Extracting Social Events for Learning Better Information Diffusion Models

Shuyang Lin[†] Fengjiao Wang[†] Qingbo Hu[†] Philip S. Yu[†*]
[†]Department of Computer Science
University of Illinois at Chicago
Illinois, USA
[*]Computer Science Department
King Abdulaziz University
Jeddah, Saudi Arabia
{slin38,fwang27,qhu5,psyu}@uic.edu

ABSTRACT

Learning of the information diffusion model is a fundamental problem in the study of information diffusion in social networks. Existing approaches learn the diffusion models from events in social networks. However, events in social networks may have different underlying reasons. Some of them may be caused by the social influence inside the network, while others may reflect external trends in the "real world". Most existing work on the learning of diffusion models does not distinguish the events caused by the social influence from those caused by external trends.

In this paper, we extract social events from data streams in social networks, and then use the extracted social events to improve the learning of information diffusion models. We propose a LADP (Latent Action Diffusion Path) model to incorporate the information diffusion model with the model of external trends, and then design an EM-based algorithm to infer the diffusion probabilities, the external trends and the sources of events efficiently.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Application—*Data Mining*

General Terms

Algorithms, Experimentation

Keywords

social event; information diffusion; social influence

1. INTRODUCTION

Recently, online social networks have become a major medium for the spread of information. News, rumors, and opinions propagate in social networks. These events are usually explained by information diffusion processes driven by social influences between users of a social network. Yet each social network is not a closed world. Users obtain information not only from the social network itself, but also from other sources, such as mass media, lectures in universities, friends in real life, etc.

Most existing work on the information diffusion processes assumes that the model has been learned somehow, and focuses on exploring the properties of the learned models. A less studied, but very important topic is the learning of information diffusion models. Work on this topic usually learns the information diffusion model from events in the social network. However, it is questionable to use all the events without any distinction, since the information diffusion processes are not the only reason triggering events in social networks [2, 18, 1, 15]. Some of the events are results of social influence or information diffusion processes inside the network, while others may reflect external trends in the world outside the network. For example, *#DidYouKnow* was a trending hashtag in the Twitter network in the last three weeks of 2011. The hashtag was used in tweets where people talked about surprising facts. It became popular because of social influence among users in the Twitter network, while *#JapanEarthquake*, another trending hashtag in the Twitter network at the same time, reflects a major event in the outside world. Most previous approaches [9, 17, 16, 8] on the learning of information diffusion models do not distinguish the two different types of events, which makes the learned models inaccurate.

In this paper, we study a problem of learning information diffusion models. We propose a new approach that can distinguish the two different sources of events, and then use the identified social events to improve the learning of information diffusion models. Although the basic idea is straightforward, it is not easy to design a solution based on this idea. There are three key challenges:

- While the sources of some events are easy to be classified as external trends or the social influence, for most events the sources are not easy to determine. For example, when the earthquake hit Japan, a great many of Twitter users prayed for people in Japan. Some of the users did that after they saw the sad news of earthquake on TV, while others did that because they saw other users in the Twitter network do that. In this case, we cannot simply classify this event to be an externally sourced event or a socially sourced event, but need to decide the source with finer granularity. As we will show in Section 2, we define the influence source on the action level.

- In order to distinguish the socially sourced actions from

KDD'13, August 11–14, 2013, Chicago, Illinois, USA.

externally sourced actions, we need the information diffusion model as well as the model of external trends. But both of them are unknown. The model of external trends can only be inferred from the externally sourced actions, while the diffusion model can only be learned from the socially sourced actions. In other words, only when we are able to decide the sources of actions, we can learn the external trend model and the information diffusion model accurately. This leads to an inherent "chicken and egg" problem. We refer to it as "**inference dependency**".

• We need to consider both external trends and information diffusion processes at the same time. It is not trivial, since the external trends are time-related, while the diffusion model depends on the structure of the social network. It requires us to integrate a temporal model and a structural model into one joint model. Besides, both of them contain plenty of parameters. It may lead to high complexity in the inference of the joint model.

In this paper, we propose a novel LADP (Latent Action Diffusion Path) model to extract social events and learn diffusion models with better accuracy. Rather than classify events into external events and social events, we determine for each action in an event whether it is caused by external trends or the social influence inside the network. We use a mixture model framework to combine the external trend model and the information diffusion model together, and decide the class of each action. As the learning algorithm of the model involves the inference of external trends, diffusion probabilities, and the sources of actions at the same time, a naive implementation can lead to prohibitively high computational cost. An inference algorithm based on the expectation maximization (EM) is devised to overcome the difficulty of inference dependency, while avoiding the high computational overhead on repeated invocation of the diffusion model.

The improved accuracy can result in better performance on many applications based on information diffusion models, such as influence maximization [10] and outbreak detection [14]. As we will show in the experiment, for the DBLP network, the top authors suggested by the LADP have an average H-index and number of citations up to 20% higher than the top authors suggested by the state-of-the-art approaches.

Organization. The rest of this paper is organized as follows: Section 2 formally defines the problem. We proposed the LADP model in Section 3, and present the learning algorithm in Section 4. In Section 5, we present experiments. We discuss related work in Section 6, and conclude in Section 7.

2. PROBLEM FORMULATION

In this section, we formally define the task of information diffusion model learning. We begin with a few key concepts as follows. The notations are summarized in Table 1.

DEFINITION 1. **Social Network** *A social network is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where a vertex $v \in \mathcal{V}$ corresponds to a user, and an edge $e = (v_i, v_j) \in \mathcal{E}$ stands for a connection between the users v_i and v_j. Edges in a social network can either be directed or undirected.*

The social network itself provides nothing more than structural information. To learn the diffusion model, we also need the contents created by users in the network, for example, the tweets created by users of Twitter network, or the publications of authors in the DBLP network. We define the collection of contents as "data stream".

DEFINITION 2. **Data Stream** *A data stream $\mathcal{S}$ on a social network $\mathcal{G}$ is defined as a chronological sequence of document sets $\mathcal{C}_t$, i.e. $\mathcal{S} = \{\mathcal{C}_t\}_{t=1}^{T}$. A textual document $d \in \mathcal{C}_t$ contains a set of terms. Each document is associated with a node in $\mathcal{V}$, denoted by v_d, and has a time stamp, denoted by t_d. The t-th document set $\mathcal{C}_t \in S$ contains the documents created at time step t, i.e. $\mathcal{C}_t = \{d; t_d = t\}$.*

If a document d is contained in one of the sets $\mathcal{C}_t$, we say that the document d is contained in the data stream $\mathcal{S}$. With a little abuse of notation, we denote it by $d \in \mathcal{S}$.

We denote with $\mathcal{L}$ the set of terms in the data stream $\mathcal{S}$. Terms in the documents can be defined in various ways. For example, we can define each word in a document as a term. We can also define each hashtag in a tweet as a term. More generally, we can define any tags or labels as terms, so that the streams are not limited to sequences of textual documents. In this paper, we focus on the analysis of textual streams. Nevertheless, the proposed LADP model can be applied to more general types of streams.

In the LADP model, we regard the generation of a document as a process that, for each term $l \in \mathcal{L}$, the author makes a decision whether to include it in the document or not. We call this decision an "action".

DEFINITION 3. **Action** *For each document $d_i \in \mathcal{S}$, for each term $l \in \mathcal{L}$, there is an action (i, l) taken by the author of the document. If the document d_i contains the term l, the action is a **positive action**, denoted by $x_{i,l} = 1$. Otherwise, the action is a **negative action**, denoted by $x_{i,l} = 0$.*

For the positive action, we also introduce the concepts of **socially sourced action**, and **externally sourced action**. A socially sourced action is a positive action that taken by a user because she is influenced by an information diffusion process inside the social network, while an externally sourced action is a positive action that is triggered by an external trend. It is true that a positive action may sometimes be triggered by both the social influence and the external trend at the same time. But in most cases, the major source of an action can be identified, since at one point of time the user usually gets information from only one source. In this paper, for simplicity of the model, we assume that a positive action can either be a socially sourced action or an externally sourced action.

DEFINITION 4. **Event** *An event in a social network is a sequence of positive actions of the same term $l \in \mathcal{L}$. Each event may include a **socially sourced portion** (or a **social event**), and an **externally sourced portion**. The socially sourced portion contains socially sourced actions, while the externally sourced portion contains externally sourced actions.*

We define the sources on the action level, rather than on the event level, on the user level, or on the document level. Although the event, the document and the user of an action are all important factors of it, each factor alone cannot perfectly capture the reasons for triggering the action. As we have discussed in Section 1, we cannot simply

SYMBOL	DESCRIPTION
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	The social network
$\mathcal{S}$	The data stream, define by a temporal sequence of $\mathcal{C}_t$
$\mathcal{C}_t$	The t-th document collection in $\mathcal{S}$
$\mathcal{L}$	The set of considered terms
N	The number of documents in $\mathcal{S}$
N_t	The number of documents in $\mathcal{C}_t$
M	The number of terms in $\mathcal{L}$
T	The number of time steps in the sequence $\mathcal{S}$
$x_{i,l}$	The label denoting whether the action with document d_i and the term l is positive or negative
$z_{i,l}$	The label denoting whether the action with document d_i and the term l is decided by the information diffusion process or not
θ_l	The mean of $z_{i,l}$ for the term l
α	The Beta prior of θ_l
$\mu_{l,t}$	The probability of an action generated from external trends about term l taken at time t being positive
β_l	The Beta prior for $\mu_{l,t}$
$q_{l,t,v}$	The probability of an action generated about term l taken by user v at time t generated from the diffusion model being positive
$p_{u,v}$	The diffusion probability along the edge (u, v)
Λ	The collection of parameters for the model, i.e. $\{\theta_l, \mu_{l,t}, p_{u,v}\}_{l=1,t=1,v,u\in\mathcal{V}}^{M,T}$

Table 1: Notations

classify an event as a socially sourced event or an externally sourced event, since actions in the event may have different sources. We cannot define the classes on user level either, because each user usually obtains information from both inside and outside the network, and actions taken by a user can be triggered by the social influence or external trends. Even the classification on document level is not good enough, since each document may contain several different terms or topics. For example, a tweet in 2011 said "#DidYouKnow that #JapanEarthquake affected the underground water in Florida?" It involves both the social event of using the hashtag "#DidYouKnow" in Twitter community and the external trend of "Japan earthquake".

By defining the classes on the action level, our approach has greatest flexibility and can infer the underlying reasons precisely. By classifying the actions as socially sourced or externally sourced actions, the inference algorithm can split socially sourced portion and the externally sourced portion of an event. In another sense, it can extract the socially sourced portion from the event. We refer to the extracted externally sourced portion as **social event** and the extracting procedure as **social event extraction**.

DEFINITION 5. **Information Diffusion Process** *The information diffusion process is the process that actions of terms propagate along the edges of the social network. The process is the result of influence among users in a social network.*

A diffusion model aims to predict diffusion processes. Typically, given the actions at time step t, the diffusion model predicts the probability for each user in the social network of taking a positive or a negative action in the next time step $t+1$. The IC (Independent Cascade) model [10] is a widely used information diffusion model. In the IC model, when a user becomes active, she has an independent chance to make each of her neighbors become active. In the proposed LADP model, we define a mechanism of positive action propagation that extends the IC model to the action level.

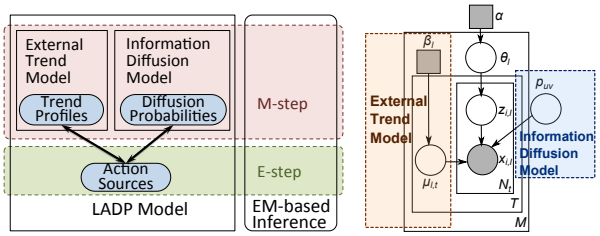
Task. Based on the definitions of the above concepts, we can formalize the task of information diffusion model learning: Given a social network $\mathcal{G}$ and data stream $\mathcal{S}$ on it, we aim to learn the diffusion model on the network $\mathcal{G}$. In varieties of information diffusion models, including the IC model and our model, the parameters of a diffusion model are the diffusion probabilities along edges, so we focus on the learning of diffusion probabilities in this paper. Different from existing approaches, we extract social events from the data stream, and learn the diffusion model from the extracted social events.

3. PROPOSED MODEL

The LADP model extracts social events from data stream, and learns the information diffusion model from the extracted social events. The block diagram of the LADP model is shown in Figure 1(a).

To extract social events, the LADP model infers the influence source for each positive action. The distribution of socially sourced actions is decided by the information diffusion model, and the diffusion probabilities are the parameters that need to be estimated. The distribution of the externally sourced actions is decided by the external trend model, and the trend profiles are the parameters of the information diffusion model that need to be estimated. A mixture framework is proposed to integrate the information diffusion model and the external trend model.

The inference of action sources depends on trend profiles and diffusion probabilities, while the inference of trend profiles and diffusion probabilities depends on the sources of actions. We design an EM-based inference algorithm to solve this "inference dependency" problem. The algorithm estimates the parameters iteratively. In each iteration, it first infers the sources of actions based on the current estimates of trend profiles and the information diffusion probabilities, and then infers the trend profiles and the diffusion probabilities based on the probability of each action being socially caused or externally caused.



(a) Block Diagram (b) Graphical representation

Figure 1: LADP Model

3.1 The framework of LADP Model

Now we formally define the LADP model. As shown in graphical representation of the LADP model in Figure 1(b), the observation variables $x_{i,l}$ are central to the model. Each $x_{i,l}$ indicates whether the corresponding action (i, l) is positive or negative. It is drawn from a mixture distribution, which integrates the external trend and the information diffusion process. The label $z_{i,l}$ decides from which component distribution the variable $x_{i,l}$ is drawn. The left part of the graphical representation (β_l and $\mu_{l,t}$) shows the model of external trends, while the right part ($p_{u,v}$) represents the information diffusion model. The two of them form the two components of the distribution of $x_{i,l}$.

Notice that, the external trend model is a temporal model, while the information diffusion model based on the structure

of the network. Both of them are complicated models with a lot of parameters. It is not easy to combine these two models together without making the inference algorithm intractable. We therefore make an assumption that each action can only be generated from either the external trend model or the information diffusion model. This assumption makes it possible to integrate the two models together under a mixture model framework.

Formally, $x_{i,l}$ is defined as:

$$x_{i,l} = \begin{cases} 1 & \text{if the action } (i,l) \text{ is positive} \\ 0 & \text{otherwise.} \end{cases}$$

The hidden variables $\mathcal{Z}$ indicates whether the action is drawn from the external trend model or from the information diffusion model.

$$z_{i,l} = \begin{cases} 1 & \text{if action } (i,l) \text{ is drawn from the information diffusion component} \\ 0 & \text{otherwise.} \end{cases}$$

Notice that $z_{i,l}$ is defined for all actions, whether they are positive or not, though the concepts of socially sourced action and externally sourced action are defined for the positive actions only. When $z_{i,l} = 1$ and $x_{i,l} = 1$, the action is a socially sourced action. When $z_{i,l} = 0$ and $x_{i,l} = 1$, the actions is an externally source action.

For each i, $z_{i,l}$ is a random variable drawn from Bernoulli distribution with mean θ_l, i.e. $z_{i,l} \sim Bernoulli(\theta_l)$. The mean θ_l is different for each term, since different term has different potential probability of being related to an information diffusion process or an external trend. We use a Beta prior for the parameter θ_l, i.e. $\theta_l \sim Beta(\alpha)$, where $\alpha = (\alpha_1, \alpha_0)$ are fixed parameters. We choose Beta distribution because (1) it has a great flexibility of the shape, and (2) it is the conjugate prior distribution for Bernoulli distribution.

Given the corresponding hidden variable, the distribution of $x_{i,l}$ is given as:

$$x_{i,l} \sim \begin{cases} Bernoulli(q_{l,t_{d_i},v_{d_i}}) & \text{if } z_{i,l} = 1 \\ Bernoulli(\mu_{l,t_{d_i}}) & \text{if } z_{i,l} = 0. \end{cases}$$

where $q_{l,t_{d_i},v_{d_i}}$ is decided by the information diffusion model, and $\mu_{l,t_{d_i}}$ is decided by the external trend model. We will discuss the two models in the following sections.

3.2 Information Diffusion Model

We use a diffusion model that can be regarded as an extension to the widely used Independent Cascade (IC) model [10]. In the IC model, information propagates along the edges in the network. When a node becomes active, it attempts to activate its neighbors. For each node, the attempts to activate it from all its active neighbors are independent. Similarly, in the LADP model, $q_{l,t,v}$, the probability that a user v uses a term l is predicted from the actions that v's neighbors took in the last time step. For each positive action about term l taken by in-neighbors of v at last time step $t-1$, there is an independent chance to make the action of v at time t to be positive. Formally, for the term l, the probability of an action taken by user v at time t being positive is given as:

$$q_{l,t,v} = 1 - \prod_{d_i \in \mathcal{C}_{t-1}, z_{i,l}=1} (1 - p_{v_{d_i},v}) \quad (1)$$

where $\mathcal{C}_{t-1}$ is the set of documents that were created at the last time step $t-1$. $p_{v_{d_i},v}$ is the diffusion probability along with the edge (v_{d_i}, v). For the convenience of notation, we define $p_{v_d,v} = 0$, if there is no edge between the nodes v_d and v. The product in the formula is the probability that all of the in-neighbors of v fail to make the action of v at time t to be positive. Under the independent assumption, this probability could be calculated by multiplying together the probabilities that each attempt fails. We then can get $q_{l,t,v}$, the probability that at least one attempt succeeds, by subtracting the product from 1.

3.3 External Trend Model

For the actions that are generated by external trend model, the probabilities of being positive are not decided by the social network structure and previous actions in the network. A reasonable assumption with these actions is that the probability of being positive only depends on the time and the term, but does not depend on the user takes the action. (A similar assumption was made in [15].) The reason underlying the assumption is that, if an action is decided by an external trend outside the network, we cannot make any prediction on whether the action is positive or not, based on the network structure, so the best assumption we can make is that each action is a random variable independently drawn from the same distribution.

Nevertheless, the probability of an action being positive should be depends on the term l and the time t. That is because different terms have different levels of popularity in the external world. The more popular a term is, the more likely that an action with regard to it is positive. For a given term, its popularity changes over time. For each term l, the parameters $\mu_{l,t}$ forms a time sequence which we call profile of the external trend.

We therefore assume an Beta prior for $\mu_{l,t}$ the probability of being positive: $\mu_{l,t} \sim Beta(\beta_l) = Beta(\beta_{l,1}, \beta_{l,0})$. Similar to the prior of θ_l, we choose Beta distribution because its shape is flexible and it is the conjugate prior for Bernoulli distribution. For each term l, we set the parameter β_l in the prior distribution according to the number of positive and negative actions over all the time steps. Specifically, $\beta_{l,1}$ is set to the average number of positive actions for the term l in all the steps, while $\beta_{l,0}$ is set to the average number of negative actions in all the steps.

4. PARAMETERS ESTIMATION

We have discussed the difficulty of inference dependency in the introduction. We design an EM-based algorithm to solve this difficulty by iteratively estimating the conditional distribution of hidden variables $z_{i,l}$ and the parameters. In the E-step of each iteration, the estimate for the conditional probability of $z_{i,l}$ is updated, while the estimates for parameters θ_l, $\mu_{l,t}$ and $p_{u,v}$ are updated in the M-step.

The parameters θ_l and $\mu_{l,t}$ are easy to estimate using the EM algorithm of the maximum a posteriori (MAP) estimate. However, it is difficult to define and calculate the MAP estimate of the parameters $p_{u,v}$, due to the complexity of the diffusion model. To solve that difficulty, we modify the PCB model in [9] so that it can be integrated into the EM framework to provide estimate of $p_{u,v}$ efficiently. We first assume that $p_{u,v}$ are known (consequently, $q_{l,t,v}$ are known), and show the EM algorithm. Then, we discuss the estimation of $p_{u,v}$ and add it into to the EM framework.

Given the LADP model above, we want to maximize expectation of the logarithm of the posterior:

$$
\begin{aligned}
&Q(\Lambda|\Lambda^{(n)}) = E_{\mathcal{Z}|\mathcal{X},\Lambda^{(n)}} log Pr(\Lambda; \mathcal{X}, \mathcal{Z}) \\
&= \sum_{l=1}^{M}[(\alpha_1 - 1)log\theta_l + (\alpha_0 - 1)log(1-\theta_l)] \\
&+ \sum_{l=1}^{M}\sum_{t=1}^{T}[(\beta_{l,1}-1)log\mu_{l,t} + (\beta_{l,0}-1)log(1-\mu_{l,t})] \\
&+ \sum_{l=1}^{M}\sum_{i=1}^{N}\Big[z_{i,l,1}^{(n)}\big(log\theta_l + x_{i,l}log(q_{l,t_{d_i},v_{d_i}}) \\
&+ (1-x_{i,l})log\left(1-q_{l,t_{d_i},v_{d_i}}\right)\big) \\
&+ z_{i,l,0}^{(n)}\big(log(1-\theta_l) + x_{i,l}log(\mu_{l,t_{d_i}}) + (1-x_{i,l})log\left(1-\mu_{l,t_{d_i}}\right)\big)\Big]
\end{aligned}
$$

where $z_{i,l,j}^{(n)}$ stands for $P(z_{i,l}^{(n)} = j|\mathcal{X}, \Lambda^{(n)})$ for the simplicity of the equation.

The first term in the formula comes from the distribution $\theta_l \sim Beta(\alpha)$. The second term comes from the distribution $\mu_{l,t} \sim Beta(\beta_l)$. The last term comes from the distribution of $\mathcal{X}$ and $\mathcal{Z}$, given the parameters θ, μ and q.

E-step. In the E-step, we calculate the conditional probability of hidden variables $\mathcal{Z}$, given the observed variables $\mathcal{X}$ and estimate of parameter Λ:

$$z_{i,l,j}^{(n)} = \frac{p(x_{i,l}|z_{i,l}^{(n)} = j, \Lambda^{(n)})}{\sum_{j'=0}^{1} p(x_{i,l}|z_{i,l}^{(n)} = j', \Lambda^{(n)})}$$

where $j = 0, 1$. $p(\cdot|z_{i,l}^{(n-1)} = 1, \Lambda^{(n)})$ is the probability mass function of $x_{i,l}$, given it is drawn from the information diffusion component:

$$p(x_{i,l}|z_{i,l}^{(n)} = 1, \Lambda^{(n)}) = \theta_l^{(n-1)}(q_{l,t_{d_i},v_{d_i}})^{x_{i,l}}(1-q_{l,t_{d_i},v_{d_i}})^{1-x_{i,l}}$$

and $p(\cdot|z_{i,l}^{(n)} = 0, \Lambda)$ is the probability mass function of $x_(i,l)$, given it is drawn from the external trend component:

$$p(x_{i,l}|z_{i,l}^{(n)} = 0, \Lambda^{(n)}) = (1-\theta_l^{(n)})(\mu_{l,t_{d_i}}^{(n)})^{x_{i,l}}(1-\mu_{l,t_{d_i}}^{(n)})^{1-x_{i,l}}$$

M-step. By taking partial derivatives of the expectation of log-likelihood, we get the new estimation of parameters.

$$\theta_l^{(n+1)} = \frac{\sum_{d=1}^{N} z_{i,l,1}^{(n)} + \alpha_1 - 1}{N + \alpha_1 + \alpha_0 - 2}$$

and

$$\mu_{l,t_{d_i}}^{(n+1)} = \frac{\sum_{d_i \in C_i} z_{i,l,1}^{(n)} x_{i,l} + \beta_{l,1} - 1}{\sum_{d_i \in C_i} z_{i,l,1}^{(n)} + \beta_{l,0} + \beta_{l,1} - 2}$$

Estimate of Diffusion Probabilities. We now discuss the estimate of diffusion probabilities. It is possible to formulize it as an inference problem for maximum likelihood or MAP estimate. Saito at el. defined a likelihood function for the IC model, and proposed an EM algorithm for inference problem[17]. However, the number of parameters in the diffusion model is very large (one parameter for each edge in the network). The inferring of model is very time-consuming, even on a fixed set of actions. In the LADP model, the set of socially sourced actions changes in each iteration of the EM algorithm. Getting maximum likelihood estimation for the diffusion probabilities in each iteration of the EM algorithm will be an enormous computational challenge. Therefore, we follow the Partial Credit Bernoulli model (PCB) [9] to estimate the diffusion probabilities.

The original PCB model is not based on the maximum likelihood or MAP estimate, so it does not work together with the EM algorithm for inferring the LADP model. We change it so that it can be incorporated into the M-step of the EM algorithm. We will first describe the PCB model, and then show that how we change it so that it can be incorporated into the EM algorithm.

The idea of the PCB model is as follows: if user v takes a positive action about a term l at time t after his in-neighbor u's positive action at time $t-1$, we regard there is a successful diffusion from u to v. If there are more than one in-neighbors of v take positive action at the previous step, they share the credit for the one successful diffusion equally. The diffusion probability $p_{u,v}$ is then given by the ratio of the number of successful diffusion from u to v to the number of positive actions taken by u.

Formally, the diffusion probabilities can be estimated by:

$$p_{u,v} = \frac{\sum_{(i,l)\in A_v}\sum_{d\in Can(i,l)}\frac{I(v_d=u)}{|Can(i,l)|}}{|A_u|} \tag{2}$$

where $A_v = \{(i,l) : x_{i,l} = 1, v_{d_i} = v\}$ is the set of all the positive actions taken by the node v and $I(\cdot)$ is the indicator function. $Can(i,l)$ is the candidate set of documents that share the credit for the successful diffusion. A document is in the candidate set if and only if it is posted by a friend of v_{d_i} and it is posted in the time step right before t_{d_i}, i.e., $Can(i,l) = \{d_j | t_{d_j} = t_{d_i} - 1,\ x_{j,l} = 1,\ u \in IN(v_{d_i})\}$, where $IN(v)$ is the set of in-neighbors of v.

According to Equation 2, all the positive actions are used for the learning of the diffusion model. Since in the LADP model we have the hidden variable $z_{i,l}$ representing whether an action is drawn from diffusion processes or not, we should train the model with the socially source actions only. We then can add the hidden variable $z_{i,l}$ to Equation 2, and replace it with the following equation:

$$p_{u,v} = \frac{\sum_{(i,l)\in A_v}\sum_{d\in Can(i,l)}\frac{I(v_d=u, z_{i,l}=1)}{|Can(i,l)|}}{|A_u|} \tag{3}$$

We then replace $z_{i,l}$ in the above function with $P(z_{i,l}^{(n)} = j|\mathcal{X}, \Lambda^{(n-1)})$, and integrate the learning of diffusion probabilities into the EM algorithm.

5. EXPERIMENT

5.1 Baselines

We compared the proposed LADP model with three baselines: two for the task of learning diffusion model, and the other for the analysis of extracted social events.

the PCB algorithm: We compare the LADP model against the Static PC Bernoulli algorithm (PCB) [9] for the task of learning diffusion model. The PCB algorithm is similar to the inference method described in Section 4, but all positive actions are regarded as socially sourced actions and are used for the probability learning. It is equivalent to the LADP model with $\theta = 1$.

Saito's Algorithm The Saito's algorithm [17] is another baseline that we used for evaluating the task of learning diffusion probabilities. It is an EM algorithm for maximum likelihood estimation of the IC model.

Myers's Algorithm: For better understanding of the LADP model, we also analyze the social events extracted by

the model. The Myers's algorithm [15] is used for comparison in the analysis of extracted social events. This algorithm is not designed for the same purpose as the LADP model. But it can divide the influence to the users into two parts: the internal part and the external part. The internal part can roughly be aligned with the socially sourced portion in our model, so we use it for a comparison in the analysis of extracted social events.

5.2 Datasets

The algorithms are tested on four real world datasets and one semi-synthetic dataset. Each real world dataset consists of a social network and a data stream on the network. The semi-synthetic dataset is based on real world network, but we generate the data stream synthetically. The semi-synthetic dataset is used to evaluate the accuracy of the inference algorithm, while the real world datasets are used to test how well the LADP model works for real applications.

Twitter-UIC dataset: This dataset consists of 974,382 tweets on a network of 2,180 users and 14,572 links. The users in the dataset are the followers of the "UIC news" account on twitter.com. Most of them are students in University of Illinois at Chicago. Directed links in the network correspond to who-follows-whom relationships. They are directed from the one being followed to the follower. The stream data on the network are generated from the tweets posted by the users over the 52 weeks of the year 2011. Hashtags in the tweets are used as the terms. Each week is regarded as a time step.

Twitter semi-synthetic dataset: We first randomly crawled a network from twitter.com that consists of 40,000 users and 544,936 links, and then generate the semi-synthetic dataset using steps as follows. The diffusion probabilities with edges are randomly picked from a Beta distribution with parameters $(1, 30)$. We use the same collection of terms as those in the Twitter-UIC dataset, and the Google Trend profiles of the corresponding terms are used as the external trends profile. The socially sourced actions are generated using the diffusion probabilities with the diffusion model described in 3.2, and the externally sourced action are generated from external trends. The synthetic data contains 223 events, and 14,170,112 positive actions. 68.1% of the positive actions are socially sourced actions, while others are externally sourced actions.

DBLP datasets: We extract three datasets from the DBLP database. Each of them contains the stream of publications in a certain area and the co-author network of that area. In the co-author network, each node corresponds to one author, while each undirected edge corresponds to a co-author relationship between two authors. The titles of publications are used as the textual stream. We remove stopwords from the titles, and use bigrams as terms. The three datasets we used for the evaluations are as follows.

• Data mining community dataset: This community contains 14,011 publications and their authors in 10 data mining conferences over 15 years (1995-2010). Each year is regarded as a time step. Conferences are listed in Table 2. The co-author network contains 6,948 nodes and 39,797 edges. Authors that have less than 3 publications are filtered out.

• Machine learning community dataset: It is similar to the data mining community, but are based on 10 machine learning conferences, as listed in Table 2. This community contains 24,184 publications, 6,845 nodes and 34,254 edges.

• Mixed dataset (data mining + machine learning): We want to test the algorithms on a more complicated community, because the above two simple communities share several desirable properties: Authors in each community are from the same area, and are strongly connected to each other; Documents in each community are from similar topics, so the events are easier to be detected. By extracting publication in the 20 listed conference and filtering out authors with less than 5 publication, we get a network that consists of 7664 nodes and 47,158 edges.

Data Mining		Machine Learning	
KDD	CIKM	AAAI	UAI
SDM	ICDE	IJCAI	KR
ICDM	WWW	ICML	IROS
SIGMOD	PKDD	ICRA	ATAL
VLDB	PAKDD	ICGA	AAMAS

Table 2: Conference Lists of Two Communities in DBLP dataset

5.3 Experiment with Twitter Datasets

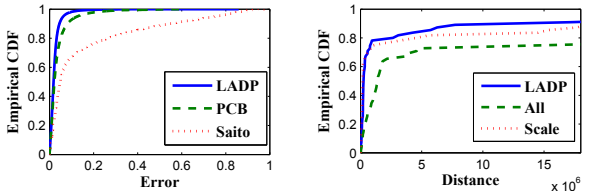
5.3.1 Experiment with Semi-synthetic Dataset

To test the accuracy of the inference algorithm, we evaluate the LADP model on the semi-synthetic dataset. As the diffusion probabilities and external trends for this dataset are known, we can evaluate the inferred values directly.

Result of diffusion model learning. The result is shown in Figure 2(a), for the diffusion probability with each edge in the network, we calculate the prediction error as the difference between the real value and the inferred value. We plot the empirical cumulative distribution function of the prediction error in Figure 2(a). Each point in the curve shows a value and the percentage of the prediction errors that are below the given value. As shown in the figure, the diffusion probabilities inferred by LADP model have much smaller error than the ones inferred by the baselines. For example, for 92.0% edges, the prediction errors of the LADP model are smaller than 0.05, while only for 79.8% and 50.8% edges respectively the prediction errors of the PCB algorithm and the Saito's algorithm are within this range.

The only difference between the LADP model and the PCB algorithm is that the LADP model extracts the social events from the entire events, and learns the diffusion probabilities from the social events, while the PCB algorithm learns the diffusion probabilities from all actions in the events. It implies that the improvement of accuracy by the LADP model is the result of extracting social events.

Extraction of social events. For better understanding of how the LADP model makes the improvement, we show the difference between inferred socially sourced portion and the ground truth in Figure 2(b). To calculate the difference, we first define the time sequence $\{a_t\}_{t=1}^{T}$ of a set of actions. For each time step $t = 1, \ldots, T$, there is an element a_t in the time sequence, which is the number of actions in the set that are created at time step t. We then calculate for each event the L2 distance between the time sequence of the extracted socially sourced portion and that of the ground truth, and plot the empirical distribution of the distance in Figure 2(b).



(a) Errors of Diffusion Probability (b) L2 Distance for Socially-Sourced Portion

Figure 2: Evaluation on Synthetic Stream Data

For the proposed LADP model, only the extracted socially sourced portion are used for the learning of diffusion model, while for the PCB and Saito's algorithms the entire events are used for the learning, in other words, they consider the entire events to be social events, so we also show the L2 distance between the time sequence of the entire event and that of socially sourced portion.

Since the time sequence of entire events is always an overestimation of the time sequence of social sourced portion, we can easily get a better estimation by simply scaling it. We then scale the time sequence for each event and make it have the same mean value as the time sequence of the social sourced portion, and calculate the L2 distance between the scaled time sequence and that of socially sourced portion as well.

As shown in Figure 2(b), the distance between the time sequence of socially sourced events inferred by LADP and that of the ground truth is smaller than the distance between the time sequence of the entire events and that of socially sourced portion, even when we scale the time sequence of entire event. This reflects that the LADP model can extract social events from the entire events, and the extraction is more than divide the entire events into two parts according to the proportion of socially sourced portion and externally sourced portion. The extraction of social events by the LADP model results in better accuracy in learning the diffusion model.

5.3.2 *Experiment with Twitter-UIC dataset*

Result of diffusion model learning. For the real stream data, the diffusion probabilities are unknown, so we are not able to evaluate a learned diffusion model directly. Instead, we evaluate it by evaluating the most influential node suggested by the models. In the Twitter community, users can re-post tweets of other users, which is called "retweet". The more retweets a tweet gets, the more widely it spreads in the network. We can expect that users with larger number of retweets per tweet are more influential in the social network.

Given the learned diffusion model, by sampling the diffusion process for 50,000 times, we calculate the average influence of each node, i.e. the average number of activations when using each single node as the seedset. The nodes are then sorted according to the descending order of their influences. In this way, we find out the most influential nodes in the models learned by the LADP, PCB and Saito's algorithms. We then evaluate the most influential node by the average number of retweets for each tweet. Figure 3(a) shows the comparison between the LADP method and the baselines for finding the most influential nodes. For each number k on the x-axis, we calculate the average number of retweets for the top k users and plot it in the figure. In the most range of x-axis, the LADP model achieves a larger average number than the baselines. Besides, the curve of the LADP model is a monotone decreasing curve. It is more desirable than the non-monotonic curve of the PCB model, because the monotone decreasing curve suggests that higher-ranking nodes inferred by the algorithm are really more influential than lower-ranking nodes. The better performance in finding top influential users suggests that the diffusion model learned by LADP is better than those learned by the baselines.

	All	Myers's	LADP
1	Chicago	Chicago (-)	Chicago (-)
2	FF	FF (-)	UIC (↑)
3	UIC	UIC (-)	FF (↓)
4	energy	energy (-)	higherEd (↑)
5	higherEd	higherEd (-)	Illinois (↑)

Table 3: Top events and top social events identified by LADP and Myers's in UIC Twitter network

Analysis of top social events. We analyze the top social events extracted by the LADP model in order to understand how the LADP model improves the learning of diffusion model. First, we rank the events according to the numbers of all actions in the events, and list the top events in the network. Then, we rank the events according to the number of actions in the inferred socially sourced portion, i.e. we rank the social events extracted by the LADP model. After excluding the externally sourced actions, the list of top social events extracted by the LADP model is different from the list of top events with the largest number of actions. For a comparison, we also list the top events with the largest internal portion inferred by the Myers's algorithm [15].

The results are shown in Table 3. The first column in the table is the list of keywords of events with the largest number of actions. Some keywords in the list are closely related to the UIC community (Chicago, UIC, higherEd[1]), but others are not (FF[2], energy). The second column is the list of top five internal events returned by the Myers's algorithm. For this dataset, the lists in the second column happens to be the same as the list in the first column, but this is not always the case, as we will show in the following experiment. In the third column, we list the top social event extracted by the LADP model. Comparing with the first column, the keywords "UIC", "higherEd", and "Illinois" move upward in the rankings, while the keywords "FF", "energy" move downward. It is obvious that keywords that are related to the community get better rankings in the list returned by the LADP model. Since the keywords that are closely related to the community is more likely to suggest social events inside the network, the top events extracted by the LADP model are more likely to reflect information diffusion processes inside the social network. (Notice that though the keyword "FF" reflects an event on the Twitter website. It is not unique to the UIC community, and its propagation does not suggest an information diffusion process inside the Twitter-UIC network. Further analysis on the keywords "UIC" and "FF" will be provided in next section.) This explains that the improvement of LADP in the learning of diffusion probabilities.

5.3.3 *Case Study*

In Figures 3(b) and 3(c), we show the results of "UIC" and

[1]higherEd is short for higher education in this context.
[2]FF is short for FollowFriday in Twitter. It is an online event that people recommend friends for other users.

"FF" inferred by the LADP model. For each of the events, we show the number of actions over time for the socially sourced portion, externally sourced portion, and the entire event. As shown in the figure, for the event of "UIC", the socially sourced portion is the majority, while for the event of "FF", the size of the two portions are similar. Beside, for the event of "UIC", the socially sourced portion explains most of the peaks in the full event, while for the event of "FF", some peaks are explained by the influence while others are explained by the external trends. This is the reason why the ranking of "UIC" moves upward in the list of top social events returned by LADP, while "FF" moves downward.

5.4 Experiment with DBLP Datasets

Result of diffusion model learning. Similar to the Twitter-UIC dataset, since there is no ground truth for the diffusion probabilities, we evaluate them by looking at the most influential nodes. For the most influential users suggested by each algorithm, we evaluate the top-k influential nodes using their H-index and the number of citations. The H-index and the number of citations of author are obtained from arnetminer.org.

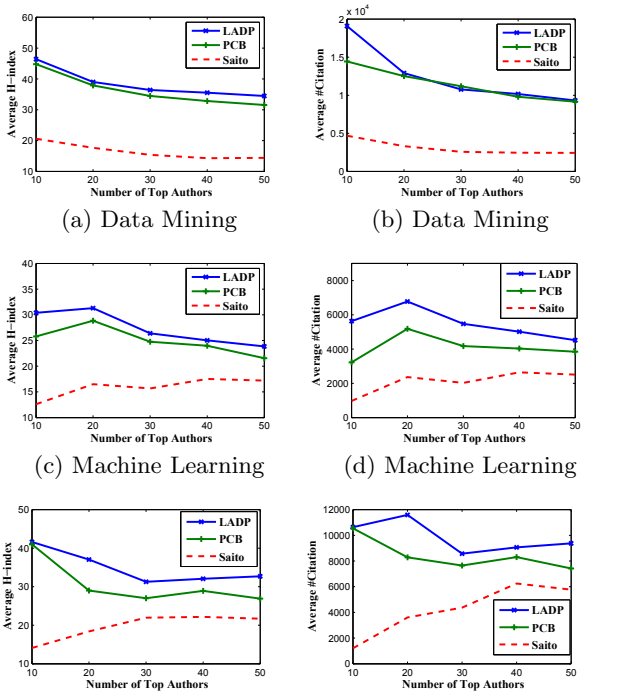
On all the three datasets, we run the LADP model, the PCB algorithm and Saito's algorithm respectively to learn the diffusion probability with each edge. These three sets of diffusion probabilities can then be used for deciding most influential nodes. By sampling the independent cascade process for 50,000 times, we calculate the average influence of each node, i.e. the average number of activations when using each single node as the seedset. We then sort nodes to the descending order of influence and plot the average H-index and the average number of citations for top-k authors.

Figures 4(a)-(f) show the comparison of the LADP model and the baselines on three datasets. The x-axes of these figures are the number of top authors, while the y-axes are the average H-index or the average number of citations of these top authors. On all but one figure, the curves of the LADP method are obviously above those of the baselines, which means the top-ranking authors reported by the LADP model are more influential than those reported by the baselines. Even for that Figure 4(b), LADP performance on the top 10 authors is significantly better than the baselines. It reflects that the diffusion model learned by LADP is more accurate than the one learned by the baselines.

Analysis of top social events. Similar to the experiment on Twitter dataset, in order to understand how the LADP model improves the learning of diffusion model, we analyze the top social events extracted by the LADP model. Since we are more familiar with topics in the data mining, we use the data mining community for the analysis.

The results are shown in Table 4. For each network in the dataset, the first column in the table is the list of keywords of events with the largest number of actions. The second column is the list of top five internal events returned by the Myers's algorithm. The top five social events extracted by the LADP model are listed in the third column.

The rankings of keywords related to specific research topics in the data mining community are moved upward by the LADP algorithm (data streams, time series, association rules), while keywords that are less related to specific topics move downward. That is desirable because specific topics are more likely to be connected to information diffusion processes in the network, and represent social events. The



(a) Data Mining (b) Data Mining

(c) Machine Learning (d) Machine Learning

(e) Data Mining + Machine Learning (f) Data Mining + Machine Learning

Figure 4: (a)(c)(e) H-index for Top Authors, (b)(d)(f) Number of Citations for Top Authors.

	All	Myers's	LADP
1	data mining	query processing (↑)	data streams(↑)
2	data streams	association rules (↑)	time series (↑)
3	time series	text classification (↑)	data mining(↓)
4	query processing	xml data (↑)	association rules (↑)
5	association rules	pattern mining (↑)	query processing(↓)

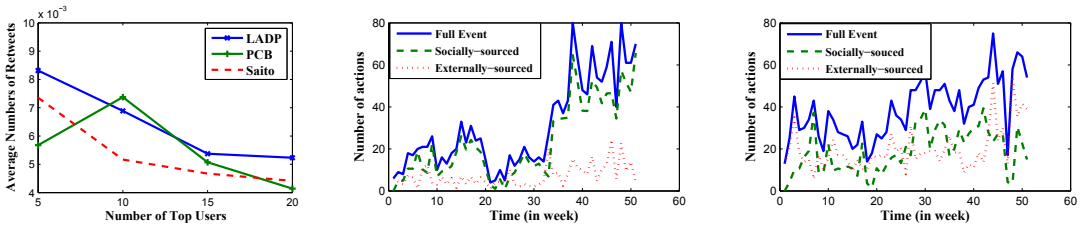
Table 4: Top events and top social events identified by LADP and Myers's in data mining community

Myers's algorithm also tends to give specific topics higher ranking, but it undesirably gives higher ranking to the topics "query processing" and "xml data", which are related to the database community, rather than the data mining community specifically.

6. RELATED WORK

Information diffusion has been intensively studied in social network analysis [3, 5, 4, 12, 1]. Earlier work on information diffusion model does not consider the time dynamic of diffusion processes. Recent work in [14, 11, 19] consider the diffusion processes that unfold along the time, so that temporal events in the social network can be explained by the information diffusion processes. Independent Cascade (IC) model and its variants [10, 14, 7, 6, 13] form most widely used class of information diffusion models. Models in this class share two features: (1) the influences from the neighbors of a user are independent; (2) there is a diffusion probability along with each edge in the network.

The problem of estimating the diffusion probabilities has been studied in [9, 17, 16, 8]. The diffusion probabilities are learned from events in social networks. Models in [9] estimate the diffusion probability for general threshold models



(a) Most Influential Nodes (b) UIC (c) FF

Figure 3: (a) Average number of retweets for top users. (b)(c) Case study on the events "UIC" and "FF"

which include the IC model and almost all of its variants. [17, 16] propose a likelihood maximization approach for the learning of diffusion probabilities. However, due to the large number of parameters and the complexity of the likelihood function, the inference algorithm is time-consuming.

Although it has long been argued that the information diffusion process is not the only reason triggering events in social networks [2, 18, 1], most existing work on the learning of diffusion probabilities neglects the propagation of information from external trends. Work in [15] explicitly models the external trends and incorporates it with the information diffusion model. While their approach adopts a simple information diffusion model and focuses on inferring the external trends, our model aims to learn the diffusion probabilities with edges in the networks, and the learned diffusion probabilities can be used in the IC model and its variants.

7. CONCLUSION

In this paper, we study the problem of learning information diffusion models on social networks. We propose an LADP model that improves the learning by extracting social events from data streams. The LADP model integrates the external trends and the information propagation process inside the social network.

Evaluation on real and synthetic datasets shows that the LADP outperforms existing method on the task of learning information diffusion models. Analysis shows that the improvement is due to the extraction of social event.

A possible future work is to use more sophisticated model for the external trends instead of the simple Beta distribution. We also expect a further extension to the LADP model that uses topic modeling methods, instead of considering each keyword independently.

8. ACKNOWLEDGMENTS

This work is supported in part by NSF through grants IIS-0905215, CNS-1115234, IIS-0914934, DBI-0960443, and OISE-1129076, and US Department of Army through grant W911NF-12-1-0066.

9. REFERENCES

[1] A. Anagnostopoulos, R. Kumar, and M. Mahdian. Influence and Correlation in Social Networks. In *KDD '08)*, 2008.

[2] S. Aral, L. Muchnik, and A. Sundararajan. Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. *PNAS*, 106(51):21544–21549, 2009.

[3] E. Bakshy, B. Karrer, and L. A. Adamic. Social Influence and the Diffusion of User-Created Content Categories and Subject Descriptors. In *EC '09*, 2009.

[4] M. Cha, H. Haddadi, F. Benevenuto, and K. Gummadi. Measuring User Influence in Twitter: The Million Follower Fllacy. In *ICWSM '10*, 2010.

[5] M. Cha, A. Mislove, and K. P. Gummadi. A Measurement-driven Analysis of Information Propagation in the Flickr Social Network. In *WWW '09*, 2009.

[6] W. Chen, C. Wang, and Y. Wang. Scalable Influence Maximization for Prevalent Viral Marketing in Large-scale Social Networks. In *KDD '10*, 2010.

[7] W. Chen and Y. Wang. Efficient Influence Maximization in Social Networks Categories and Subject Descriptors. In *KDD '09*, 2009.

[8] L. Dickens, I. Molloy, J. Lobo, P.-C. Cheng, and A. Russo. Learning stochastic models of information flow. In *ICDE*, 2012.

[9] A. Goyal, F. Bonchi, and L. Lakshmanan. Learning Influence Probabilities in Social Networks. In *WSDM '10*, 2010.

[10] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the Spread of Influence through a Social Network. In *KDD '03*, 2003.

[11] G. Kossinets, J. Kleinberg, and D. Watts. The Structure of Information Pathways in a Social Communication Network. In *KDD '08*, 2008.

[12] M. Lahiri, A. Maiya, R. Sulo, Habiba, and T. Y. B. Wolf. The Impact of Structural Changes on Predictions of Diffusion in Networks. In *ICDMW '08*, 2008.

[13] T. Lappas, E. Terzi, D. Gunopulos, and H. Mannila. Finding Effectors in Social Networks. In *KDD '10*, 2010.

[14] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. Vanbriesen, and N. Glance. Cost-effective Outbreak Detection in Networks. In *KDD '07*, 2007.

[15] S. Myers, C. Zhu, and J. Leskovec. Information Diffusion and External Influence in Networks. In *KDD '12*, 2012.

[16] K. Saito, M. Kimura, K. Ohara, and H. Motoda. Learning continuous-time information diffusion model for social behavioral data analysis. In *ACML*, 2009.

[17] K. Saito, R. Nakano, and M. Kimura. Prediction of information diffusion probabilities for independent cascade model. In *KES*, 2008.

[18] C. R. Shalizi and A. C. Thomas. Homophily and contagion are generically confounded in observational social network studies. *Sociological Methods and Research*, 40(2):211–239, 2011.

[19] T. Wang, M. Srivatsa, D. Agrawal, and L. Liu. Microscopic Social Influence. In *SDM '12*, 2012.