Joint Post and Link-level Influence Modeling on Social Media

Liangzhe Chen* B. Aditya Prakash†

Abstract

Microblogging websites, like Twitter and Weibo, are used by billions of people to create and spread information. This activity depends on various factors such as the friendship links between users, their topic interests and social influence between them. Social influence can be thought of as a latent factor, that may alter users posting and linking behaviors. Making sense of these behaviors is very important for fully understanding and utilizing these platforms.

Most prior work in this space either ignores the effect of social influence, or considers its effect only on link formation or post generation. In contrast, we propose PoLIM, leveraging simple weak supervision, a novel model which *jointly* models the effect of influence on *both* link and post generation. We also give PoLIM-FIT, an efficient parallel inference algorithm which scales to large datasets.

In our experiments on a large tweets corpus, we detect meaningful topical communities, celebrities, as well as the influence strengths patterns among them. Further, we find that there are significant portions of posts and links that are caused by influence, and this portion increases when the data focuses on a specific event. We also show that differentiating and identifying these influenced content benefits other specific quantitative downstream tasks as well, like predicting future tweets and link formation, where we significantly outperform state-of-the-art.

1 Introduction

Modeling microblogging data, such as Twitter and Weibo, has attracted great attention in recent years [9, 24, 27]. Social media data is easy to obtain, and understanding how the data is generated provides insights to many applications such as community detection, influence maximization, public-health surveillance, etc. Ultimately we want to understand how people generate content and how online information diffuses across the underlying social network.

This problem is compounded by the fact that due to social influence, individual's behaviors and attributes may conform to her neighbors' [16,18]. In the context of social media, social influence can be thought of as a latent factor, that may alter users' posting and linking behaviors. For example, a Twitter user may follow Barack Obama simply because he is a celebrity with high popularity, regardless of his/her own interests. Similarly, a Twitter user may retweet a close friend because of their mutual friendship regardless of whether the tweet content is of her interest. Without understanding how such latent social influence affects the generation of posts and links, we cannot fully and correctly understand the complex information patterns.

To this end, we propose a novel generative model PoLIM (**Po**st and **L**ink level **I**nfluence **M**odel) which extracts and models the latent influence that affects the generation of *both* posts and links. We assume the existence of some weak supervision (like tweets with the 'RT' label in Twitter), that are more likely to be affected by social influence [20]. We use them to guide the inference of PoLIM, which then generalizes and learns the latent influence for all the posts as well as the follower-followee links. Modeling the extent of this latent influence for both posts and links helps us learn better topic interests for users and communities. This helps us achieve higher performance on other downstream tasks too, such as predicting the link formation and retweet generation in the future. Informally, our goal of proposing PoLIM can be stated as:

Problem 1. *Given a microblogging dataset (such as Twitter and Weibo), with an underlying directed friendship network $G(E, V)$; and set of textual posts $\mathcal{T}_i$ from each user n_i, identify social influence among users, and the posts and friendship connections that are caused by social influence.*

Surprisingly, most of the existing models for social media datasets either completely ignore the effect of social influence (they assume all behaviors are driven by self interests), or only consider its effect on one of the two aspects (links or posts). The most closely related model is COLD by Hu et al. [14]: although they propose a model that covers social influence, links and posts (like us), they only use social influence to control link generation (in contrast, we use it to control each link and post). Moreover we learn this social influence at multiple granularities including at the community and individual level, helping us better understand content generation. Doing so ultimately helps us to get better prediction and analysis of the diffusion. For example,

*Pinterest. Email: bottlestar@gmail.com

†Department of Computer Science, Virginia Tech. Email: badityap@cs.vt.edu.

Copyright © 2019 by SIAM
Unauthorized reproduction of this article is prohibited

we show in our experiments how PoLIM can achieve better link prediction and retweet volume prediction than state-of-the-art competitors which do not perform this integrated modeling.

In summary, our main contributions include proposing a novel influence-based model PoLIM and an efficient inference algorithm to jointly model post and link generation, using it to understand a large Twitter dataset (containing more than 27 million tweets) and also demonstrate better predictive performance. The rest of the paper is organized in the usual way, and we omit some derivations and experiments, for space.

2 Related Work

Topic Models. LDA (Latent Dirichlet Allocation) models have been widely studied and applied in different domains (see [8] for a review). To better adapt topic models to microblogging data, where a document/tweet is typical very short in length, Zhao et al. [33] assume that each tweet has only one hidden topic assignment. Based on their work, Qiu et al. [24] further introduce the behavioral aspect of tweets into the model, while Qu et al. [25] enrich the model by adding location and temporal topics. Another rich line of work focus on dynamic topics [4, 7, 11, 27], and dynamic topic distributions [13] where either the word distributions of topics, or the topic distributions themselves evolve over time. Paul et al. [23] and Chen et al. [9] use weak supervision together with markov models and topic models to capture health and ailment aspects of twitter users. All of these works only model the generation of the tweet content without considering the network structure or the social influence.

In contrast, Nallapati et al. [21], Zhu et al. [34] and Bi et al. [6] jointly model the word generation and the link formation in Twitter. In this line, the most closely related work include COLD by Hu et al. [14], a generative model that considers text content, temporal information, link structure and community level influence; and a topic-level influence model for heterogeneous networks [17]. The former models social influence's affect on link generation, while the latter models social influence for text generation. To the best of our knowledge, our model is the first that uses weak supervision to integrate all three aspects: text content, social links and user influence, where the user influence controls *both* the text content and the social links.

Influence Analysis. The influence maximization problem (as opposed to simply tie strengths) aims to identify global influencers that would maximize the spread of the information based on Linear Threshold or Independent Cascade models [15]. [19] generalizes to deal with group of nodes.There is also much interest in further identifying influencers based on topics. Weng et al. [28] propose TwitterRank to find topic-level influencers in twitter. Tang et al. [26] model topic-level social influence on large networks. He et al. [12] learn influence function from incomplete observations. Zhang et al. [32] propose three sampling algorithms to detect structural influence. Pal et al. [22] propose a set of designed features to characterize social media authors, and use probabilistic clustering and with-in cluster ranking to identify topical authorities. While these works either focus on global or topic-level influencers, we model social influence in a more fine-grained level between all the users. The combination of our community-level influence, with-in community popularity, and other parameters in our model can be used to compute the influence probability between any two users.

3 Model Formulation

We formulate our proposed model in this section. Our main hypotheses are 1) social influence controls both the post generation, and the social link formation, and 2) we have some supervision on users' posts, which are good indicators of whether the post is generated by social influence or not. Given these hypotheses, we formulate our model **Po**st And **L**ink level **I**nfluence **M**odel (PoLIM) to jointly model the post content, social structure and the social influence, using weak supervision from the influence indicators. We first explain the main concepts in PoLIM in the following. The notations we used are shown in Table 1. For simplicity, we only show the most important symbols, and skip the prior parameters (α, β, η, etc.) and those symbols explained in the text (λ, c^*, c', etc.).

3.1 Main Concepts

We denote the social network as a directed graph G, where each node represents a user n_i in the social network, and a directed edge represents a following relation. Every user n_i has a sequence of posts $\mathcal{T}_i = \{t_{i1}, t_{i2}, ...\}$, and each post t_{ij} contains a sequence of words. In the following, we define the most important concepts in PoLIM.

Communities. Communities sharing similar interests naturally exist in social networks. Hence, to make the model expressive but still tractable, in PoLIM, we define the following community concept to aggregate users with the same topic interests.

Definition 1. (Community) *A community c_i is a group of users, who share the same topic interest θ_i ($Z \times 1$ vector), where θ_{ij} represents the probability of generating a post with topic z_j.*

In spite of the above definition, a user can still be influenced by another user. In the following, we define the user-to-user influence through the lens of the community

Copyright © 2019 by SIAM
Unauthorized reproduction of this article is prohibited

Table 1: Terms and Symbols

Symbol	Definition and Description
G	The follower-followee network
N	Number of users in G
n_i	User i
T_i	Number of posts by n_i
Z	Number of topics
W	Number of unique words
K	Number of communities
E_i	Number of links from n_i
c_i	Thei_{th} community
θ_i	The topic interest of c_i: Zx1 probability vector
ϕ_i	The word distribution for z_i: Wx1 probability vector
z_i	The i_{th} topic
I	K by K influence matrix
$N(n_i)$	Neighbors of n_i in G
t_{ij}	The j_{th} tweet of n_i
e_{ij}	The j_{th} followee of n_i
v_i	The probability that n_i is not influenced by another user
ρ	The probability that a user follows someone in her own community (vs. random following)
u	The probability of generating word from the background topic
r	The switch value for each tweet, when $r = 0$, the tweet is caused by self interest, otherwise social influence
ϵ	The switch value for each link, when $\epsilon = 0$, the link is caused by social influence
A_i	The $N \times 1$ celebrity vector for c_i
M	The community distribution: Kx1 probability vector

concept. Given an instance that n_a in c_1 is influenced by n_b in c_2 to generate a post/link, we decompose such an influence to two steps: *community-to-community influence*, and *user-in-community selection*.

Community-to-community influence. When a user n_a in community c_i gets influenced by another user, we first select where (which community) the influence comes from. We define the following influence matrix I to captures the probability of a community being influenced by another.

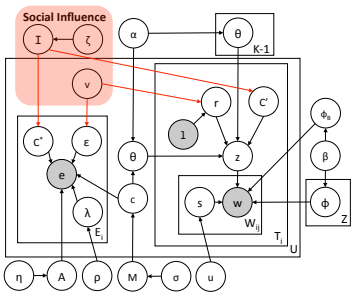
Definition 2. (Influence matrix) *An influence matrix I is a K by K matrix, where I_{ij} represents the probability of a user n_a in c_i being influenced by some user in c_j given that n_a is influenced by another user.*

Note that the diagonal entries of I can be non-zero, i.e. we allow a user to be influenced by someone in the same community. This matches the fact that one may be influenced by someone with either different, or similar interests. For example, n_a in the data mining community may be influenced by n_b in the politics community to retweet a political news, and n_a can also retweet his/her colleague n_c from the same community about a new paper in the field. Both cases (intra and inter community influence) are commonly seen in social media [5].

User-in-community selection. Once we decide which community the influence comes from, we select an influencer from the influencing community. Notice that different members in a community have different powers to influence other users (for example, the dean's tweets are more likely to be retweeted than a student's), we define the following 'celebrity' vector to capture users' popularity in his/her community.

Definition 3. (Celebrity vector) *Each community c_i has a celebrity vector A_i (a U by 1 vector), which shows the popularity of users in c_i.*

A user with higher popularity in a community is more influential than other members in affecting others' behaviors, and attracts more followers and retweets. Naturally, a user can only be a celebrity in her own community (having non-zero values in A_i), but can still be influential in other communities via social influence I.



3.2 Our Model PoLIM Now we describe our PoLIM model. The plate notation of our model is shown left. The main feature of PoLIM is that social influence contributes to the generation of each post and link. We will first explain how each post and link can be generated without any supervision (generation process shown in Alg. 1), and then explain why and how we introduce the weak supervision l to the model. To generate the data, we first initialize $\{\phi, \theta, I, A, c, u, v, \rho\}$ using the prior parameters (α, β, etc.). We use the standard conjugate priors (Beta and Dirichlet distribution) for Bernoulli and Multinomial distributions.

Generating Posts. Considering the characteristics of posts on microblogging websites such as Twitter and Weibo, we make the following two assumptions about the posts: 1) each post has only one latent topic (also commonly used in past work [24, 25, 33]) and 2) each word in a post can be generated either by the corresponding latent topic, or by a background topic which generates common words like 'I', 'and', 'to', 'for', etc. Each user n_i has a probability v_i to behave from his/her self interest. Using this v_i probability, we draw the switch values r and ϵ, which represent whether the corresponding post and link are generated from social influence or not. If a post is generated from self interest, we draw a topic from the user's own topic interest (the same as the topic interest of the community he/she belongs to), and generate the post accordingly. On the other hand, if the post is influenced by another user, we select an influencing community c' according to the influence matrix I, and generate the post based on the topic interest of c'.

Copyright © 2019 by SIAM
Unauthorized reproduction of this article is prohibited

Generating Links. Similarly for the link generation, we first draw the switch value ϵ. If $\epsilon = 0$, the link is generated from social influence, and we first choose an influencing community c^* according to I, and then choose an influencing user in c^* to follow based on the corresponding A^* vector. When $\epsilon = 1$, differently from the post generation above, the link is considered either generated from self interest or random following (decided by the switch value λ). If the link is generated from self interest, we select a celebrity user in $n_i's$ own community to follow based on the corresponding A vector; otherwise, we choose a random user in the network to follow.

Adopting Weak Supervision. To correctly learn social influence, a big challenge for PoLIM is to learn when a post/link is influenced (namely to learn the switch values r, ϵ correctly). In general, distinguishing social influence from other compounding variables is a very hard task [1,3]. Without any guidance, the change of the data likelihood caused by an arbitrary change of these switch values, can be undesirably compensated by updating the other parameters accordingly. In this paper, motivated by the usage of aspects in topic models [23], we assume the existence of good influence indicators/markers l for each post. Intuitively, if $l = 1$, it suggests that the post is *likely* (not necessarily) generated by influence, and we are more likely to learn $r = 1$ for the post, and vice versa. Retweets have been regularly used as an proxy for influence in Twitter social influence studies [20] in past. Hence in our experiments, we simply use the RT label as a weak influence indicator ($l = 1$ if the tweet contains the RT label), and bias learning r using these equations (with $\tau = 0.1$):

$$p(r = 0|l = 0) = 1 - \tau + \tau v; \quad p(r = 0|l = 1) \quad = \tau \cdot v$$

The probability of $p(r = 1|l)$ can be calculated accordingly. Note that this (weak) supervision only applies to r, however, it also affects the learning of ϵ because both switch values are controlled by the same influence probability v. Therefore, although we use l as the weak supervision, we would be able to leverage both the posts and links information to learn beyond l and extract social influence that best describes the data.

4 PoLIM-FIT: Model Inference

The main parameters in PoLIM are $\{\theta, I, A, \phi, z, r, s, c', c^*, c, \epsilon, \lambda, u, v, \rho\}$ (other prior parameters can be inferred once these parameters are learned). To fit PoLIM on real datasets (e.g. given tweets and the follower network), we propose PoLIM-FIT to automatically learn all these parameters from the data in linear time. Further, we improve the efficiency of PoLIM-FIT by parallelization.

Due to intractability of exact inference in such models [4], we propose a Collapsed-Gibbs-Sampling-based

Algorithm 1 Generative process for PoLIM

```
Initialize φ, θ, I, A, c, u, v, ρ using prior parameters like α, β, etc.
//Generate tweets
for each user n do
    for each tweet t do
        Choose an indicator r ~ Ber(v_n) //Bernoulli distribution
        if r=0 //from self interest then
            Choose a topic z ~ Multi(θ_n) //Multinomial distribution
        else
            Choose an influencing community c' ~ I_n//from social influence
            Choose a topic z ~ Multi(θ_{c'})
        for each word w do
            Choose an indicator s ~ Ber(u)
            if s=0 then
                Choose a word w ~ Multi(φ_B)//background word
            else
                Choose a word w ~ Multi(φ_z)
//Generate links
for each user n do
    for each link l of user n do
        Choose ε ~ Ber(1 − v_n)
        if ε = 0 then
            Choose a community c* ~ I_n//from social influence
            Choose an influencing user in the community to follow e_{n,l} ~ Multi(A_{c*})
        else
            Choose λ ~ Ber(ρ)
            if λ = 0 then
                Choose an influencing user from n's own community to follow e_{n,l} ~ Multi(A_{c_n}) //self interest
            else
                Choose a random user to follow.
```

[10] algorithm PoLIM-FIT (Alg. 2) to learn the model parameters. PoLIM-FIT first marginalizes several parameters ($\{\theta, I, A, \phi\}$) when sampling other parameters ($\{z, r, s, c', c^*, c, \epsilon, \lambda, u, v, \rho\}$). Once the sampling process converges to stable values, we then estimate the marginalized parameters from the sampled values. For lack of space, we only show the final sampling equations for two of the important parameters in PoLIM.

Algorithm 2 Pseudo-code for PoLIM-FIT

1: Initialize prior parameters like α, β, etc.
2: Sample values for $\{z, r, s, c', c^*, c, \epsilon, \lambda, u, v, \rho\}$.
3: Repeat step 2 until convergence.
4: Sample values for the marginalized variables $\{\theta, I, A, \phi\}$.

Community assignment c_i. Each user n_i has a unique community assignment c_i. Given the other parameters, c_i is sampled using the following probabilities.

$$p(c_i|...) \propto \prod_{t_{ij},r=0} \frac{\alpha + N^{z_{ij}}_{-n_i,c_i}}{Z\alpha + N_{-n_i,c_i}} \cdot \prod_{t_{ij},r=1} \frac{\zeta + N^{c'_{c_i}}_{-n_i,c_i}}{K\zeta + N_{-n_i,c_i}} \cdot$$

$$\prod_{e_{ij},\epsilon=1,\lambda=0} \frac{\eta + N^{e_{ij}}_{-n_i,c_i}}{|c_i|\eta + N_{-n_i,c_i}} \cdot \prod_{e_{ij},\epsilon=0} \frac{\zeta + N^{c^*_i}_{-n_i,c_i}}{K\zeta + N_{-n_i,c_i}} \cdot M(c_i)$$

where $N^{z_{ij}}_{-n_i,c_i}$ denotes the number of times the topic z_{ij} is generated by a user in c_i; $N^{c'_{c_i}}_{-n_i,c_i}$ denotes the number of times that c'_{c_i} is influenced by c_i; and $N^{e_{ij}}_{-n_i,c_i}$ is the number of times that e_{ij} is chosen from c_i by other

Copyright © 2019 by SIAM
Unauthorized reproduction of this article is prohibited

users to follow. All the $-n_i$ means excluding n_i in the counting. Intuitively, it goes over all the tweets and edges from n_i, and calculates the likelihood of the user belonging to c_i given all the switch values. Note that the edges caused by random following are not used in the Eq. because their likelihoods are independent to c_i.

Latent topic z_{ij}. Each post t_{ij} has a latent topic z_{ij} which can be sampled as:

$$p(z_{ij}|...) \propto \prod_{w_{ijt},s_{ijt}=1} \frac{\beta + N^{w_{ijt}}_{-t_{ij},zij}}{W\beta + N_{-t_{ij},zij}} \cdot$$

$$(\frac{\alpha + N^{z_{ij}}_{-n_i,c_i}}{Z\alpha + N_{-n_i,c_i}})^{\mathbb{1}(r_i=0)} \cdot (\frac{\alpha + N^{z_{ij}}_{-n_i,c'_i}}{Z\alpha + N_{-i,c'_i}})^{\mathbb{1}(r_i=1)}$$

where $\mathbb{1}()$ is an indicator function. Basically, we first go over all the words in t_{ij} that are not generated by the background topic ($s_{ijt} = 1$), and calculate the likelihood of t_{ij} being generated by z_{ij}; then based on the r value, if t_{ij} is generated from influence, we calculate the likelihood of c_i generating the topic z_{ij}, otherwise the post is influenced, and we calculate the likelihood of the influencing community c'_i generating z_{ij}.

In sum, PoLIM-FIT has a linear time complexity of $O(R(WZ+TK+EK+N))$, where R is the number of iterations the algorithm runs. Note that theoretically, sampling the marginalized parameters I is quadratic to the number of communities, but since it only needs to be sampled once after the sampling process converges, it is not the bottleneck of the running time.

Speeding Up & Parallelization. To further improve the running time, we implement a parallel version of PoLIM-FIT. Since many of PoLIM's parameters are based on users, we allocate data from different users to different worker processes, and each individual process samples the parameters related to its users simultaneously. The counters used in the sampling process (such as $N^{z_{ij}}_{-n_i,c_i}$) are maintained as global variables that are shared by all the processors. The final time complexity for our sampling algorithm is therefore reduced to $O(\frac{R}{p}(WZ + TK + EK + N))$, where p is the number of processes.

5 Empirical Study

We implement PoLIM in Java[1]. Our experiments were conducted on a 4 Xeon E7-4850 CPU with 512GB of 1066Mhz main memory. We design various experiments to answer the following questions.

- **Q1** [Downstream task 1] Can PoLIM improve the performance of link prediction?
- **Q2** [Downstream task 2] Can PoLIM improve the performance of retweet volume prediction?
- **Q3** In the entire dataset, what is the extent of social influence?
- **Q4** Does PoLIM find meaningful topics?
- **Q5** What are the influence strengths among different communities? Who are the celebrities?
- **Q6** Can PoLIM help understand the content generation and communities during a specific event?

With regards to scalability, in short we found our algorithm scales linearly with both # topics and # communities, and also has a near-linear parallelized speed-up.

5.1 Set-up Dataset.

We use a Twitter dataset *Tweets-Whole* collected over a 7-month period during 2009 [30]. We preprocess this data by first filtering out users that do not have at least 15 tweets in each month. Then for each tweet from these users, we perform standard tokenization, stemming, lemmatization, infrequent words removal, and get our final dataset. In addition to this large complete data, for fast performance comparison purposes (which requires multiple runs for cross-validations with different parameter settings), we generate three sample datasets. We randomly sample 2%, 5%, and 20% of the users based on their degrees in the social network. Finally, to analyze more specific events, we create an *Tweets-Iran* dataset by extracting tweets that contain keywords in the 2009 Iran Election (such as 'iran', 'iran elect', 'neda', etc.[1]).

Table 2: Datasets used.

Dataset	#Users	#Edges	#Tweets
Tweets-Whole	$46.5K$	$2.1M$	$27.5M$
Tweets-2%	$0.9K$	$28K$	$0.7M$
Tweets-5%	$2K$	$0.1M$	$1.8M$
Tweets-20%	$9.2K$	$0.7M$	$6.5M$
Tweets-Iran	$3K$	$40K$	$62K$

Baselines. To the best of our knowledge, there are no existing methods which model how social influence affects the generation of *both* posts and links as we do. As a result, unlike PoLIM which can predict both links and retweets, most state-of-the-art algorithms can only be used for either one of the task. We list all the baselines in the following.

1. *COmmunity Level Diffusion* (COLD) [14] is a state-of-the-art generative model that covers social influence, posts and links as we do. However, the social influence it models only contributes to the link generation, while all the posts are still considered generated from users' own interests.
2. *Mixed Membership Stochastic Blockmodel* (MMSB) [2] combines dense connectivity (blockmodel) with node specific variability

[1]Code can be found at: `https://goo.gl/zWMvFy`

Copyright © 2019 by SIAM
Unauthorized reproduction of this article is prohibited

(mixed-membership). Each user's membership is a mixture of communities with different weights.

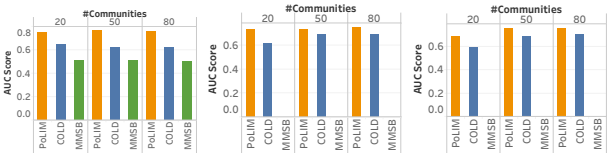
3. *Topical Affinity Propagation* (TAP) [26] is a popular model which finds influence between users given the network and users' topic interests. For our experiments, we feed TAP with the topic interests learned by PoLIM. We then combine topic posterior and the topical influence probabilities to calculate the probability of a tweet being influenced.
4. EMP is a baseline we designed for the retweet prediction task. It uses the empirical retweet ratio in the training data as its estimation of retweet probability in the testing data.

5.2 Q1: Link Prediction We show that the parameters learned from PoLIM can be used to predict whether a user will follow another user well. In this experiment, we design a 5-fold cross validation on *Tweets-2%*, *Tweets-5%*, and *Tweets-20%*: in each instance, we leave out 20% of the links as the test set. Further, we randomly choose 1% of the non-existing links and include them in the test set. We then train our model on the remaining links, and evaluate the link prediction performance on the test set. To calculate the probability of user n_a in c_i following user n_b in c_j, we use:

$$\Pr(n_a \to n_b) = v_a \rho A_i(n_b) + (1 - v_a) I_{ij} A_j(n_b) + v_a(1-\rho)\frac{1}{N} \tag{5.1}$$

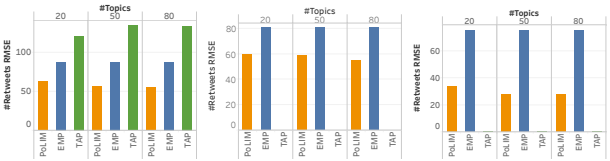
The first term represents the case where n_a behaves from self interests and chooses a followee from his/her own community c_i; the second term measures the probability that n_a is influenced by c_j, and chooses a celebrity in c_j to follow; and finally, the third term represents the probability of a random following. Given the $\Pr(n_a \to n_b)$ value calculated from Eq. 5.1, we can compare it with a discrimination threshold value between 0 and 1 to predict whether the link exists or not. Hence, we use the AUC (area under the curve) metric for evaluation, which calculates the performance (true positive rate divided by false positive rate) at various threshold settings, and the area under this performance curve.

As we can see in Fig. 1, in all three settings where we vary the number of communities from low to high while keeping the number of topics constant (20), PoLIM consistently outperforms the baseline algorithms. MMSB performs the worst among all three, and it does not finish (converge) even in a day for the larger *Tweets-5%* and *Tweets-20%*. Note that our method also outperforms COLD in all different settings. This is expected since PoLIM combines both community-level influence (I) and personalized influence (v), while COLD only contains the former. Hence PoLIM predicts the link generation with higher accuracy.



(a) *Tweets-2%* (b) *Tweets-5%* (c) *Tweets-20%*

Figure 1: Link prediction results. Higher the AUC, better the performance (MMSB does not finish for *Tweets-5%* and *Tweets-20%*).



(a) *Tweets-2%* (b) *Tweets-5%* (c) *Tweets-20%*

Figure 2: Retweet volume prediction results. Lower the RMSE, better the performance (TAP does not finish for *Tweets-5%* and *Tweets-20%*).

5.3 Q2: Retweet Volume Prediction In this section, we show that PoLIM captures well how social influence affects the tweet generation. We design a retweet volume prediction task, where we train our model on data in a training time period, and then use the model parameters to predict how many tweets in the testing time period are retweets. We use a 7-fold cross validation on *Tweets-2%*, *Tweets-5%* and *Tweets-20%*. Repeatedly we leave one month's tweets for testing, and train PoLIM on the rest six months' data. For each tweet (from user n_a in community c_i) in the testing data, we can calculate the probability of the tweet being generated from social influence using:

$$\begin{aligned}&\Pr(r=1|t) \propto \Pr(t|r=1) * \Pr(r=1)\\ &= (1-v_a)\sum_j I_{ij} \sum_z \theta_j(z) \prod_{w\in t}[u\phi_B(w) + (1-u)\phi_z(w)]\end{aligned} \tag{5.2}$$

where r is the switch value for the tweet ($r = 1$ means the tweet is cause by social influence). The latter part of the equation basically goes over all possible influencing communities and topics and calculate the likelihood of the tweet. Similarly we have:

$$\Pr(r=0|t) \propto v_a \sum_z \theta_i(z) \prod_{w\in t}[u\phi_B(w) + (1-u)\phi_z(w)] \tag{5.3}$$

We then normalize these probabilities to get the final influence probability for the tweet. If this probability is greater than 0.5, we predict it as a retweet. Finally, we calculate the average RMSE value between all users' predicted number of retweets and the ground truth number obtained from the RT labels. Note that the word 'RT' is only used to obtain the labels for tweets,

Copyright © 2019 by SIAM
Unauthorized reproduction of this article is prohibited

the word itself is filtered as stopwords from the text corpus for training and testing.

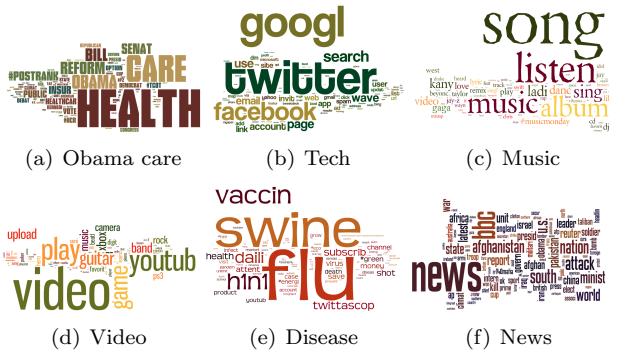
We see in Fig. 2, PoLIM achieves the best performance in all three datasets with different number of topics (number of communities fixed as 20). This is mainly because PoLIM directly models for each tweet if it is influenced. In contrast, TAP is designed to capture the overall topical influence and authority based on users' topic interests and their connections, and it turns out to perform badly when it is applied on a lower level for each individual tweet (it does not converge in a day for *Tweets-5%* and *Tweets-20%*). EMP also performs worse because it does not adapt to the text content of the testing tweets. COLD makes no distinction between retweets and tweets, so it can not predict the number of retweets.

5.4 Q3: Identifying Influenced Content We identify the portion of tweets and links in *Tweets-Whole* that are caused by social influence, and show that PoLIM indeed learns beyond the weak supervision (i.e. the retweet labels). After running PoLIM on *Tweets-Whole*, similarly we use Eq. 5.1, Eq. 5.2, Eq. 5.3 to calculate the influence probability of a tweet/link. We find that among a total of 27.5M tweets, there is a significant portion ($\sim 4.7\%$) of $\sim$ 1.3M tweets which are not retweets but are actually identified as being influenced; and among a total of 2.1M connections, there is a large portion (27.3%) of $\sim$ 574K links that are affected by social influence. This shows the impact of social influence on both the tweets and links generations. Further, when we run PoLIM on *Tweets-Iran*, these portions of influenced tweets and links increase to 54% and 50.7% respectively (as the communities are all about the same event - Iran election - they have higher influence among them).

5.5 Q4: Quality of Topics We show in Fig. 3 that PoLIM learns high quality topics on the entire Tweets dataset *Tweets-Whole*. We show six example topics, and there are many other topics covering different domains such as economy, Iran election, traffic, sports, design, religious, energy, education, etc. Note that all the words in the word clouds are stemmed and lemmatized.

5.6 Q5: Influence Analysis Here we examine the influence PoLIM learns at the community level on *Tweets-Whole*. It correctly learns the probability of a person being influenced as well, but we omit those results for space.

Community influence and celebrities. Broadly we find a very influential community c_6 and a set of



(a) Obama care (b) Tech (c) Music

(d) Video (e) Disease (f) News

Figure 3: Word clouds for topics learned by PoLIM on the entire tweet data *Tweets-Whole*. For each topic, we show the top 100 words with highest weights. Each word is already stemmed and lemmatized. The layout of the word cloud is randomized, and the size of the word is proportional to its weight in the topic.

communities (c_3 to c_8) with high *within*-community influence instead (I matrix not shown due to lack of space). We focus on these communities and plot their corresponding word clouds in Fig. 4. The size of each word is proportional to a weighted importance of the word calculated by using the topic distribution of the community (θ) and the word distribution for the topic (ϕ). In Fig. 4(a), we observe that many frequent words used in c_6, such as 'design', 'busi', 'market', 'facebook' come from different disciplines and topics. In fact, the topic distribution of c_6 has a 'flat' shape, showing that this community is interested in a wide range of topics, varying from economy, politics to food, animal, iphone. Combined with the fact that c_6 has influence over almost all other communities, this depicts the type of users who are influential in the social network, who respond to a wide range of topics and are more likely to be well-known figures/companies. See Fig 4(b)), where we show the celebrities (the users with top A_i values) in the community. The individuals with highest A_i values in c_6 are mashable (media and entertainment company), chrisbrogan (very highly rated influencer online), and problogger (a popular blog website). All of these celebrities in c_6 are famous online users with high influence over a wide range of topics, PoLIM correctly groups them as communities, and our influence matrix correctly captures their high influence over the other communities.

On the other hand, PoLIM also detects communities with very focused topic interests. Consider c_3, we see in Fig. 4(c)(d), c_3 shows a focused interest in Obama health care. Moreover, the 'local' celebrities in c_3, such as timeoreilly, politicalticker, whitehouse_rss, are consistent with this topic interests. We make similar observations for c_{10} in Fig. 4(e)(f), where the main

Copyright © 2019 by SIAM
Unauthorized reproduction of this article is prohibited

(a) Topic interests of c_6

(b) Topic interests of c_3

(c) Topic interests of c_{10}

User Screenname	A_6 Value
mashable	0.0092
chrisbrogan	0.0051
problogger	0.0045
nansen	0.0041
brooksbayne	0.0038
the_gman	0.0036
chrispirillo	0.0034
garyvee	0.0034
stephenkruiser	0.0033
dcagle	0.0032
tedmurphy	0.0032

(d) Celebrities in c_6

User Screenname	A_3 Value
timoreilly	0.1446
nprnews	0.0820
anamariecox	0.0818
bbctech	0.0236
markknoller	0.0209
gleonhard	0.0187
politicalticker	0.0185
harrislacewell	0.0149
howardlindzon	0.0138
joanwalsh	0.0129
whitehouse_rss	0.0126

(e) Celebrities in c_3

User Screenname	A_{10} Value
engadget	0.0662
journalismnews	0.0432
shanselman	0.0350
hatebu	0.0343
twfeed	0.0326
pvponline	0.0317
crackberry	0.0237
theiphoneblog	0.0214
whiteafrican	0.0160
watch_akiba	0.0156
shauninman	0.0142

(f) Celebrities in c_{10}

Figure 4: The topic distributions for several communities and the celebrities (users with the highest A_i values) in these communities. For the most important topics in each community, we annotate them with the top three words (stemmed and lemmatized) in those topics.

topic interests are related to technology (keywords like 'window', 'appl', 'iphone', 'app', etc.), and the celebrities in c_{10} (such as engadget, theiphoneblog, crackberry) also show similar technology focus.

Celebrity structures. We examine the celebrity values in each community and find different celebrity structures for different communities. By examining the histogram and entropy of the A values for the top 20 celebrities in the communities, we make an interesting observation. For a community about a specific event, such as c_7 which is mainly about the Iran election, the importance/authority are more spread out to multiple users; while for a community about a general topic, like c_{14} which focuses on garden and art, a few users would have the leading authority and the others are much less influential. This leads to an insight that when a new topic/event emerges, different perspectives/arguments of the subject can be discussed, which offers more chances for users to be noticed and hence become influential. On the other hand, for a very developed and general topic, the 'heat' of the discussion has decreased to a stable level, and the authority has started to concentrate rather than diverge.

5.7 Q6: Case Study on Iran election PoLIM can help understand and detect finer-grained communities even for a specific topic. In this experiment, we run PoLIM with 15 topics and 4 communities on *Tweets-Iran* which is a sub-dataset about the 2009 Iran Election (corresponding word clouds for the communities ommitted for space). We observe that each community corresponds to a specific aspect of the event. c_1 is about the results or progress of the election itself; c_2 is about the Iranian green movement, a political movement that arose after the election to demand the removal of Mahmoud Ahmadinajad from office; c_3 is related to the video of the death of Neda, a student of philosophy, during the protest; and finally c_4 is mainly about the online petition during the election. Hence PoLIM can also be applied on a specific event to discover communities with subtle difference.

6 Discussion and Conclusions

To summarize our observations, our novel model PoLIM uses weak supervision to jointly model links and posts in social media data. It efficiently learns real communities with meaningful topic interests, as well as the celebrities with high influence in each communities. Broadly, it extracts the social influence strength among different communities well. At the same time, at a finer level, it also correctly learns a person's tendency of being influenced.

Our experimental results also confirm the existence of social influence in real datasets, and its impact on various applications. We find that the same latent social influence governs a significant portion of posts and links in Twitter, and such a portion tends to increase when the dataset is about more specific events. By differentiating these posts and links from those caused by self interests, PoLIM is able to learn more precise topic interests, and therefore achieve significantly better performance in concrete quantitative tasks such as predicting future links and retweets.

It would be interesting to study the detectability of influence by our model as function of its strength.

Copyright © 2019 by SIAM
Unauthorized reproduction of this article is prohibited

Note that an important hypothesis we used is that retweets are more likely to be caused by social influence. This hypothesis then effectively guides the learning of social influence in PoLIM in a weakly supervised fashion. While retweets are natural and good indicators of influence in Twitter, they may not be available for other social media websites. In these cases, as future work, we may design other indicators such as the replies, mentions, or even train a low-cost low-accuracy feature-based classifier as the weak supervision for PoLIM. We can further consider incorporating word embeddings into our model to improve topic learning [29, 31].

Acknowledgements: This article is based on work supported by the NSF CAREER IIS-1750407, the NEH (HG-229283-15), ORNL and a Facebook faculty gift.

References

[1] L. M. Aiello, A. Barrat, R. Schifanella, C. Cattuto, B. Markines, and F. Menczer. Friendship prediction and homophily in social media. *TWEB*, 6(2), 2012.

[2] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *JMLR*, 9(Sep):1981–2014, 2008.

[3] A. Anagnostopoulos, R. Kumar, and M. Mahdian. Influence and correlation in social networks. In *KDD*. ACM, 2008.

[4] M. Andrews and G. Vigliocco. The Hidden Markov Topic Model: A Probabilistic Model of Semantic Representation. *Topics in Cog. Sc.*, 2(1):101–113, 2010.

[5] V. Belák, S. Lam, and C. Hayes. Cross-community influence in discussion fora. In *ICWSM*, 2012.

[6] B. Bi, Y. Tian, Y. Sismanis, A. Balmin, and J. Cho. Scalable topic-specific influence analysis on microblogs. In *WSDM*, pages 513–522. ACM, 2014.

[7] S. Blasiak and H. Rangwala. A Hidden Markov Model Variant for Sequence Classification. In *IJCAI*, 2011.

[8] D. Blei. Probabilistic Topic Models. *Communications of the ACM*, 55(4):77–84, 2012.

[9] L. Chen, K. S. M. T. Hossain, P. Butler, N. Ramakrishnan, and B. A. Prakash. Flu gone viral: Syndromic surveillance of flu on twitter using temporal topic models. In *ICDM*, pages 755–760, 2014.

[10] T. L. Griffiths and M. Steyvers. Finding scientific topics. *PNAS*, 101(suppl 1):5228–5235, 2004.

[11] A. Gruber, Y. Weiss, and M. Rosen-Zvi. Hidden topic markov models. In *AISTATS*, 2007.

[12] X. He, K. Xu, D. Kempe, and Y. Liu. Learning influence functions from incomplete observations. In *NIPS*, pages 2073–2081, 2016.

[13] L. Hong, D. Yin, J. Guo, and B. Davison. Tracking Trends: Incorporating Term Volume into Temporal Topic Models. In *KDD*, pages 484–492, 2011.

[14] Z. Hu, J. Yao, B. Cui, and E. Xing. Community level diffusion extraction. In *SIGMOD*, 2015.

[15] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *KDD*, pages 137–146. ACM, 2003.

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

[17] L. Liu, J. Tang, J. Han, M. Jiang, and S. Yang. Mining topic-level influence in heterogeneous networks. In *CIKM*, pages 199–208. ACM, 2010.

[18] P. V. Marsden and N. E. Friedkin. Network studies of social influence. *Sociological Methods & Research*, 22(1):127–151, 1993.

[19] Y. Mehmood, N. Barbieri, F. Bonchi, and A. Ukkonen. Csi: Community-level social influence analysis. In *ECML PKDD*, 2013.

[20] S. A. Myers, C. Zhu, and J. Leskovec. Information diffusion and external influence in networks. In *KDD*. ACM, 2012.

[21] R. M. Nallapati, A. Ahmed, E. P. Xing, and W. W. Cohen. Joint latent topic models for text and citations. In *KDD*, pages 542–550. ACM, 2008.

[22] A. Pal and S. Counts. Identifying topical authorities in microblogs. In *WSDM*, pages 45–54. ACM, 2011.

[23] M. J. Paul and M. Dredze. You are what you tweet: Analyzing twitter for public health. *ICWSM*, 2011.

[24] M. Qiu, F. Zhu, and J. Jiang. It is not just what we say, but how we say them: Lda-based behavior-topic model. In *SDM*. SIAM, 2013.

[25] Q. Qu, C. Chen, C. S. Jensen, and A. Skovsgaard. Space-time aware behavioral topic modeling for microblog posts. *IEEE Data Eng. Bull*, 38(2), 2015.

[26] J. Tang, J. Sun, C. Wang, and Z. Yang. Social influence analysis in large-scale networks. In *KDD*, pages 807–816. ACM, 2009.

[27] Y. Wang, E. Agichtein, and M. Benzi. Tm-lda: efficient online modeling of latent topic transitions in social media. In *KDD*, pages 123–131. ACM, 2012.

[28] J. Weng, E.-P. Lim, J. Jiang, and Q. He. Twitterrank: finding topic-sensitive influential twitterers. In *WSDM*, pages 261–270. ACM, 2010.

[29] G. Xun, V. Gopalakrishnan, F. Ma, Y. Li, J. Gao, and A. Zhang. Topic discovery for short texts using word embeddings. In *ICDM*, pages 1299–1304, Dec 2016.

[30] J. Yang and J. Leskovec. Patterns of temporal variation in online media. In *WSDM*, 2011.

[31] C. Zhang, L. Liu, D. Lei, Q. Yuan, H. Zhuang, T. Hanratty, and J. Han. Triovecevent: Embedding-based online local event detection in geo-tagged tweet streams. KDD '17, pages 595–604, 2017.

[32] J. Zhang, J. Tang, Y. Zhong, Y. Mo, J. Li, G. Song, W. Hall, and J. Sun. Structinf: Mining structural influence from social streams. In *AAAI*, 2017.

[33] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li. Comparing twitter and traditional media using topic models. In *Advances in Information Retrieval*, pages 338–349. Springer, 2011.

[34] Y. Zhu, X. Yan, L. Getoor, and C. Moore. Scalable text and link analysis with mixed-topic link models. KDD '13, pages 473–481. ACM, 2013.

Copyright © 2019 by SIAM
Unauthorized reproduction of this article is prohibited