Community Level Diffusion Extraction

Zhiting Hu[1,3], Junjie Yao[2], Bin Cui[1], Eric P. Xing[3]
[1]Key Lab of High Confidence Software Technologies (MOE), School of EECS, Peking University
[2]East China Normal University
[3]Language Technologies Institute, Carnegie Mellon University
{zhitingh,epxing}@cs.cmu.edu, junjiey@gmail.com, bin.cui@pku.edu.cn

ABSTRACT

How does online content propagate on social networks? Billions of users generate, consume, and spread tons of information every day. This unprecedented scale of dynamics becomes invaluable to reflect our zeitgeist. However, most present diffusion extraction works have only touched individual user level and cannot obtain comprehensive clues.

This paper introduces a new approach, i.e., COmmunity Level Diffusion (COLD), to uncover and explore temporal diffusion. We model topics and communities in a unified latent framework, and extract inter-community influence dynamics. With a well-designed multi-component model structure and a parallel inference implementation on GraphLab, the COLD method is expressive while remaining efficient.

The extracted community level patterns enable diffusion exploration from a new perspective. We leverage the compact yet robust representations to develop new prediction and analysis applications. Extensive experiments on large social datasets show significant improvement in prediction accuracy. We can also find communities play very different roles in diffusion processes depending on their interest. Our method guarantees high scalability with increasing data size.

Primary Classification:

H.4.0 [**Information Systems**]: General

Keywords:

Information Diffusion, Community Detection, Graph Model

1. INTRODUCTION

A longstanding question in communication media research is Lasswell's 5W maxim: "Who says What to Whom in What channel with What effect?" [**?**, **?**]. In the prevalent online social networks, such as Twitter, Facebook and Weibo, billions of users share messages and interact with others. User activities exhibit rich temporal dynamics. Understanding these dynamics can reveal unique insight into our society. As a first and critical step, what popular topics do users talk and how do they spread the topics?

SIGMOD'15, May 31 – June 04, 2015, Melbourne, VIC, Australia.

http://dx.doi.org/10.1145/2723372.2723737.

Recently, there have been extensive studies in this direction [**?**, **?**, **?**, **?**]. On one hand, most information diffusion extraction works focus on individual-level interactions and structural topologies [**?**, **?**, **?**], where the influence between individual users and the bridging nodes (structural hole) spanning network structures are used to model the diffusion process. On the other hand, temporal modeling methods are developed to capture aggregated temporal trends of online content [**?**, **?**].

These variety of diffusion extraction methods have enjoyed impressive success but still have large drawbacks. First, the structural methods largely ignore topical differences and cannot capture the rich diversity of information patterns. Moreover, the highly volatile user behaviors usually render it difficult to accurately uncover diffusion patterns for *individual level* approaches. Finally, the aggregation methods fail to reveal detailed dissemination processes.

Can we unify these different lines, and obtain a rich spectrum of online temporal diffusion? This paper offers a new perspective. We propose to extract *community level* diffusion, i.e., modeling diffusion patterns of topics across different communities. Community is a collection of users with more intense interactions amongst its members than the rest of the global network [**?**]. It provides the basis for user engagement in social networks. Meanwhile, the "Strength of Weak Ties" theory[1] [**?**] has suggested a critical role of inter-community interactions in online diffusion. In this work, we extract communities' temporal dynamics as well as influence between communities, and provide a coarse-grained diffusion representation. The compact community level extraction captures the backbone of information spreading, improves temporal modeling, and finally better predicts and analyzes the diffusion.

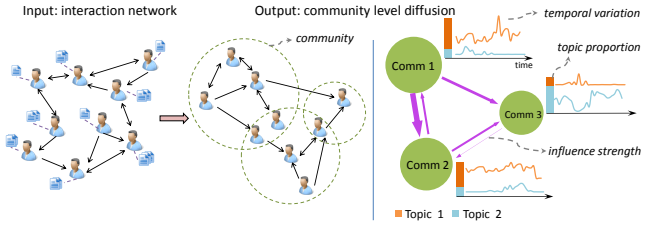


Figure 1: Overview of Community Level Diffusion.

Figure 1 shows an overview of the community level diffusion problem. The input is an interaction network among users, along with user-generated content over time. Our goal

[1]Weak social ties are responsible for the majority of the information spreading through human networks.

is to uncover the information dynamics across communities, including the communities' varying interest in different topics (topic proportion),

Community level diffusion extraction is challenging. Communities and topics are both hidden. Pipeline approaches to extract these two factors in sequence fail to capture their interdependence. Though in recent years an array of techniques [**?**, **?**, **?**] have been developed for jointly leveraging these two critical factors, they all fall short of suitably modeling the correlation between them. Besides the task of simultaneously extracting community and topics, we are even required to include vibrant temporal factors.

To tackle the difficulties, this paper develops a latent model, COLD (COmmunity Level Diffusion), to extract communities, topics, and community level topic dynamics in a unified way. We model community and topic as latent variables, and set up a generative process for observed network, text, and time to accurately characterize the topic diffusion at community level. An efficient sampling-based inference algorithm and its parallel implementation are designed.

Based on the extraction, we design an effective diffusion prediction method. Extensive experiments on large datasets show our approach greatly improves the prediction accuracy. We further use COLD to investigate rich social dynamics from the new community level perspective, and reveal meaningful diffusion patterns, such as the interest shift of communities and the time lag of topic spreading. Finally, we demonstrate how the new representation and improved prediction can lead to novel real-world applications such as influential community identification for viral marketing.

Our latent model brings up several innovations. We model users in overlapping communities with various affiliation degrees to capture users' personalities. Each community is associated with a mixture of topics, indicating its diverse interest with varying levels. This improves over previous one-to-one topical community limitations. We also associate each topic with community-specific temporal distributions, which is able to distinguish different dynamic processes within different communities.

To summarize, we make the following contributions:

1. **Novel Perspective**. We identify the problem of community level diffusion. It brings up new insights into the information dissemination process. To the best of our knowledge, such a new angle has not been studied before.
2. **Comprehensive Model**. We propose a latent model to uncover the hidden topics and communities as well as capture the community-specific temporal diffusion. It exhibits improved capacity to model information dynamics.
3. **Scalable Inference**. We decouple the model into several components, based on which an efficient inference algorithm is developed. We further set up a scalable parallel implementation.
4. **Inspiring Prediction & Exploration**. An effective diffusion prediction approach is developed which leverages community level patterns and shows significant superiority. We also study the real-world applications of COLD in viral marketing.

The rest of this paper is organized as follows: §2 reviews related literature. §3 formulates the problem and introduces the model. §4 describes the inference method and parallel implementation. §5 illustrates the model analysis and develops a new prediction approach. §6 evaluates the solution. And finally, we conclude this work in §7.

2. RELATED WORK

Information Diffusion. Online information diffusion has received increasing interests over the recent years [**?**]. One fundamental problem is to estimate the influence strength (or diffusion probabilities) between users [**?**, **?**]. However, the large volume of existing works have only focused on individual level, i.e., extracting user-to-user influence directly from the volatile individual behaviors. This can be limited and fragile to noises. In contrast, our approach proposes to model diffusion at community level, a new granularity that provides compact yet robust representations.

The extracted influence strength can be used in downstream applications such as future propagation prediction [**?**, **?**] and influential spreader identification [**?**, **?**, **?**], and eventually promote viral marketing and social network management. We develop an effective prediction method based on the new community level patterns and improve the previous work significantly.

Temporal Modeling. Another line of research captures temporal trends of content [**?**, **?**, **?**]. Topics Over Time (TOT) [**?**] is a latent generative model over the text and time stamps of documents. These works only reveal the aggregated topical trends, while ignoring the diversity of different users' temporal behaviors. By comparison, our work distinguishes temporal dynamic patterns across different communities, and provides a more thorough and versatile view.

Community Detection. Communities are natural groups formed by users with close connections and similar interest [**?**]. A mixed membership stochastic block model is introduced in [**?**] where each user has a probability distribution over communities. A growing number of recent works [**?**, **?**] incorporate both the network structure and content to improve community detection performance, e.g., Link-PLSA-LDA [**?**], RTM [**?**], and PMTLM [**?**]. In these models, content and links are both generated by the same latent variables. Thus communities are limited to have one-to-one correspondence with topics. Our model decomposes these two factors, which opens up an array of meaningful and desired extraction such as community interest over varying topics and popularity variation of topics across different communities.

3. COMMUNITY LEVEL DIFFUSION EXTRACTION METHOD

The central task is to extract community level diffusion patterns from social records (§3, §4), and utilize them to promote novel diffusion prediction and analysis (§5).

In this section, we first formulate the diffusion modeling problem from the new community level perspective. We then propose COLD (COmmunity Level Diffusion), a comprehensive latent variable model, to address the problem.

3.1 Problem Formulation

The notations used in this paper are listed in Table 1.

Definition 1. **(Interaction Network)**. Consider a social network $\mathcal{G} = (\mathcal{U}, \mathcal{E})$. $\mathcal{U}$ is a set of U users. The link set $\mathcal{E}$ denotes interactions between users and can be derived from various types of user interactions such as following, retweeting and commenting. A directed link $(i, i') \in \mathcal{E}$ represents there exists communication from user i to i', e.g., i' once retweeted i.

Symbol	Description
U, T, C, K	number of users, time slices, communities, and topics
D_i, E_i	number of posts by user i, and links from user i
d_{ij}	the jth post by user i
t_{ij}	the time stamp of post d_{ij}
w_{ijl}	the lth word in post d_{ij}
c_{ij}	community associated with post d_{ij}
z_{ij}	topic associated with post d_{ij}
$e_{ii'}$	indicator of the existence of link (i, i')
$s_{ii'}, s'_{ii'}$	communities associated with user i and i' in link (i, i')
$\boldsymbol{\pi}_i$	multinomial distribution over communities specific to user i
$\boldsymbol{\theta}_c$	multinomial distribution over topics specific to community c
$\boldsymbol{\phi}_k$	multinomial distribution over words specific to topic k
$\boldsymbol{\psi}_{kc}$	multinomial distribution over time specific to topic k and community c
$\eta_{cc'}$	general influence strength (diffusion probability) from community c to c'
$\zeta_{kcc'}$	influence strength on topic k from community c to c'
λ_0, λ_1	Beta priors on $\boldsymbol{\eta}$

Table 1: Notations Used in This Paper.

Each user $i \in \mathcal{U}$ is associated with a set of D_i posts, denoted as $\mathcal{D}_i$, where each post d_{ij} contains a bag of words from a given vocabulary, along with a posting time stamp t_{ij}. We use $\mathcal{E}_i$ to denote the set of links from i to other users, and define $E_i = |\mathcal{E}_i|$.

Community is a collection of users with more intense interactions amongst its members than the rest of the global network. It can be characterized not only by interaction link structures, but also the content (i.e. posts) generated by its members. While existing works on community modeling generally assume one community corresponds to one interest/topic, we associate each community with a topic distribution representing its different topical interests and give the following new definition.

Definition 2. **(Community)**. A community $c \in \{1, \ldots, C\}$ has two components: a multinomial distribution over topics $\boldsymbol{\theta}_c$, where each component θ_{ck} represents the probability that a post from the community is related with the corresponding topic k; and a probability vector $\boldsymbol{\eta}_c$ where each component $\eta_{cc'}$ is the mean of a Bernoulli distribution representing the diffusion probability from a user in community c to a user in community c'.

In social networks, users usually bear multiple roles and are influenced by different community contexts [**?**]. We therefore employ the *mixed-membership* approach: each user i is associated with a multinomial distribution over communities $\boldsymbol{\pi}_i$, where π_{ic} indicates her affiliation degree to community c.

Definition 3. **(Topic)**. A topic $k \in \{1, \ldots, K\}$ is a multinomial distribution over the vocabulary, denoted as $\boldsymbol{\phi}_k$.

A topic has changing popularity over time. We represent it as a temporal distribution. Besides, a topic can exhibit very different temporal dynamics within different communities, which leads to the following definition:

Definition 4. **(Community-level Temporal Variation)**. At community level, a temporal topic k is associated with a set of C multinomial distributions $\boldsymbol{\psi}_k = \{\boldsymbol{\psi}_{k1}, \boldsymbol{\psi}_{k2}, \ldots, \boldsymbol{\psi}_{kC}\}$, each of which represents the changing popularity of topic k within the corresponding community, i.e., a multinomial distribution over time slices.

Different communities are expected to play different roles in topic dissemination process. This can be modeled by influence strength between communities:

Definition 5. **(Community-level Influence Strength)**. For each topic k, the community-level influence strength is represented as the diffusion probabilities between any two communities c and c', denoted as $\zeta_{kcc'}$.

Note that we aim to model *topic-sensitive* influence between communities. The topics and communities are both latent factors to be extracted, and we are also required to uncover their correlations over time, i.e., community-level temporal variations and influence strength on topics.

The perspective of community level diffusion is quite different from prior works which study diffusion at individual level and largely ignore the effects of community structure. The new granularity provides a compact yet accurate representation, but also brings about unique complexity in modeling and computing. Next we develop our new model by describing its general structure as well as three properly decoupled components.

3.2 Model Structure

COLD is a generative model jointly over text, time and network. It uncovers latent communities and topics, and infers community-level dynamics in a unified way.

Although some of its building blocks are inspired by recent successful attempts, including the Mixed Membership Stochastic Blockmodel (MMSB) [**?**] over networks, and Topics over Time (TOT) [**?**] over text and time, COLD significantly goes beyond those on more comprehensive input features and powerful modeling ability. Compared to previous joint text and link models, COLD better fits the social network setting by accurate community interest recovering and refined user post treatment.

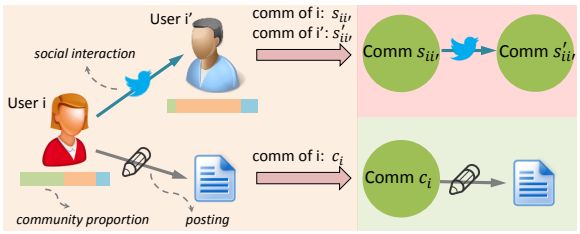


Figure 2: An illustration Showing the User Behaviors Explained by Community-specific Context. The color of each component corresponds to that of Figure 3.

COLD aims to model two basic types of user behaviors, i.e., posting (which generates text) and social interaction (which forms links), together. Figure 2 illustrates the user behavior modeling. Specifically, each user may assume different community memberships when taking these behaviors, and each behavior is further explained by *community-specific context*. That is, for the behavior of posting, the words and time stamp of the post are assumed to be generated by community-specific mixture of topics (i.e. $\boldsymbol{\theta}_c$); while for the behavior of social interaction, the link is governed by the community-specific interaction strengths with other communities (i.e. $\boldsymbol{\eta}_c$).

Figure 3 shows the graphical structure of COLD. By jointly considering the two types of user behaviors with properly separated generative process, COLD naturally combines content and network data while still keeping the model tractable. Both community interest over topics and topic dynamics within communities can be inferred from the timestamped posts specific to particular communities.

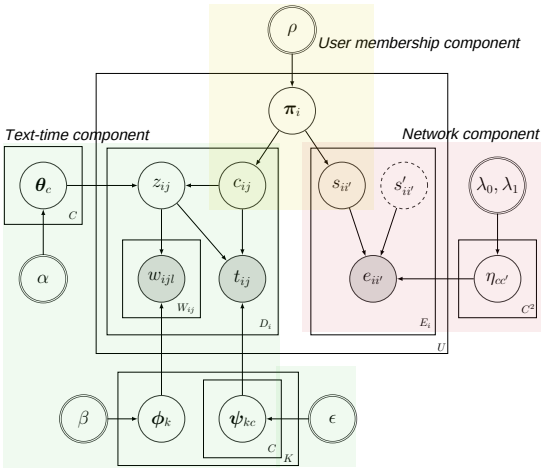


Figure 3: Graphical Model Representation of COLD. The latent variable $s'_{ii'}$ is represented as dashed circle since it is drawn from $\boldsymbol{\pi}_{i'}$ which is not shown in the graphical model.

3.3 Individual Components

Figure 3 shows three components together form COLD: the *text-time component* uncovers the semantic topics, and captures topic temporal variations; the *network component* accounts for the link structure; and the *user membership component* models user membership to communities, which also serves to seamlessly unify the other two components.

User membership component. Users in social network usually have multiple community memberships. We associated each user i with a community probability vector $\boldsymbol{\pi}_i$. Each post $d_{ij} \in \mathcal{D}_i$ is assigned to a community c_{ij}, denoting the community membership of user i when she writes the post. In addition, each link $e_{ii'} \in \mathcal{E}_i$ is associated with two communities $s_{ii'}$ and $s'_{ii'}$, one for each of the two users i and i' respectively, denoting their community memberships when user i builds relationship with i'.

The community membership vector $\boldsymbol{\pi}$ captures the personality of individual user, which can be utilized for predicting message propagation between users (§5).

Text-time component. Each post $d_{ij} \in \mathcal{D}_i$ contains a bag of words $\{w_{ij1}, \ldots, w_{ij|d_{ij}|}\}$ where $|d_{ij}|$ denotes the length of the post. In traditional topic models such as latent Dirichlet allocation (LDA) [?], a document is associated with a mixture of topics and each word has a topic label. This is reasonable for long documents such as academic papers. However, on social media like micro-blog, a post is usually short, and thus is most likely to be about a single topic [?]. We therefore associate with d_{ij} a single latent topic variable z_{ij} drawn from $\boldsymbol{\theta}_{c_{ij}}$ to indicate its topic. The words are then sampled from the corresponding word distribution $\boldsymbol{\phi}_{z_{ij}}$.

To model the temporal information of posts, we first discretize the time by dividing the entire time span of all users' posts into T time slices, then use a multinomial distribution $\boldsymbol{\psi}_{kc}$ over time slices to model the temporal variation specific to each topic k and each community c. Thus, a post d_{ij} is generated at the time t_{ij} drawn from $\boldsymbol{\psi}_{z_{ij}c_{ij}}$. Moreover, compared to Topics over Time (TOT) [?] which uses a Beta distribution to model time variations and only allows a unimodal distribution over time for each topic, our use of multinomial distribution can capture multimodal variations. It is more flexible and expressive in capturing real-life topics which usually rise and fall for many times.

Network component. We use pairwise community Bernoulli distributions $\boldsymbol{\eta}$ to model the presence and absence of links between pairs of users. For each link (i, i'), a boolean indicator $e_{ii'}$ is drawn from $\eta_{s_{ii'}s'_{ii'}}$ which represents the relationship strength between community $s_{ii'}$ and $s'_{ii'}$.

Social network is typically sparse, thus we only model positive links: the variables $s_{ii'}, s'_{ii'}$ exist if and only if $(i, i') \in \mathcal{E}_i$. As in [?], the negative links $(i, i') \notin \mathcal{E}_i$ are implicitly modeled in a Bayesian fashion: we use a Beta(λ_0, λ_1) prior on each $\eta_{cc'}$, and set $\lambda_0 = \kappa \cdot \ln(n_{neg}/C^2)$ and $\lambda_1 = 0.1$, where $n_{neg} = U(U-1) - \sum_i |\mathcal{E}_i|$ is the number of negative links and κ is a tunable weight. In this way, we reduce large amount of computation and achieve linear complexity on network modeling, as explained later in §4.2.

3.4 Generative Process

The generative process is summarized in Alg 1. Consider a user i who posts and interacts with others. When she publishes a post d_{ij}, she first selects the community membership c_{ij} by her community distribution $\boldsymbol{\pi}_i$, then selects a topic by the community's topic distribution $\boldsymbol{\theta}_{c_{ij}}$. With the chosen topic, words are generated from the topic's word distribution, and time stamp from the temporal distribution of that community and topic. On the other hand, when she interacts another user i', a community is sampled for each of them by their own community distributions, and the link is formed by community-community influence strength.

Algorithm 1 Generative Process for COLD

1. For each topic $k = 1, 2, \ldots, K$,
   (a) Sample the distribution over words, $\boldsymbol{\phi}_k | \beta \sim \text{Dir}(\beta)$.
   (b) For each community $c = 1, 2, \ldots, C$,
      i. Sample the distribution over time stamps, $\boldsymbol{\psi}_{kc} | \epsilon \sim \text{Dir}(\epsilon)$.
2. For each community $c = 1, 2, \ldots, C$,
   (a) Sample the distribution over topics, $\boldsymbol{\theta}_c | \alpha \sim \text{Dir}(\alpha)$.
   (b) For each community $c' = 1, 2, \ldots, C$,
      i. Sample community-community link probability, $\eta_{cc'} | \lambda_0, \lambda_1 \sim \text{Beta}(\lambda_0, \lambda_1)$.
3. For each user $i = 1, 2, \ldots, U$
   (a) Sample the distribution over communities, $\boldsymbol{\pi}_i | \rho \sim \text{Dir}(\rho)$.
   (b) For each post $j = 1, 2, \ldots$,
      i. Sample community indicator, $c_{ij} | \boldsymbol{\pi}_i \sim \text{Mul}(\boldsymbol{\pi}_i)$.
      ii. Sample topic indicator, $z_{ij} | \boldsymbol{\theta}_{c_{ij}} \sim \text{Mul}(\boldsymbol{\theta}_{c_{ij}})$.
      iii. For each word $l = 1, 2, \ldots$,
         A. Sample word, $w_{ijl} | \boldsymbol{\phi}_{z_{ij}} \sim \text{Mul}(\boldsymbol{\phi}_{z_{ij}})$.
      iv. Sample time stamp, $t_{ij} | \boldsymbol{\psi}_{z_{ij}c_{ij}} \sim \text{Mul}(\boldsymbol{\psi}_{z_{ij}c_{ij}})$.
   (c) For each link $(i, i') \in \mathcal{E}_i$,
      i. Sample community indicator, $s_{ii'} | \boldsymbol{\pi}_i \sim \text{Mul}(\boldsymbol{\pi}_i)$.
      ii. Sample community indicator, $s'_{ii'} | \boldsymbol{\pi}_{i'} \sim \text{Mul}(\boldsymbol{\pi}_{i'})$.
      iii. Sample link, $e_{ii'} | \boldsymbol{\eta}_{s_{ii'}s'_{ii'}} \sim \text{Ber}(\boldsymbol{\eta}_{s_{ii'}s'_{ii'}})$.

3.5 Discussion

COLD is designed to reveal the rich spectrum of online temporal diffusion across communities. Such a highly expressive model had not been explored due to its inherent complexity in modeling and computing. Here we discuss the modeling choices for handling the complexity.

Previous works on text and link modeling [**?**, **?**, **?**, **?**] have assumed one-to-one correspondence between communities and topics. Such modeling is limited since the rich correlation between communities and topics are ignored. We decouple these two factors, which not only improves both community and topic extraction (§6), but also opens up an array of meaningful and desired extraction. For instance, we can explore communities' varying topical interest by associating each community with a mixture of topics; we are also enabled to distinguish topic's temporal variation across communities, which, compared to previous aggregation methods [**?**, **?**], leads to new insights in user attention shift (§5.3).

However, the decomposition also leads to $C \cdot C \cdot K$ parameters in the *topic-sensitive* community-level diffusion ($\zeta_{kcc'}$), which can be prohibitive for inference. We therefore employ a *two-stage* approach by first inferring the general inter-community influence $\eta_{cc'}$ and the community interest θ_{ck}, then deriving $\zeta_{kcc'}$ as combination of these intermediate parameters (§5.1). The formulation effectively reduces the complexity to $C \cdot (C+K)$, and still exhibits strong predictive power in diffusion prediction (§5.2).

It is also worth noting that previous text-link methods are typically proposed for the document citation network, which, in the social network setting, requires to view each user's post collection as a huge document. A latent topic is then selected for each word of the "document". In contrast, COLD models individual posts separately, which helps to preserve the correlation among words in each post—we can associate a single topic to each post as a whole. This has not only overcome the heavy noise in social network-style text, but also decreased the inference complexity.

Finally, as discussed above, the carefully-designed Bayesian prior avoids explicit modeling of the negative links. Hence the computation for network modeling is drastically reduced given that social networks are typically very sparse.

4. INFERENCE & IMPLEMENTATION

This section develops an efficient inference method for COLD. We first present the basic inference algorithm using a sampling method. It scales linearly w.r.t. the data size, which is still intolerant for growing large social data. To ensure its scalability, we further design a parallel implementation based on GraphLab.

4.1 Approximate Inference

Exact inference for COLD model is difficult due to the intractable normalizing constant of the posterior distribution (see the appendix for more details), we therefore exploit collapsed Gibbs sampling [**?**] for approximate inference. As a widely used *Markov chain Monte Carlo* (MCMC) algorithm, Gibbs Sampling iteratively samples latent variables (i.e. $\{\boldsymbol{c}, \boldsymbol{s}, \boldsymbol{z}\}$ in COLD) from a Markov chain, whose stationary distribution is the posterior. The samples can therefore be used to estimate the distributions of interest (i.e. $\{\boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\phi}, \boldsymbol{\psi}\}$).

At each iteration of our Gibbs sampler, for each post d_{ij} by user i, we sample both the corresponding community indicator c_{ij} and the topic indicator z_{ij}; for each link (i, i'), we sample the corresponding community indicators $s_{ii'}$ and $s'_{ii'}$. Here we directly give the sampling formulas, and provide the detailed derivation in the appendix (Appx A).

Sampling community indicator c_{ij} for post d_{ij} according to,

$$P(c_{ij} = c | z_{ij} = k, t_{ij} = t, \boldsymbol{c}_{-ij}, \boldsymbol{s}, \boldsymbol{z}_{-ij}, \boldsymbol{t}_{-ij}, .) \propto \frac{n_i^{(c)} + \rho}{n_i^{(\cdot)} + C\rho} \cdot \frac{n_c^{(k)} + \alpha}{n_c^{(\cdot)} + K\alpha} \cdot \frac{n_{ck}^{(t)} + \epsilon}{n_{ck}^{(\cdot)} + T\epsilon}, \quad (1)$$

where $n_i^{(c)}$ denotes the number of posts and links of user i assigned to community c; $n_c^{(k)}$ is the number of posts assigned to community c and generated by topic k; $n_{ck}^{(t)}$ denotes the number of times that time stamp t is generated by community c and topic k. Marginal counts are represented with dots; e.g., $n_{ck}^{(\cdot)}$ denotes the total number of time stamps generated by community c and topic k. All the counters are calculated with the post d_{ij} excluded.

Sampling community indicators $s_{ii'}$ and $s'_{ii'}$ for link (i, i'). Recall that we only model $s_{ii'}$ and $s'_{ii'}$ for positive links $e_{ii'} = 1$:

$$P(s_{ii'} = c, s'_{ii'} = c' | e_{ii'} = 1, \boldsymbol{s}_{-ii'}, \boldsymbol{c}, \boldsymbol{e}, .) \propto \frac{n_i^{(c)} + \rho}{n_i^{(\cdot)} + C\rho} \cdot \frac{n_{i'}^{(c')} + \rho}{n_{i'}^{(\cdot)} + C\rho} \cdot \frac{n_{cc'} + \lambda_1}{n_{cc'} + \lambda_0 + \lambda_1}, \quad (2)$$

where $n_{cc'}$ is the number of positive links, with (i, i') excluded, whose communities indicators are $\{c, c'\}$.

Sampling topic indicator z_{ij} for post d_{ij} according to:

$$P(z_{ij} = k | c_{ij} = c, t_{ij} = t, \boldsymbol{z}_{-ij}, \boldsymbol{w}, \boldsymbol{t}_{-ij}, .) \propto \frac{n_c^{(k)} + \alpha}{n_c^{(\cdot)} + K\alpha} \cdot \frac{n_{ck}^{(t)} + \epsilon}{n_{ck}^{(\cdot)} + T\epsilon} \cdot \frac{\prod_{v=1}^{V} \prod_{q=0}^{n_{ij}^{(v)}-1} (n_k^{(v)} + q + \beta)}{\prod_{q=0}^{n_{ij}^{(\cdot)}-1} (n_k^{(\cdot)} + q + V\beta)}, \quad (3)$$

where $n_{ij}^{(v)}$ is the number of times word v occurs in the post d_{ij}; $n_k^{(v)}$ denotes the number of times word v is assigned to topic k. Note that $n_k^{(v)}$ and $n_k^{(\cdot)}$ are calculated with the post d_{ij} excluded.

After a sufficient number of sampling iterations as described above, we obtain a set of samples. The unknown distributions can then be computed by integrating across the samples [**?**].

4.2 Time Complexity

We now analyze the time complexity of our inference algorithm. It is shown that the devised algorithm scales linearly in terms of the size of data, i.e., the number of words and positive links.

In each iteration, the communities of user posts are first sampled. Since all the counters (e.g. $n_i^{(c)}$) involved in Eq.(1) can be cached and updated in constant time for each c_{ij} being sampled, Eq.(1) can be calculated in constant time. Thus, sampling all $\boldsymbol{c}$ takes linear time w.r.t. the number of posts. Next, we sample community indicators $\boldsymbol{s}$ using Eq.(2). Since we have implicitly modeled negative links in Bayesian prior (i.e., the Beta prior for $\eta_{cc'}$), we only need to sample $s_{ii'}$ and $s'_{ii'}$ for positive links $e_{ii'} = 1$. Hence the

complexity is reduced from quadratic (w.r.t. the number of users) to linear (w.r.t. the number of links). It significantly saves computation cost due to the sparseness of networks. Finally, sampling all $\boldsymbol{z}$ by Eq.(3) is linear in the number of words.

Though linear computational complexity might not seem prohibitive, it limits the applicability of the model for growing data. We further give a parallel implementation of the inference algorithm to scale up our model to large social dataset.

4.3 Parallel Implementation

We implement a parallel COLD inference algorithm on the distributed GraphLab system [**?**]. GraphLab is a vertex-centric programming framework, expressing computational dependencies with a distributed graph. It has demonstrated superior performance over popular parallel systems, e.g., MapReduce and Spark, for many machine learning algorithms.

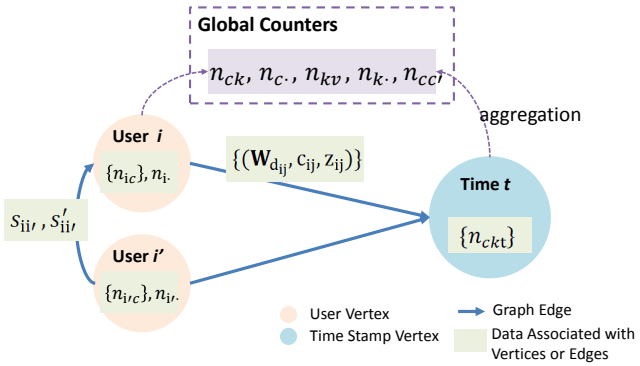


Figure 4: Abstraction of the Distributed Gibbs Sampler.

GraphLab implements the *gather-apply-scatter (GAS)* model which abstracts the program into three phases. In the *gather* phase, each vertex aggregates data from its neighborhood, and uses the result to update its own associated data in the *apply* phase. In the *scatter* phase, each vertex triggers neighboring vertex or modifies adjacent edge data.

The parallelization of our collapsed Gibbs Sampler is achieved by drawing new assignments in Eqs.(1–3) simultaneously. We first define the graph abstraction of our algorithm as shown in Figure 4. Specifically, we construct a bipartite graph connecting each user with each time stamp. An edge between a user i and a time stamp t contains the words of posts generated by user i at time t, as well as the community and topic indicators for the posts. We then incorporate the network data by connecting corresponding user vertices, where each edge contains the community indicators of two users.

The counters in Eqs.(1–3) are either maintained globally or in vertices. Global counters are aggregated from the vertices periodically, while counters in vertices are updated during the gather and apply phases. The program samples new assignments in the scatter phase. Alg 2 shows the GAS procedures of COLD Gibbs Sampler. We monitor the convergence of the algorithm by periodically computing the likelihood of training data [**?**].

The designed graph abstraction and GAS decomposition ensure most of the state maintenance to be processed locally, while global counters are generally only related to latent spaces which are low-dimensional. This maximizes the parallelism of the algorithm. Meanwhile, the data, as well as computation tasks, is partitioned into fine granularity and evenly distributed to each vertex and edge (Figure 4). Such an abstraction can promote better load balance among cluster nodes in the distributed setting.

Finally, as we avoid directly modeling the absence of links, a tremendous amount of communication is saved. Equipped with all the above features, our inference implementation leads to satisfying efficiency and scalability on large real data, as shown latter in our empirical study.

Algorithm 2 Vertex Program (GAS) of COLD Gibbs Sampler

Gather (v, e)
if $v.type = user$ **then**
  if $e.type = user\_time\_edge$ **then**
    return counts of each comm acc. to $e.\{c_{ij}\}$
  else if $e.type = user\_user\_edge$ **then**
    return counts of each comm acc. to $e.s_{ii'}$ or $e.s'_{ii'}$
  end if
else if $v.type = time$ **then**
  return counts of each comm-topic pair acc. to $e.\{z_{ij}, c_{ij}\}$
end if

Apply (v, *gather_result*)
if $v.type = user$ **then**
  update $v.n_{ic}$ by *gather_result*
else if $v.type = time$ **then**
  update $v.n_{ckt}$ by *gather_result*
end if

Scatter (v, e)
if $e.type = user\_time\_edge$ **then**
  for all doc j in e **do**
    sample $e.c_{ij}$ by Eq.(1) and $e.z_{ij}$ by Eq.(3)
  end for
else if $e.type = user\_user\_edge$ **then**
  sample $e.s_{ii'}$ and $e.s'_{ii'}$ by Eq.(2)
end if

5. DIFFUSION PREDICTION & ANALYSIS

Modeling information diffusion at community level can provide insights into social dynamics at a brand new granularity. This section first illustrates the compact diffusion patterns extracted by COLD, based on which novel prediction methods and diffusion analysis are designed.

5.1 Community Level Diffusion

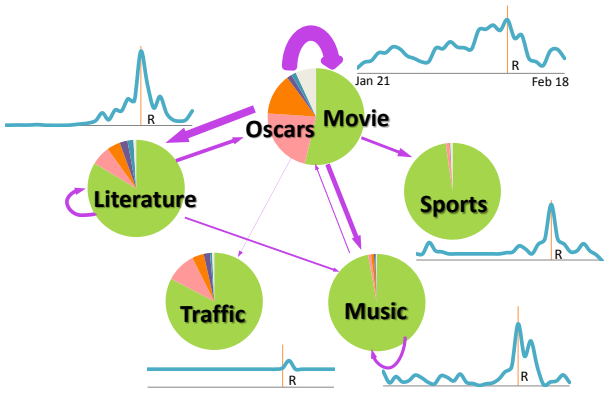
Here we demonstrate how our approach can be utilized to reveal topic dynamics and diffusion across communities.

In COLD, $\eta_{cc'}$ models the *general* influence strength of a community c on another community c', while θ_{ck} and $\theta_{c'k}$ captures the interest levels on topic k of community c and community c', respectively. As discussed in § 3.5, we can combine these intermediate factors to infer the *topic-specific* influence strength of c on c':

$$\zeta_{kcc'} = \theta_{ck}\theta_{c'k}\eta_{cc'}. \quad (4)$$

An example excerpted from our empirical study is shown in Figure 5, which demonstrates the extracted community level diffusion. The word cloud of topic *Movie The Journey to the West* and its diffusion path across different communities are included. Each community is represented as a *"pie chart"*-style node showing its top-5 interested topics (acc.

(a) Word Cloud



(b) Diffusion Path

Figure 5: Community-level Diffusion of Topic *Movie The Journey to the West*—a record-breaking box-office hit in China. We refer to the topic as *Journey West* for short.

to $\boldsymbol{\theta}$), with the dominant ones manually labeled by concise names. The time line near each community node shows the temporal variation (i.e. $\boldsymbol{\psi}$) of the topic specifically inside that community. The spike (time stamp R) of each line coincides with the release date of the movie. Interestingly, the spikes in communities *Literature*, *Sports* and *Music* are huge. In contrast, though community *Movie-Oscars* exhibited increasing activities at that time, the degree is not remarkable. We investigate the seemingly unintuitive phenomenon in §5.3.

The thickness of edges represents the influence strength between communities at this topic (i.e. $\boldsymbol{\zeta}$). We can clearly see that the community most interested in *Movie* and *Oscar* is the most influential one on topic *Journey West*. We formally discuss in §6.6 how COLD can be applied to identify most influential communities which is critical in viral marketing. The community *Traffic* is not active on the topic. By investigating the data we found this is because the majority of its members are traffic police official accounts.

The community level diffusion not only reveals compact yet meaningful overview of the topic diffusion process, but also forms the bases for the accurate diffusion prediction and in-depth analyses of social media dynamics, as shown next.

5.2 Prediction Method

A common task of diffusion analysis is to predict whether a message will propagate from one individual to another [**?**]. Taken retweeting as the example, given the content $\mathbf{w}_d$ of a message d, its publisher i, and another user i' (e.g. a follower of i), the objective is to estimate the probability that i' will retweet d from i.

Motivated by the above COLD model and the identified diffusion patterns, we develop our diffusion prediction method. Unlike traditional diffusion prediction methods which attempts to extract user-to-user diffusion probability directly from individual's interaction history, our solution conducts a two-step strategy: we first get the community-level diffusion probability as described in Eq.(4), and then combine personality of individuals through user-specific community memberships.

Our solution takes advantage of the community members' collective behavior patterns which are stable and predictable over time. User-specific community memberships can also be accurately captured based on both text and network features. In contrast, traditional methods can be ineffective due to the volatility of individual's actions and the sparsity of individual's records. We present the details of our method as follows.

Given a user-user-post triple (i, i', d), we estimate the probability that the post d will be spread from user i to user i'. Based on the topic modeling component in COLD, we first infer the underlying topics of the post through its text and its publisher's interest:

$$\begin{aligned} P(k|d,i) &\propto P(\mathbf{w}_d|k)P(k|i) \\ &\propto \prod_l \phi_{kw_{dl}} \cdot \sum_{c \in \mathbf{TopComm}(i)} \pi_{ic}\theta_{ck}, \end{aligned} \tag{5}$$

where $P(k|i)$ is the topical preferences of user i; $\mathbf{TopComm}(i)$ is the set of top communities of user i according to $\boldsymbol{\pi}_i$. Prior study has shown that a user on social media is typically active in a small number of communities [**?**], indicating that just the top few (e.g. 5 to 10) communities are sufficient enough to characterize the user's interest. We thus fix the size of **TopComm** as a small constant value (i.e. 5 in our setting).

Next, for one topic k, the influence of user i on user i' at topic k can be inferred by combining user community memberships and community-level influence strengths:

$$P(i, i'|k) = \sum_{\substack{c \in \mathbf{TopComm}(i) \\ c' \in \mathbf{TopComm}(i')}} \pi_{ic}\pi_{i'c'}\zeta_{kcc'}. \tag{6}$$

By combining Eq.(5) and (6), we obtain the final user-to-user diffusion probability of post d from user i to i':

$$P(i, i', d) = \sum_k P(k|d, i)P(i, i'|k). \tag{7}$$

Though seeming costly by combining several components together, we can actually get the result efficiently by offline filtering. In practice, we pre-collect the top communities of each user and then get her topic preferences. These are processed offline. In online determining of diffusion for user pairs, we simply collect the intermediate representation and calculate the final value, which only requires a weighted linear combination (Eq.(7)). The online computation complexity is $O(K|\mathbf{w}_d|)$, making the prediction very efficient, as further validated in Section 6.4.

Unlike most traditional diffusion prediction methods, the proposed approach also accounts for the semantics of spreading messages by comprehensive topic modeling as shown above, and is able to distinguish different diffusion processes across topics. The advantage and improved performance of COLD's prediction method will be illustrated in the experimental study.

5.3 Diffusion Patterns

Here we reveal several interesting and useful characters of diffusion based on the extracted community-level representations. We focus on the rich interplay between user interest and temporal dynamics.

Correlation between Community Interest and Topic Fluctuation. Users' concern usually changes over time, which is reflected by the popularity trends of topics. Here we study the connection between topics' temporal trends (i.e. ψ) and communities' topical focus (i.e. θ). Specifically, how do topics with varying interest levels fluctuate within communities? What is the difference between the variation patterns of topics with high weights and those with low?

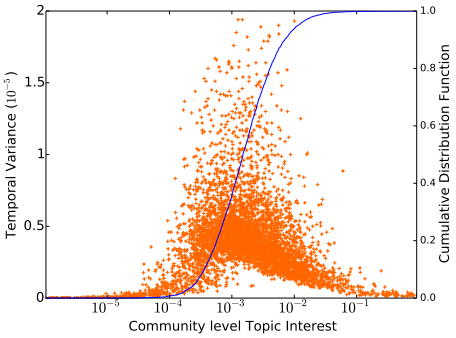


Figure 6: Topic Fluctuation and Community Interest. (X-axis is log scale.)

We use the *variance* of topic's community-specific temporal distribution ψ_{kc} to measure the fluctuation intensity of topic popularity. The scatter diagram in Figure 6 displays its relation with community's topic preference degree θ_{ck}. We also include the cumulative distribution function for the distribution of all communities' interest strengths. Topic popularity tends to fluctuate heavily throughout the time period within those communities with moderate interest in that topic. It generally exhibits higher variance when the topic has a proportion between 0.01% and 1% in community's interest distribution. In contrast, the popularity usually keeps steady in other communities which exhibit extremely low and high preference.

We can conclude that temporal topics are usually more dynamic in medium-interested communities. Members' attention on these topics tends to rise and fall intensely over time; in contrast, their engagement in communities' dominant topics is more stable.

Popularity lag between different communities. The influence strengths between communities have revealed the diffusion paths of topics. We then try to determine the time lag of topic propagating across communities. Specifically, how much time does it take for a topic diffusing from its initiators to others?

Figure 5, along with the above study, shows that different communities have varying temporal popularity on a specific topic. We can generally classify two categories of communities, i.e., communities with high preference for the topic and those with medium interest. Take the topic *Oscars2013* for example, the *Movie* and *Literature* communities (Figure 5) have the highest preference and others' are lower. We recognize the top 10 communities with largest probability on this topic as the highly-interested ones, yielding a set of communities whose average probability on the topic is 4.1%. Medium-interested communities are the remaining communities except those with extremely low probability (threshold is set to 0.01%), which have an average probability 0.37%. Due to the scale of social data, these thresholds are not low.

We then plot the *median topic dynamic curve* [**?**] for these two categories of communities. Specifically, for each community category, we align all of its community-specific temporal distributions of *Oscars2013* so that the peak popularity is equal to 1; we then at each time stamp plot the median probability over these aligned curves. Figure 7 shows the two categories' peak-time curves for topic *Oscars2013*.

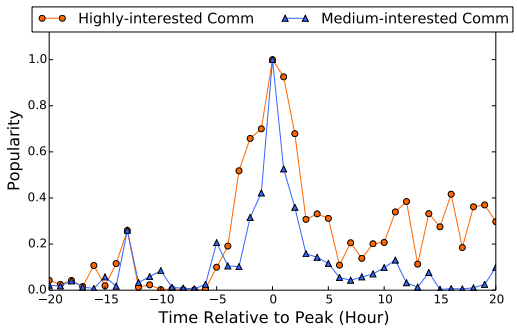


Figure 7: Time Lag between Highly and Medium-interested Communities on Topic *Oscars2013*.

We find that the topic popularity in the highly-interested communities rises earlier than the medium-interested communities. Moreover, the popularity in highly-interested communities lasts longer. It has a durable popularity. Investigating this time lag phenomenon on other topics, we also find similar results. Due to the space limit, we omit them.

The diffusion path and time lag confirm the usefulness of community level diffusion analysis and can be valuable in a variety of applications such as viral marketing, which we study in §6.6.

6. EMPIRICAL DEMONSTRATION

We conduct extensive experiments on large real datasets to evaluate the extraction and prediction performance of the proposed approach. The empirical study is divided into multiple stages. We first quantitatively evaluate the model's capability to extract topics and communities. We then test the diffusion prediction of the proposed approach, where the temporal modeling accuracy and the diffusion prediction performance are reported. Besides, efficiency and scalability of our model are validated by measuring the running time of model training, parallel extension, and online prediction. We also systematically study the sensitivity of model parameters. At last, we demonstrate novel real-world applications supported by our method.

All experiments are conducted on a cluster of Linux machines, each with eight 2.4GHz CPU cores and 48G memory. Data is stored in a RAID5 SAS storage server.

6.1 Setup

Datasets. Two real-world datasets are crawled from Sina Weibo[2], one of the most popular micro-blog platforms. We uniformly at random sample temporal messages from its streaming API. The messages are distributed nearly evenly

[2]http://weibo.com

in a three-month period from Dec 2012 through Feb 2013. Both datasets choose hour as the basic time interval.

Dataset 1. After removing stop words and low active users (with fewer than 20 posts), we create a dataset consisting of about 53K users, 11M posts and 91M words. The vocabulary size of these messages is 89K. Interaction network is derived from retweeting interactions between users, i.e., a link from user i to i' exists if i' once retweeted i's post. 2.7M links are observed.

Dataset 2. To evaluate the scalability of our approach, we also generate a larger dataset obtained in a similar manner as the first dataset. It consists of about 0.52M users, 10M links, 14M posts and 112M words.

Baselines. We compare the proposed COLD approach with several latest competitors. As there exists no prior approach capable of modeling communities and topic dynamics at the same time, we carefully select appropriate state-of-the-art methods to cover different aspects of diffusion modeling methods.

Table 2 lists the characters of these methods. Be aware that though some methods can be tuned to be used in several tasks, they are inadequate or uncompetitive compared with others; thus we omit them in this paper.

	features			tasks			
	text	social	time	topic ext	comm detec	temp modl	diff pred
PMTLM [?]	•	•		•	•		
MMSB [?]		•			•		
EUTB [?]	•	•	•	•		•	
Pipeline	•	•	•	•	•	•	
WTM [?]	•	•					•
TI [?]	•	•		•			•
COLD	•	•	•	•	•	•	•

Table 2: Feature and Task Comparison of Different Methods

Methods (1-2) models text and network features. Methods (3-5) aim at modeling temporal dynamics by integrating text and temporal features. At last, we include two diffusion prediction methods, which are used to compare the performance of direct individual diffusion modeling against community level modeling.

1. **Poisson Mixed-Topic Link Model (PMTLM).** PMTLM [?] defines a generative process for both text and links between users. Text generation follows the LDA [?] model, and links are modeled as a Poisson distribution. In PMTLM, links and text are generated by the same latent factor, which means one community is bounded to one topic (the latent factor is treated as community when generating links, and topic when generating text).

2. **Mixed Membership Stochastic Block model (MMSB)** [?] uses pairwise Bernoulli distributions to model links, and infers a distribution over communities for each user.

3. **Enhanced User-Temporal Model with Burst-weighted Smoothing (EUTB)** [?] assumes a topic is generated either by a user or a time stamp. It models the topic distributions for both users and time stamps. With network regularization and burst-weighted smoothing, EUTB performs best in time stamp prediction task among a set of competitors.

4. **COLD without Link (COLD-NoLink).** As a subpart of COLD without network component, COLD-NoLink can be used to test the contributions of network features.

5. **Pipelined Approach of Community-level Temporal Dynamics (Pipeline).** We first use MMSB model to assign each user to two most probable communities. Then we extract topic variation in particular communities by running TOT [?] on the posts generated by their members. This approach does not include interdependence between network and content.

6. **Whom to Mention (WTM)** [?] proposes a ranking method aimed at finding top users who would retweet a post and make contribution to its further diffusion. Features including user interest match, content-dependent user relationship and user influence are adopted.

7. **Topic-level Influence (TI)** [?] is a probabilistic generative model capturing influence between users at different topics. Both direct and indirect user influences are considered when predicting whether a user will retweet a friend's post.

6.2 Latent Factor Extraction

We report the model's capacity in extracting latent factors, showing its accurate modeling ability in both topic and community identification.

Topic Extraction. For an intuitive understanding of the extracted topics, we present four example word clouds in Figure 8. Meaningful subjects can be observed.

Figure 8: Word Clouds of Extracted Topics.

We next quantitatively evaluate COLD's topic extraction capacity by *perplexity* [?]. As a widely used metric in text modeling, perplexity measures how well a probability model predicts a sample. It can be interpreted as being proportional to the distance (formally, the cross-entropy) between the word distribution learned by the model and the actual distribution in test set [?]. Thus lower scores are better, indicating the model distribution is closer to the actual one. For a test set of M posts,

$$perplexity(\mathrm{D}_{test}) = \exp\Big\{-\frac{\sum_{d=1}^{M}\log p(\mathbf{w}_d)}{\sum_{d=1}^{M} N_d}\Big\},$$

where N_d is the length of the test post d, and $p(\mathbf{w}_d)$ is the probability of the words in the post; for COLD, it is computed as:

$$p(\mathbf{w}_d) = \sum_c \pi_{ic} \sum_k \theta_{ck} \prod_l \phi_{kw_{dl}},$$

where i is the post author.

We use a 5-fold cross validation testing, i.e., at each time interval, 80% of the posts as the train set, while the remaining 20% posts and all links as test set. Figure 9 shows the perplexity values under varying number of topics. It reveals

that COLD ($K = 100$, $C = 100$) has the lowest perplexity, indicating best topic discovery performance among all the competitors. Perplexity scores for EUTB and COLD are close, and both significantly outperform PMTLM. PMTLM's topics are tangled with communities in the same latent factor, weakening their fitness in modeling text.

In contrast, COLD comprehensively models communities and topics while explaining text well. Compared to EUTB, COLD not only models the dynamic content, but further captures social influence between users by incorporating network features. Therefore, the proposed method delievers improved advantages in topic extraction.

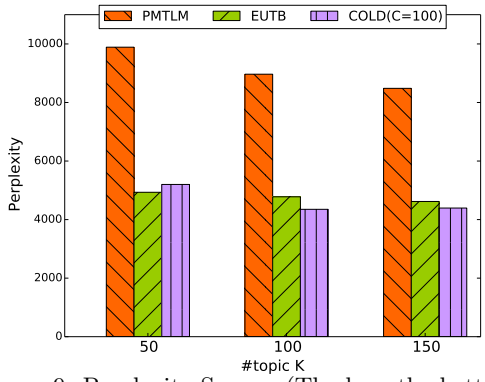


Figure 9: Perplexity Scores. (The less, the better.)

Community Detection. We then continue to evaluate the quality of the extracted communities. An example of the mixed-membership communities is visualized in Figure 16. We can see most users are clustered at the pentagon's corners and sides/diagonal lines, indicating that most of them have primary communities for engagement.

For quantitative evaluation, since no ground truth is available on Weibo network (which is the most common situation on social networks), we use link prediction, a widely-used quantitative measurement in the mixed-membership community setting without community labels [**?**].

Link prediction [**?**] is defined to estimate the probability of a link between two users. The probability of a link from user i to i' is measured by:

$$P_{i \to i'} = \sum_{s} \sum_{s'} \pi_{is} \pi_{i's'} \eta_{ss'}.$$

Since there is no pre-defined threshold for link existence, we turn to *area under the receiver operating characteristic curve* (AUC)[3] as the prediction accuracy. Given a rank of all non-observed links, the AUC value can be interpreted as the probability that a randomly chosen true positive link is ranked above a randomly chosen true negative link. In 5-fold cross validation, each time we use 20% of the positive links and randomly select 1% of the negative links to evaluate AUC; models are trained on the remaining links and all posts.

Figure 10 shows the AUC values for the three models. COLD ($K = 100$, $C = 100$) outperforms all other methods. Moreover, PMTLM and COLD are significantly more accurate than MMSB, showing that incorporating content feature benefits network structure modeling. COLD is also slightly better than PMTLM, with the comprehensive modeling ability.

[3]http://en.wikipedia.org/wiki/Roc_curve

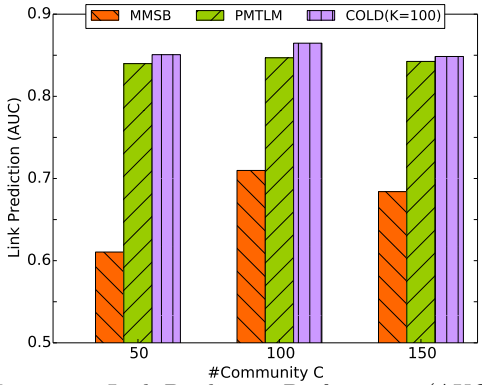


Figure 10: Link Prediction Performance (AUC).

The above two latent extraction experiments show the proposed COLD is versatile enough to simultaneously capture the critical characters in temporal diffusion. Though PMTLM performs almost as well as COLD in network modeling, the low topic extraction quality hinders its ability in modeling communities and topics at the same time. It reveals that, a single latent variable is not expressive enough to capture the diverse structures underlying the rich heterogeneous text and network context. COLD decouples these two critical factors and suitably models their correlations, ensuring both flexibility and accuracy.

6.3 Diffusion Prediction

Here we demonstrate the significant superiority of COLD over existing methods in terms of diffusion modeling and prediction.

Temporal Modeling. Time stamp prediction [**?**] is to estimate the occurring time stamp of a previously unseen document. Given the words in a post d and its author user i, we choose its time stamp as the candidate giving maximum likelihood:

$$\hat{t}_d = \arg\max_{t} \sum_{c} \pi_{ic} \sum_{k} \theta_{ck} \psi_{kct} \prod_{l} \phi_{kw_{dl}}.$$

Here 5-fold cross validation is used. The best results are obtained by setting $K = 100$ for the four models, and $C = 100$ for COLD and COLD-NoLink. Figure 11 shows the prediction accuracy for different models as a function of tolerance range, i.e., the maximum allowed difference between the real and predicted time stamps. COLD performs best among all competitors. COLD-NoLink outperforms EUTB, showing that the superiority of our approach derives not only from the the integration of network feature, but also from the fine-grained representation by distinguishing temporal topic dynamics across different communities.

It is notable that Pipeline, despite taking into account community specific topic dynamics, has poor performance. The reason is that it exploits network and content information separately, and ignores the interdependence between them. This experiment further justifies the advantage of the unified way which COLD uses to model dynamic social data.

Diffusion Prediction. The prediction task is to estimate whether a post d by a user i will be retweeted by another user i'. We use the averaged AUC values [**?**]. Specifically, given a tuple $\mathcal{RT}_{id} = (i, d, \mathcal{U}_{id}, \overline{\mathcal{U}}_{id})$, where $\mathcal{U}_{id}$ is the set of i's followers who retweeted d from i while $\overline{\mathcal{U}}_{id}$ is the rest of i's followers who ignored d, its AUC is computed by treating

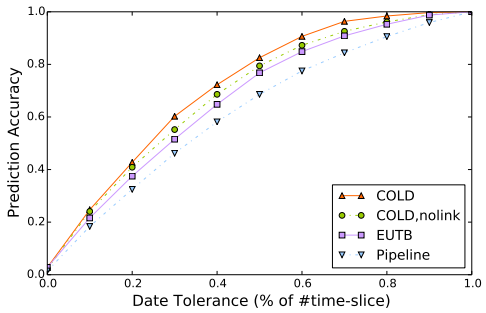


Figure 11: Time Stamp Prediction Accuracy.

the set $\{(i, i', d)|i' \in \mathcal{U}_{id}\}$ as positive examples and the set $\{(i, i', d)|i' \in \overline{\mathcal{U}}_{id}\}$ as negative examples. We then average the AUC values for all the tuples in test set. Here 5-fold cross validation is used by randomly holding out 20% tuples (with non-empty $\mathcal{U}_{id}$ and $\overline{\mathcal{U}}_{id}$) from the dataset as the test set each time.

Figure 12 gives the averaged AUC values for the three competitors. COLD ($K = 100$, $C = 100$) outperforms all other methods. Both TI and WTM attempt to model diffusion probability directly based on individual records, which can be prohibited by the volatility of individual behaviors as well as the sparsity of individual data. In contrast, COLD adopts a new paradigm by taking the advantage of the stability and predictability of collective behaviors of community members; user-specific characters are captured by user distribution over communities which can be accurately modeled by leveraging both text and network features.

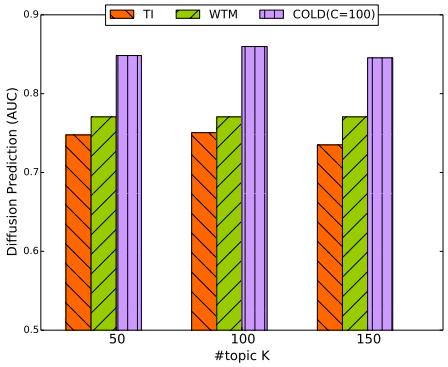


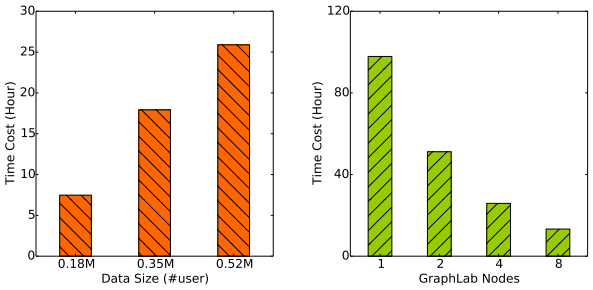
Figure 12: Diffusion Prediction Performance (AUC).

6.4 Efficiency

Parallel Scaling. We deploy our inference algorithm on the distributed GraphLab system to tackle the challenge of large data size. We study its scalability on the larger dataset with 0.52M users, 10M links and 14M posts. Training time is reported under different machine and data settings in Figure 13.

Figure 13(a) shows that the required time increases linearly as the data size grows. This confirms that by implicitly modeling negative links in Bayesian prior, our proposed collapsed Gibbs Sampler scales linearly with the size of dataset, as analyzed in §4.2.

The results in Figure 13(b) demonstrate satisfying efficiency of our distributed implementation on GraphLab. The running time decreases significantly with growing size of distributed GraphLab nodes. We reduce the training time for 10M links and 14M posts from hundred hours to just a few. This clearly shows the advantage of parallel processing. The model structure of COLD is loosely coupled enough to guarantee the parallel processing, showing advantage in growing social data size.



(a) Three Subsets, 4 Nodes. (b) Whole Dataset, # nodes.
Figure 13: Training Time of COLD on Parallel GraphLab.

Training time. Figure 14 shows the running time of different methods on the whole dataset. Note that while COLD jointly models text, network and time information, the baselines generally only account for a limited portion of the data. Though the basic implementation of COLD is costly, the parallel implementation guarantees the efficiency very well. It is feasible in actual deployment.

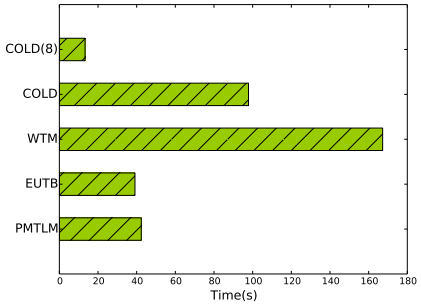


Figure 14: Training Time (C=K=100). "COLD (8)" is the distributed implementation on 8 nodes.

Prediction Time. Figure 15 presents the online diffusion prediction time of different methods after model training.

Our proposed method has lowest cost. In contrast, the baseline methods, i.e., TI and WTM are usually costly. This can be attributed to their lack of compact representation of user's profile. Specifically, TI's prediction is based on the influence of a user's multi-hop friends which can be a large set and require much processing time. For WTM, due to the absence of topic modeling, computing content-dependent features can be costly. Different from these methods, our proposed approach is able to characterize user's profile by extracted communities' representations, which can be efficiently computed by only a few operations.

6.5 Parameter Sensitivity

Though seeming complex, hyper-parameters for Bayesian priors, as discussed later, generally have negligible impact. COLD model is largely affected by two parameters, i.e., community number C and topic number K. We test the sensitivity in different tasks. The empirical results show that K primarily impacts topic modeling, while C has an effect

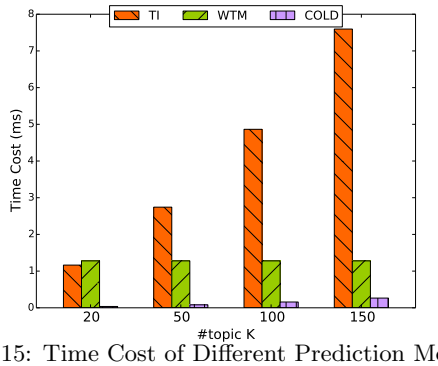


Figure 15: Time Cost of Different Prediction Methods.

mainly on community extraction. They together exert influence on diffusion prediction accuracy. The performance is stable under a broad range of parameter settings, indicating little tuning is required in actual deployment. Due to the space limitations, we defer the experimental results of parameter effects to the appendix (Appx B).

It is worth pointing out that Dirichlet hyper-parameters have low impact on model performance, and can be set as fixed values following the common strategy [?, ?] (i.e., $\rho = 50/C, \alpha = 50/K, \beta = \epsilon = 0.01$), while λ_0 and λ_1 are set as described in §3 for modeling negative links. Empirical studies also show our model is insensitive to these hyperparameters, which we omit due to space limitations.

6.6 Application of COLD

The improved diffusion prediction and the compact community level representation can not only help in traditional diffusion analysis, but also open up various novel applications at community scale. Here we demonstrate a concrete application to show how COLD can be applied to identify influential users and communities, which can be crucial in viral marketing and social network management [?].

Mining the influential nodes on social network has been studied extensively. Most existing works [?, ?, ?] usually assume pre-defined influence strength between users and focus on the information cascade simulation. Hence COLD is complementary, and can be directly applied, to these works by providing accurate influence strength estimation.

Moreover, COLD also enables us to go beyond the traditional user level and study the most *influential communities*. Selecting communities as the initial target has been increasingly employed due to its economy (e.g. by creating fanpages on Facebook) [?, ?] and effectiveness (e.g. by word-of-month among closely-connected members) [?]. Therefore, influential community identification enjoys practical values.

Analogous to measuring user's influence degree [?], we compute the influence degree of each community by setting the single community as the seedset and applying the well-known Independent Cascade [?] model on the extracted community level diffusion graph (e.g. Figure 5). Figure 16 shows the 4 most influential communities on topic *Sports*, as well as the aggregated other communities, by the 5 corners of a pentagon. Every user i is displayed as a point whose position is determined by her community membership π_i, i.e., a π_i-weighted convex combination of the 5 pentagon corners. For better understanding of the result, we also compute the influence degree of users, which is reflected by the size of the points in the figure. The characteristics of communities can then be obtained by analyzing their influential members. We can see that most of the influential users are from

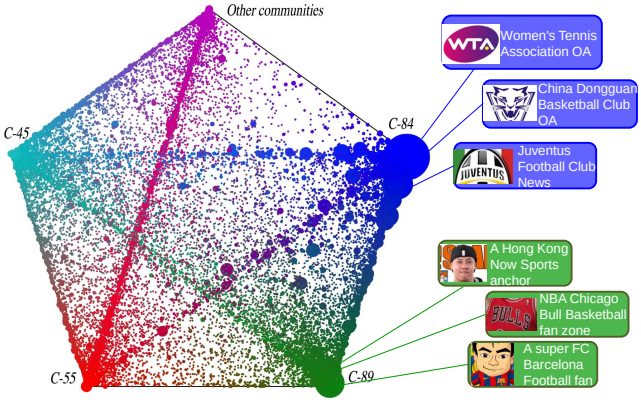


Figure 16: The Most Influential Communities on Topic *Sports*. Points at the pentagon's corners represent single-membership users, while points on the sides and diagonal lines represent users with mixed-membership in 2 communities. Only the top 20k influential users are shown (thus the points around "other communities" are few). Top-3 influential members of community 84 and 89 are labeled, where "OA" stands for "Official Account". Best viewed in color.

community 84 (blue) and 89 (green), which are the top-2 influential communities. Figure 16 further labels the top-3 users of each community. Interestingly, the top members of C-84 are usually the official accounts of sports organizations, while those of C-89 are personal or unofficial ones, indicating distinct interaction patterns of these two influential types.

7. CONCLUSION

This paper addressed the problem of community-level diffusion analysis. We presented COLD (COmmunity Level Diffusion), a generative latent model jointly over network, text and time, to simultaneously uncover the hidden topics, communities, and inter-community influence.

With the well-designed model structure and parallel inference, COLD is effective and scalable. Based on the extracted community level representations, we developed an effective diffusion prediction approach. We also applied the model on real large datasets and performed temporal diffusion analysis. Meaningful patterns were discovered. COLD can be further used in influential community identification to promote viral marketing.

The community level diffusion analysis is a novel angle, and opens up several promising future directions. For example, the mechanism behind the coarse-grained diffusion and user engagement are beneficial for better temporal analysis and user targeting. More efficient and compact summarization techniques are also vital for dynamic and noisy data scenarios. We would like to extend current extraction method for advanced prediction and diffusion problems.

Acknowledgement: The research is supported by the National Natural Science Foundation of China under Grant No. 61272155, 61272340 and 61232006, and 973 program under No. 2014CB340405. This work is also supported by NSF IIS1218282.

8. REFERENCES

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

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[3] J. Chang and D. M. Blei. Relational topic models for document networks. In *Proc. of AISTATS*, pages 81–88, 2009.

[4] C. Chemudugunta, A. Holloway, P. Smyth, and M. Steyvers. *Modeling documents by combining semantic concepts with unsupervised statistical learning*. Springer, 2008.

[5] Q. Diao, J. Jiang, F. Zhu, and E.-P. Lim. Finding bursty topics from microblogs. In *Proc. of ACL*, pages 536–544, 2012.

[6] L. Dietz, S. Bickel, and T. Scheffer. Unsupervised prediction of citation influences. In *Proceedings of the 24th international conference on Machine learning*, pages 233–240. ACM, 2007.

[7] M. Eftekhar, Y. Ganjali, and N. Koudas. Information cascade at group scale. In *Proc. of KDD*, pages 401–409. ACM, 2013.

[8] J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing letters*, 12(3):211–223, 2001.

[9] M. Granovetter. The strength of weak ties: A network theory revisited. *Sociological theory*, 1(1):201–233, 1983.

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

[11] A. Guille, H. Hacid, C. Favre, and D. A. Zighed. Information diffusion in online social networks: A survey. *ACM SIGMOD Record*, 42(1):17–28, 2013.

[12] Q. Ho, R. Yan, R. Raina, and E. P. Xing. Understanding the interaction between interests, conversations and friendships in facebook. *CoRR*, abs/1211.0028, 2012.

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

[14] R. V. Kozinets, K. De Valck, A. C. Wojnicki, and S. J. Wilner. Networked narratives: understanding word-of-mouth marketing in online communities. *Journal of marketing*, 74(2):71–89, 2010.

[15] H. D. Lasswell. The structure and function of communication in society. *The communication of ideas*, 37, 1948.

[16] J. Leskovec, L. Backstrom, and J. Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proc. of KDD*, pages 497–506, 2009.

[17] J. Leskovec, K. J. Lang, and M. Mahoney. Empirical comparison of algorithms for network community detection. In *Proc. of WWW*, pages 631–640, 2010.

[18] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proc. of CIKM*, pages 556–559, 2003.

[19] S. Lin, F. Wang, Q. Hu, and P. S. Yu. Extracting social events for learning better information diffusion models. In *Proc. of KDD*, pages 365–373, 2013.

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

[21] Y. Liu, A. Niculescu-Mizil, and W. Gryc. Topic-link lda: joint models of topic and author community. In *Proc. of ICML*, pages 665–672, 2009.

[22] T. Lou and J. Tang. Mining structural hole spanners through information diffusion in social networks. In *Proc. of WWW*, pages 825–836, 2013.

[23] Y. Low, D. Bickson, J. Gonzalez, C. Guestrin, A. Kyrola, and J. M. Hellerstein. Distributed graphlab: a framework for machine learning and data mining in the cloud. *PVLDB*, 5(8):716–727, 2012.

[24] Y. Matsubara, Y. Sakurai, B. A. Prakash, L. Li, and C. Faloutsos. Rise and fall patterns of information diffusion: model and implications. In *Proc. of KDD*, pages 6–14, 2012.

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

[26] Y. Ruan, D. Fuhry, and S. Parthasarathy. Efficient community detection in large networks using content and links. In *Proc. of WWW*, pages 1089–1098, 2013.

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

[28] L. Tang and H. Liu. Community detection and mining in social media. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 2(1):1–137, 2010.

[29] Y. Tang, X. Xiao, and Y. Shi. Influence maximization: Near-optimal time complexity meets practical efficiency. In *Proc. of SIGMOD*, pages 75–86, 2014.

[30] C. Treadaway and M. Smith. *Facebook marketing: An hour a day*. John Wiley & Sons, 2012.

[31] B. Wang, C. Wang, J. Bu, C. Chen, W. V. Zhang, D. Cai, and X. He. Whom to mention: expand the diffusion of tweets by@ recommendation on micro-blogging systems. In *Proc. of WWW*, pages 1331–1340, 2013.

[32] X. Wang and A. McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *Proc. of KDD*, pages 424–433, 2006.

[33] S. Wu, J. M. Hofman, W. A. Mason, and D. J. Watts. Who says what to whom on twitter. In *Proc. of WWW*, pages 705–714, 2011.

[34] J. Xie, S. Kelley, and B. K. Szymanski. Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Comput. Surv.*, 45(4):43:1–43:35, Aug 2013.

[35] J. Yang and J. Leskovec. Patterns of temporal variation in online media. In *Proc. of WSDM*, pages 177–186, 2011.

[36] T. Yang, R. Jin, Y. Chi, and S. Zhu. Combining link and content for community detection: a discriminative approach. In *Proc. of KDD*, pages 927–936, 2009.

[37] H. Yin, B. Cui, H. Lu, Y. Huang, and J. Yao. A unified model for stable and temporal topic detection from social media data. In *Proc. of ICDE*, 2013.

[38] H. Yin, Y. Sun, B. Cui, Z. Hu, and L. Chen. Lcars: a location-content-aware recommender system. In *Proc. of KDD*, pages 221–229, 2013.

[39] Y. Zhu, X. Yan, L. Getoor, and C. Moore. Scalable text and link analysis with mixed-topic link models. In *Proc. of KDD*, pages 473–481, 2013.

APPENDIX

A. INFERENCE VIA COLLAPSED GIBBS SAMPLING

Here we describe the inference algorithm for COLD based on collapsed Gibbs Sampling.

Given an interaction network $\mathcal{G} = (\mathcal{U}, \mathcal{E})$ with a set of posts $\mathcal{D}$, and the pre-defined hyperparameters $\rho, \alpha, \beta, \epsilon$ and $\boldsymbol{\lambda}$, COLD specifies the following full posterior distribution:

$$\begin{aligned} &P(\boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\phi}, \boldsymbol{\psi}, \boldsymbol{c}, \boldsymbol{s}, \boldsymbol{z} | \mathcal{U}, \mathcal{E}, \mathcal{D}, \rho, \alpha, \beta, \epsilon, \boldsymbol{\lambda}) \propto \\ &\Big(P(\boldsymbol{\pi}|\rho) P(\boldsymbol{\theta}|\alpha) P(\boldsymbol{\eta}|\boldsymbol{\lambda}) P(\boldsymbol{\phi}|\beta) P(\boldsymbol{\psi}|\epsilon) P(\boldsymbol{c}, \boldsymbol{s}|\boldsymbol{\pi}) P(\boldsymbol{z}|\boldsymbol{\theta}, \boldsymbol{c}) \\ &P(\boldsymbol{w}_{\mathcal{D}}|\boldsymbol{\phi}, \boldsymbol{z}) P(\boldsymbol{t}_{\mathcal{D}}|\boldsymbol{\psi}, \boldsymbol{c}, \boldsymbol{z}) P(\boldsymbol{e}_{\mathcal{E}}|\boldsymbol{\eta}, \boldsymbol{s}) \Big), \end{aligned} \tag{8}$$

where $\boldsymbol{w}_{\mathcal{D}}$ is the words in the post set; $\boldsymbol{t}_{\mathcal{D}}$ is the time stamps of the posts; $\boldsymbol{e}_{\mathcal{E}}$ is the set of positive links; the constant of proportionality is the marginal likelihood of the observed data.

The task of posterior inference for COLD is to determine the probability distribution of the hidden variables given the observed words, time stamps and network. However, exact inference is intractable due to the difficulty of calculating the normalizing constant in the above posterior distribution.

We use collapsed Gibbs Sampling, a well-established Markov chain Monte Carlo (MCMC) technique for approximate inference. In collapsed Gibbs Sampling, the multinomial distributions $\boldsymbol{\Phi} = \{\boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\phi}, \boldsymbol{\psi}\}$ are first marginalized (collapsed), a Markov chain over the latent indicators $\{\boldsymbol{c}, \boldsymbol{s}, \boldsymbol{z}\}$ is then constructed, whose stationary distribution is the posterior. We obtain samples of latent variables from the Markov chain. Point estimates for the collapsed distributions $\boldsymbol{\Phi}$ can then be computed given the samples, and predictive distributions are computed by averaging over multiple samples.

Sampling Procedure. Gibbs Sampler repeatedly samples each latent variable conditioned on the current states of other hidden variables and observations; a configuration of latent states of the system is then obtained. Next we provide the derivation of the sampling formulas (Eqs.(1–3)).

By marginalizing out $\boldsymbol{\Phi}$ in Eq.(8), we obtain:

$$\begin{aligned} &P(\boldsymbol{c}, \boldsymbol{s}, \boldsymbol{z}|.) \\ &\propto P(\boldsymbol{c}, \boldsymbol{s}|\rho) P(\boldsymbol{z}|\boldsymbol{c}, \alpha) P(\boldsymbol{w}|\boldsymbol{z}, \beta) P(\boldsymbol{t}|\boldsymbol{c}, \boldsymbol{z}, \epsilon) P(\boldsymbol{e}|\boldsymbol{s}, \boldsymbol{\lambda}) \\ &= \int P(\boldsymbol{\pi}|\rho) P(\boldsymbol{c}, \boldsymbol{s}|\boldsymbol{\pi}) d\boldsymbol{\pi} \int P(\boldsymbol{\theta}|\alpha) P(\boldsymbol{z}|\boldsymbol{\theta}, \boldsymbol{c}) d\boldsymbol{\theta} \\ &\cdot \int P(\boldsymbol{\phi}|\beta) P(\boldsymbol{w}|\boldsymbol{\phi}, \boldsymbol{z}) d\boldsymbol{\phi} \int P(\boldsymbol{\psi}|\epsilon) P(\boldsymbol{t}|\boldsymbol{\psi}, \boldsymbol{c}, \boldsymbol{z}) d\boldsymbol{\psi} \\ &\cdot \int P(\boldsymbol{\eta}|\boldsymbol{\lambda}) P(\boldsymbol{e}|\boldsymbol{\eta}, \boldsymbol{s}) d\boldsymbol{\eta}. \end{aligned} \tag{9}$$

The conditional of c_{ij} can be computed by dividing the joint distribution of all variables by the joint of all variables except c_{ij} (denoted as $\boldsymbol{c}_{-ij}$):

$$\begin{aligned} &P(c_{ij} = c | \boldsymbol{c}_{-ij}, \boldsymbol{s}, \boldsymbol{z}, \boldsymbol{t}, .) \\ &= \frac{P(\boldsymbol{c}, \boldsymbol{s}, \boldsymbol{z}|.)}{P(\boldsymbol{c}_{-ij}, \boldsymbol{s}, \boldsymbol{z}|.)} \\ &= \frac{P(\boldsymbol{c}, \boldsymbol{s}|\rho)}{P(\boldsymbol{c}_{-ij}, \boldsymbol{s}|\rho)} \cdot \frac{P(\boldsymbol{z}|\boldsymbol{c}, \alpha)}{P(\boldsymbol{z}|\boldsymbol{c}_{-ij}, \alpha)} \cdot \frac{P(\boldsymbol{t}|\boldsymbol{c}, \boldsymbol{z}, \epsilon)}{P(\boldsymbol{t}|\boldsymbol{c}_{-ij}, \boldsymbol{z}, \epsilon)}. \end{aligned} \tag{10}$$

We now derive the first fraction of Eq.(10), i.e.,

$$\frac{P(\boldsymbol{c}, \boldsymbol{s}|\rho)}{P(\boldsymbol{c}_{-ij}, \boldsymbol{s}|\rho)} = \frac{\int P(\boldsymbol{\pi}|\rho) P(\boldsymbol{c}, \boldsymbol{s}|\boldsymbol{\pi}) d\boldsymbol{\pi}}{\int P(\boldsymbol{\pi}|\rho) P(\boldsymbol{c}_{-ij}, \boldsymbol{s}|\boldsymbol{\pi}) d\boldsymbol{\pi}}. \tag{11}$$

As we assume each c is generated from a multinomial distribution $\boldsymbol{\pi}$, and the hyper-parameter for conjugate Dirichlet prior is ρ, we have:

$$\begin{aligned} &\int P(\boldsymbol{\pi}|\rho) P(\boldsymbol{c}, \boldsymbol{s}|\boldsymbol{\pi}) d\boldsymbol{\pi} \\ &= \int \prod_i \frac{\Gamma(C\rho)}{\prod_c \Gamma(\rho)} \prod_c \pi_{ic}^{\rho - 1} \cdot \prod_i \prod_c \pi_{ic}^{n_i^{(c)}} d\boldsymbol{\pi} \\ &= \prod_i \frac{\Gamma(C\rho)}{\prod_c \Gamma(\rho)} \cdot \frac{\prod_c \Gamma(n_i^{(c)} + \rho)}{\Gamma(n_i^{(\cdot)} + C\rho)}. \end{aligned}$$

Combining the above equation with Eq.(11) leads to:

$$\begin{aligned} \frac{P(\boldsymbol{c}, \boldsymbol{s}|\rho)}{P(\boldsymbol{c}_{-ij}, \boldsymbol{s}|\rho)} &= \frac{\Gamma(n_i^{(c_{ij})} + \rho) \Gamma(n_{i,-ij}^{(\cdot)} + C\rho)}{\Gamma(n_{i,-ij}^{(c_{ij})} + \rho) \Gamma(n_i^{(\cdot)} + C\rho)} \\ &= \frac{n_{i,-ij}^{(c)} + \rho}{n_{i,-ij}^{(\cdot)} + C\rho}, \end{aligned} \tag{12}$$

where the count with subscript $-ij$ denotes a quantity with the current instance (i.e. post d_{ij}) excluded. Here we use the identity $\Gamma(x+1) = x\Gamma(x)$. The second and third fractions of Eq.(10) can be derived analogously. The Dirichlet-Multinomial conjugates ensure the tractability of the integrals. Specifically, the second fraction can be written as:

$$\frac{P(\boldsymbol{z}|\boldsymbol{c}, \alpha)}{P(\boldsymbol{z}|\boldsymbol{c}_{-ij}, \alpha)} = \frac{n_{c,-ij}^{(k)} + \alpha}{n_{c,-ij}^{(\cdot)} + K\alpha}, \tag{13}$$

while the third fraction as:

$$\frac{P(\boldsymbol{t}|\boldsymbol{c}, \boldsymbol{z}, \epsilon)}{P(\boldsymbol{t}|\boldsymbol{c}_{-ij}, \boldsymbol{z}, \epsilon)} = \frac{n_{ck,-ij}^{(t)} + \epsilon}{n_{ck,-ij}^{(\cdot)} + T\epsilon}. \tag{14}$$

Finally, by combining Eqs.(12-14) we obtain the sampling formula as in Eq.(1). Note that we omit the subscripts $-ij$ in Eq.(1) for clarity. Eq.(2) and Eq.(3) are derived in a similar manner.

Distribution Estimation. After a sufficient number of sampling iterations, we obtain a set of samples. For any single sample, we can estimate the unknown distributions as follows:

$$\hat{\pi}_{ic} = \frac{n_i^{(c)} + \rho}{n_i^{(\cdot)} + C\rho},$$

$$\hat{\theta}_{ck} = \frac{n_c^{(k)} + \alpha}{n_c^{(\cdot)} + K\alpha},$$

$$\hat{\eta}_{cc'} = \frac{n_{cc'} + \lambda_1}{n_{cc'} + \lambda_0 + \lambda_1},$$

while $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ can be estimated similarly. The final predictive distributions are obtained by integrating across all the samples.

B. RESULTS ON PARAMETER SENSITIVITY

***K* on Topic Extraction.** Figure 17 shows the topic perplexity values under different parameter settings. Given a fixed C, the perplexity decreases with the increasing number of topics, and levels off after K is larger than 100. On the other hand, under any fixed K, the result remains stable as C varies, indicating that the number of communities is less important than the number of topics for text modeling. In COLD, text is generated by mixture of topics, hence the number of topics directly impacts the capacity of modeling text. In contrast, although there exists correlations between content and network, the influence of communities on text modeling is indirect.

***C* on Community Extraction.** Figure 18 shows the impacts of C and K w.r.t. the quality of COLD in community modeling. The AUC value at first increases as C increases, and there is an

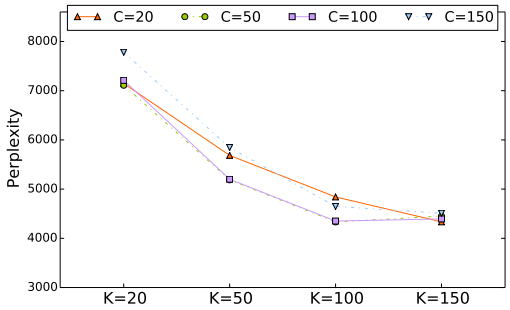


Figure 17: #Community C and #Topic K Impacts on Topic Extraction.

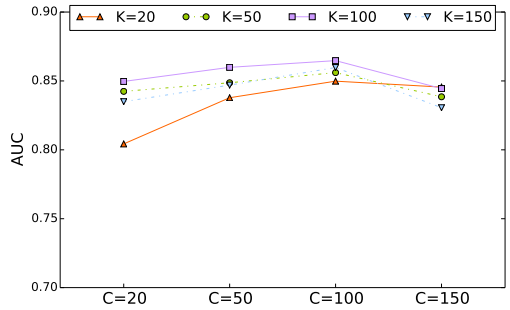


Figure 18: #Community C and #Topic K Impacts on Link Prediction.

intermediate value of C (i.e. 100) at which COLD performs best. After that the AUC value decreases as C continues to increase. In contrast, the result fluctuates slightly without a clear pattern as K varies. The result is expected as links are directly generated by mixture of communities. The effect of topic number is not significant.

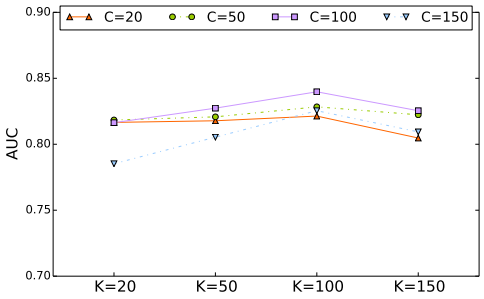


Figure 19: #Community C and #Topic K Impacts on Diffusion Prediction.

Joint Impacts on Diffusion Prediction. Figure 19 shows the diffusion prediction performance under different parameter settings. We find that the prediction AUC values are affected by both C and K, e.g., the performance gets better when K and C increases from 20 to 100, respectively. The joint effect indicates that communities and topics are both critical factors in modeling diffusion process. Besides, the clear trends w.r.t. K and C provide a useful guidance for model selection, e.g., a wide range of K values (from 20 to 100 for this data) would provide good performance.