. Given a stream of network structures, an associated stream of text documents, and the primitive form of events, we could track the popularity of the events on the network and content revolution of the events over time. We make several key observations about how the interest, topics and network structures mutually influence each other, and propose a novel statistical model that can handle all the constraints. The proposed model, PET, not only provides a unified probabilistic framework to

model different factors in modeling the evolution of interests and contents, but also covers classical models as special cases. Comprehensive experimental studies on two real-world datasets show that our approach outperforms existing ones, and two of them are actually special cases of our model in certain circumstances. Our approach can potentially enable many more informative analysis of certain topics on specific networks, and interesting real-world applications.

One interesting future direction is to apply our model to detect and track the evolution of ideas, gossips, and scientific innovations. Another interesting future work is to consider the mixture of multiple events in PET. One may envision a real-time event search system which finds and summarizes events in social communities.

8. REFERENCES

[1] L. A. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, 2003.

[2] L. Araujo, J. A. Cuesta, and J. J. M. Guervós. Genetic algorithm for burst detection and activity tracking in event streams. In *PPSN*, pages 302–311, 2006.

[3] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: membership, growth, and evolution. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 44–54, 2006.

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

[5] J. J. Brown and P. H. Reingen. Social ties and word-of-mouth referral behavior. *Journal of Consumer Research: An Interdisciplinary Quarterly*, 14(3):350–62, 1987.

[6] C. Chatfield. The analysis of time series. In *Chapman and Hall*, 1984.

[7] L. Chen and A. Roy. Event detection from flickr data through wavelet-based spatial analysis. In *CIKM*, pages 523–532, 2009.

[8] K. W. Church and W. A. Gale. Poisson mixtures. *Natural Language Engineering*, 1:163–190, 1995.

[9] T. L. Fond and J. Neville. Randomization tests for distinguishing social influence and homophily effects. In *WWW*, pages 601–610, 2010.

[10] G. P. C. Fung, J. X. Yu, P. S. Yu, and H. Lu. Parameter free bursty events detection in text streams. In *VLDB*, pages 181–192, 2005.

[11] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 491–501, 2004.

[12] T. Hofmann. Probabilistic latent smantic analysis. In *UAI*, 1999.

[13] A. T. Ihler, J. Hutchins, and P. Smyth. Adaptive event detection with time-varying poisson processes. In *KDD*, pages 207–216, 2006.

[14] J. M. Kleinberg. Bursty and hierarchical structure in streams. In *KDD*, pages 91–101, 2002.

[15] T. Lappas, B. Arai, M. Platakis, D. Kotsakos, and D. Gunopulos. On burstiness-aware search for document sequences. In *KDD*, pages 477–486, 2009.

[16] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. In *EC '06: Proceedings of the 7th ACM conference on Electronic commerce*, pages 228–237, 2006.

[17] S. Z. Li. Markov random field modeling in image analysis. In *Springer-Verlag New York, Inc.*, 2001.

[18] G. McLachlan and T. Krishnan. The em algorithm and extensions. Wiley series in probability and statistics, Hoboken, NJ, 2008. Wiley.

[19] Q. Mei, D. Cai, D. Zhang, and C. Zhai. Topic modeling with network regularization. In *WWW*, pages 101–110, 2008.

[20] Q. Mei and C. Zhai. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In *KDD*, pages 198–207, 2005.

[21] S. Morris. Contagion. In *Review of Economic Studies*, pages 57–78, 2000.

[22] R. Nickalls. A new approach to solving the cubic: Cardan's solution revealed. In *The Mathematical Gazette*, page 354Ű359, 1993.

[23] N. Parikh and N. Sundaresan. Scalable and near real-time burst detection from ecommerce queries. In *KDD*, pages 972–980, 2008.

[24] D. Preston, P. Protopapas, and C. E. Brodley. Event discovery in time series. In *SDM*, pages 61–72, 2009.

[25] Y. Sun, J. Han, J. Gao, and Y. Yu. itopicmodel: Information network-integrated topic modeling. In *ICDM*, pages 493–502, 2009.

[26] X. Wang, C. Zhai, X. Hu, and R. Sproat. Mining correlated bursty topic patterns from coordinated text streams. In *KDD*, pages 784–793, 2007.

[27] C. Zhai and J. D. Lafferty. Model-based feedback in the kl-divergence retrieval model. In *CIKM*, pages 403–410, 2001.

[28] C. Zhai, A. Velivelli, and B. Yu. A cross-collection mixture model for comparative text mining. In *KDD*, pages 743–748, 2004.

[29] Q. Zhao, P. Mitra, and B. Chen. Temporal and information flow based event detection from social text streams. In *AAAI*, pages 1501–1506, 2007.

[30] D. Zhou, X. Ji, H. Zha, and C. L. Giles. Topic evolution and social interactions: how authors effect research. In *CIKM*, pages 248–257, 2006.

[31] X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, pages 912–919, 2003.

[32] Y. Zhu and D. Shasha. Efficient elastic burst detection in data streams. In *KDD*, pages 336–345, 2003.

APPENDIX

A. MODEL INFERENCE

$$
\begin{aligned}
&E_{\Lambda^{(n-1)}}\{\log p(C|\Lambda)p(\Lambda)\} \propto \\
&\quad -\log Z - \frac{1}{\lambda_T}\sum_{i=1}^{N}\left((h_k(i)-h_{k-1}(i))^2+\lambda_{k,i}(h_k(i)-h'_k(i))^2\right) \\
&\quad +\sum_{i=1}^{N}\sum_{w\in W} c(d_{k,i},w)p^{(n)}(z_{d_{k,i},w}=\theta_k^E)\log(h_k(i)p(w|\theta_k^E)) \\
&\quad +\sum_{i=1}^{N}\sum_{w\in W} c(d_{k,i},w)p^{(n)}(z_{d_{k,i},w}=\theta_k^B)\log((1-h_k(i))p(w|\theta_k^B)) \\
&\quad +\sum_{w\in W}\mu_E p(w|\theta_0^E)\log(p(w|\theta_k^E))
\end{aligned}
$$