From Community Detection to Community Profiling

Hongyun Cai [†‡], Vincent W. Zheng [†], Fanwei Zhu [#], Kevin Chen-Chuan Chang [⋄], Zi Huang [‡]
[†] Advanced Digital Sciences Center, Singapore [‡] School of ITEE, The University of Queensland, Australia
[#] Zhejiang University City College, China [⋄] University of Illinois at Urbana-Champaign, USA
{hongyun.c, vincent.zheng}@adsc.com.sg, zhufanwei@zju.edu.cn, kcchang@illinois.edu, huang@itee.uq.edu.au

ABSTRACT

Most existing community-related studies focus on detection, which aim to find the community membership for each user from user friendship links. However, membership alone, without a complete profile of what a community is and how it interacts with other communities, has limited applications. This motivates us to consider systematically profiling the communities and thereby developing useful community-level applications. In this paper, we for the first time formalize the concept of community profiling. With rich user information on the network, such as user published content and user diffusion links, we characterize a community in terms of both its internal content profile and external diffusion profile. The difficulty of community profiling is often underestimated. We novelly identify three unique challenges and propose a joint Community Profiling and Detection (CPD) model to address them accordingly. We also contribute a scalable inference algorithm, which scales linearly with the data size and it is easily parallelizable. We evaluate CPD on large-scale real-world data sets, and show that it is significantly better than the state-of-the-art baselines in various tasks.

1. INTRODUCTION

Thanks to the pioneer studies on community detection [17, 33], we have been able to model a community in terms of its member users. Such community membership assists us to better understand the network structure. However, membership alone, without knowing *what a community is* and *how it interacts with others*, has only limited applications– e.g., we cannot rank communities by desired characteristics, exploit inter-community diffusions, and visualize communities and their interactions. With this critical lacking of community "understanding", this paper proposes systematic *community profiling*– to characterize the intrinsic nature and extrinsic behavior of a community– thereby enabling useful community-level applications. As social networks increasingly capture more and richer user information, it is now feasible to profile communities. E.g., beyond traditional *friendship links* which connect users on a social network, there are also users' attributes, published content, diffused content and so on. We can leverage such rich user data to estimate the community profiles.

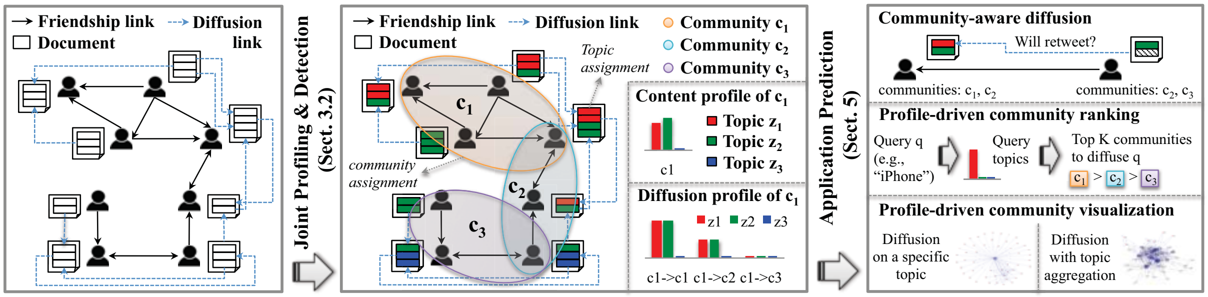
In this paper, we for the first time formalize the concept of "community profile". We ask two fundamental questions:

• *What is a community profile*? By name, the profile should characterize a community, both internally (i.e., what it is) and externally (i.e., how it interacts with others). Since a community is an aggregation of users, its profile is essentially an aggregation of user information. Denote X as some type of user information. To accommodate uncertainty in X, we define an internal profile as probabilities of "community-X", and an external profile as probabilities of "community-community-X". Here we focus on X as *content*, which is the primary user information in many social networks. E.g., in Twitter, users write tweets and retweet from others; in DBLP, authors publish papers and cite papers from others. We call the probabilities of "community-content" as *content profile* (i.e., what a community is about), and those of "community-community-content" as *diffusion profile* (i.e., how a community diffuses certain content with another). Other types of X's may exist in different networks, e.g., attributes in Facebook. Thus, "community profile" is a flexible concept. We leave other types of X's as future work.

• *Is a community profile good*? Due to network homophily, users in the same community tend to have similar behaviors. Thus, a community's profile should explain the common behaviors of its users. In other words, we do not see any "community-content" distribution as a good content profile; instead, only that well explaining the observations of user content from communities' perspectives is a "good" one (i.e., contents are generated by communities through profiles). Analogously, only the "community-community-content" distribution that well explains the observations of user-to-user content diffusion as generated by the communities is a "good" diffusion profile. This quality criterion will later guide us to estimate the profiles accurately. It is also a key to differentiating us from other work– some prior attempt simply aggregates user information to output community properties (mostly internal ones) [12], but it does not require such properties to best explain the observations of user behaviors as generated by the communities through them.

We consider "community profiling" as a new problem due to three reasons. First, community profiling is different from community detection, because detection focuses on getting users' community memberships, whereas profiling focuses on the "community-X" and "community-community-X" probabilities. Second, community profile has never been defined. Some recent work exploits rich user information [13, 28, 35, 36, 37] to improve community detection and outputs aggregated user information as a by-product. But they neither define a community profile both internally and externally, nor identify new applications that community profiles enable. Finally, the difficulty of community profiling is often underestimated. As we shall discuss soon, due to the inter-dependency with community detection, the heterogeneity of social observations

Proceedings of the VLDB Endowment, Vol. 10, No. 7
Copyright 2017 VLDB Endowment 2150-8097/17/03.



(a) Problem input (Sect. 3.1) **(b) Problem output after inference (Sect. 4)** **(c) Applications (Sect. 6)**

Figure 1: The framework of joint community profiling and detection.

and the nonconformity of user behaviors, finding a good community profile is challenging. None of the existing work has ever identified and addressed such challenges (more discussions in Sect. 2).

Our goal is to infer content profile and diffusion profile for each community, and ultimately enable new applications. In Fig. 1(a), we show the input for community profiling: a set of users, each of whom publishes documents; users are connected by friendship links, and interact with each other by diffusion links. E.g., in Twitter, each user posts tweets, users are connected by followership links, and they retweet each other to diffuse information. In Fig. 1(b), for each community, we output: a content profile (e.g., community c_1 tends to publish topics z_1 and z_2) and a diffusion profile (e.g., c_1 tends to diffuse itself and c_2 on z_1). In Fig. 1(c), we enable three new applications as follows (novelty to be discussed in Sect. 2, applications to be concretized in Sect. 5 and evaluated in Sect. 6):

- *Community-aware diffusion*. As community profiles aggregate user behaviors, we can use them to more robustly model the diffusion at a community level, rather than an individual level [8, 19, 21]. E.g., we can explain a retweet happens as one user's communities often retweet the other's on a certain topic. We acknowledge diffusion as a complex decision– beyond community profiles, there are also nonconformity factors such as individual preference and topic popularity. This partially explains why community profiling is challenging– we cannot account community profiles for all the diffusions; instead, we have to model different factors, to accurately estimate the profiles and the community-aware diffusion.

- *Profile-driven community ranking*. We often need to target audiences for disseminating information on the networks. E.g., a company wants to target communities, which are most likely to retweet about its product, so as to launch a campaign. A funding agency wants to target communities, which actively cite papers about its grant theme on "deep learning", so as to disseminate the grant call. Since we have known what content each community is interested in and how it diffuses that content with others, we can rank the communities. Profile-driven community ranking is different from the traditional community recommendation, which often relies on only "community-X" properties and is unaware of diffusion [6, 12].

- *Profile-driven community visualization*. Holistic modeling leads to rich visualization– we can now visualize not only how communities feature distinct contents (e.g., what an IT community tweets), but also how they interact (e.g., how an IT community retweets others) which is often overlooked before [7, 20].

We make two remarks about the above applications: 1) we complete one task of community profiling to support multiple applications at a time, thus community profiling is only done once offline; 2) we build an interactive system[1] for profile-driven community visualization and ranking, which for the first time allows people to freely browse the communities by both content and diffusion.

The difficulty of community profiling is often largely underestimated; as we shall discuss next, there exist many challenges:

- *Inter-dependency with community detection*. A straightforward approach of community profiling is to first detect communities and then aggregate each community's user observations as the profiles. However, because this approach does not try to "best explain" the user observations as generated by the communities through their profiles, it is often suboptimal. Take content profile as an example. Denote a user as u and a community as c. For simplicity, we denote c's content profile as $p(\text{content}|c)$ and the likelihood of u's content as $p(\text{content}|u)$. To best explain the user content as generated by the communities through their content profiles, we effectively solve

$$\max \prod_u p(\text{content}|u) = \prod_u \sum_c p(\text{content}|c)p(c|u), \qquad (1)$$

where $p(c|u)$ is the probability of user u assigned to community c. Ideally, to optimize Eq. 1, we shall optimize both the profile $p(\text{content}|c)$'s and the community assignment $p(c|u)$'s. But in the straightforward approach, the detection first fixes the $p(c|u)$'s, then the best result this aggregation can return is the $p(\text{content}|c)$'s that maximize Eq. 1. It is clear that, the maximal likelihood we get with fixed $p(c|u)$'s is suboptimal, unless the $p(c|u)$'s are "perfect". However, a perfect detection of $p(c|u)$'s also needs to maximize the likelihood in Eq. 1, which depends on the profile $p(\text{content}|c)$'s. In all, content profiles and community detection are coupled. We further show in our technical report [3] that diffusion profiles and community detection are also coupled.

- *Heterogeneity of social observations*. Social observations, especially the user links (i.e., friendship links and diffusion links), often carry different semantics; e.g., friendship links indicate user connections and diffusion links indicate user interactions. Traditionally, we often try to enforce user connections to be denser within each community than across communities [14, 17]. But in diffusion, the "weak ties" theory recognizes that the inter-community interactions may not be weak [10]. E.g., software engineering community cites more papers from machine learning community than itself on "deep learning". This means we have to separate the modeling of user connection and user diffusion. Such user link heterogeneity is largely overlooked in the previous work [27, 30], thus how to model heterogeneous user links together remains unclear.

- *Nonconformity of user behaviors*. User behaviors, especially their diffusion decisions, can happen for many reasons. Community-level conformity is just one reason, thus we have to consider other

[1] http://sociallens.adsc.com.sg/

factors as well. E.g., some diffusion happens due to its topic (e.g., presidential election) being popular at the moment or its author (e.g., Lady Gaga) being preferred as a celebrity. Such topic popularity and user preference are the other two typical nonconformity factors for diffusion, and we must accommodate them. No prior work has explored both community factor and nonconformity factors [15, 21], and it is not clear how to balance them in diffusion.

Our technical novelty is identifying the above challenges and developing a unified Community Profiling and Detection (CPD) model (Sect. 3) to address them accordingly.

- To model the *inter-dependency with community detection*, we propose a novel profile-aware generative approach– we realize the detection by latent membership variables and the profiling by latent community profile variables, which together generate friendship links, user content and diffusion links. Then we infer these latent variables by maximizing the likelihood. None of the existing work has taken a profile-aware generative approach– they may use a generative model for detection [22, 28], but they never consider internal and external profiles together with detection.
- To address the *heterogeneity of social observations*, we propose to separate the generation of friendship links from latent community assignments and the generation of diffusion links from latent profiles. In particular, we require that two users are more likely to share a friendship link if they have similar community assignments. Thus maximizing the likelihood of observing the friendship links enforces intra-community friendship links to be denser than inter-community ones. In contrast, we use the community diffusion profiles to generate the diffusion links without requiring inter-community diffusion strengths to be always smaller than the intra-community ones; instead, the diffusion profiles are freely learned in maximizing the likelihood of the diffusion link observations.
- To accommodate the *nonconformity of user behaviors*, we propose to define the generative probability of observing a diffusion link as a logistic function over multiple factors, including the topic-aware community diffusion profiles, the time-sensitive topic popularities and the individual user preferences. By maximizing the likelihood of diffusion link observations, we learn the diffusion profiles, as well as the weights to combine these different factors.

Finally, we design a scalable inference algorithm for CPD (Sect. 4). As shown later, our inference algorithm scales linearly to the data set size. We further parallelize our inference algorithm, by taking the data skewness into account.

We summarize our contributions as follows:

- We identify a *new problem of community profiling*, which together with detection enables a holistic modeling of communities.
- We identify *three unique challenges* and design *a novel CPD model* for joint community profiling and detection.
- We develop *a scalable inference algorithm* for CPD, and we further parallelize it by taking the data skewness into account.
- We perform *extensive experiments* to evaluate CPD over large-scale data sets, and show both its effectiveness and scalability.

2. RELATED WORK

In this section, we review the related work on community detection and relevant applications, and distinguish the differences between existing work and our CPD model. We further organize such differences by a table in our technical report [3].

Community Detection. Detecting communities from various networks has been extensively studied in the last decade. There exist comprehensive surveys [34, 17, 31] on community detection, which review different community detection methods in terms of detection algorithms, quality measures, benchmarks and so on.

Conventionally, a community is defined as a group of nodes, in which intra-group connections are much denser than inter-group ones [33]. The pioneer community detection studies aim to generate the community membership for each node purely based on user links [17, 33]. The prevalence of social networks offers a rich collection of user links to be used for community detection, such as the followership in Twitter [27], Flickr [26] and Facebook/Google+ [22, 36], the co-authorship in DBLP [35, 37], the email exchange [27]. However, most of these existing work only consider one single type of links. There are other different types of user links; e.g., users comment/reply other users in digg [20], contact/co-contact/co-subscribe other users in YouTube [30]. But these different links were often modeled in the same way. So far as we know, none of the existing community work considers the heterogeneity among user links (i.e., friendship links and diffusion links) as we do.

Recent studies start to exploit the rich user information, such as content [27], attribute [28, 35], action [20], to improve the detection. Consequently, in addition to community membership, they also occationally output some "community-X" associations, such as "community-content" [27], "community-attribute" [28, 35] and "community-action" [20]. In our work, we simultaneously discover communities and characterize them with both internal and external profiles. Although some forms of internal community profiles may be obtained in some prior work [20, 27, 28, 35] as the by-products, the external profiles are greatly overlooked.

There are some recent studies on aggregating each community's user preferences as some form of community profiles, so as to enable item recommendation to each community. Their work is different from ours in two aspects. On one hand, most of these studies are given the communities as input [12, 25]. Even though some of them did try to detect communities [1, 23], their definition of a community is a group of users who share similar preferences to a recommended item, which is not based on network links at all. In contrast, our community is a group of densely connected users, who share similar interests and diffusion behaviors. On the other hand, their community profile is obtained by aggregating users' preferences, which is usually based on a *least misery* or *aggregate voting* approach. In contrast, we formalize the community profiles as the probabilities of "community-X" and probabilities of "community-community-X". Besides, we estimate community profiles by a generative model together with community detection.

Community-aware Applications. The community profiles deepen our understanding of the detected communities and thus benefit a lot of community-level applications. Here we review the related work to our three example applications, including community ranking, community diffusion and community visualization. Firstly, for community ranking, most of existing studies [12, 6] rank communities based on users' interests on them, i.e., to find the favorite communities for users. Moreover, the communities to be ranked are often already predefined over the networks. In our work, the communities are not provided as the input, and our focus is to rank communities by both their internal content profiles and external diffusion profiles together. This will help the company/author to choose the promising community to promote their products/papers as much as possible. Secondly, for community diffusion, in contrast to our community-level diffusion modeling, most diffusion models are at the individual level [8, 19, 21]. Recently, there are some studies that consider diffusion at the community level, but either the communities are predefined [9] or the topic-awareness is overlooked [13, 15]. Besides, unlike our modeling of various diffusion factors together, the individual factor is missing in [15] and the

Table 1: Notations

Notations	Description
U, D, W	The set of users, documents and words
C, Z	The set of communities and topics
F, E	The set of friendship links and diffusion links
$\|A\|$	The number of elements in set A
D_u	The set of documents published by user u
d_{ui}	The i-th document published by user u
W_{ui}	The set of words in document d_{ui}
w_{uik}	The k-th word in document d_{ui}
c_{ui}, z_{ui}	The community assignment and topic assignment for d_{ui}
E^t_{ij}	A diffusion link from document i to document j at time t
F_{uv}	A friendship link from user u to user v
$\boldsymbol{\pi}_u$	Multinomial distribution over communities specific to user u
$\boldsymbol{\theta}_c$	Multinomial distribution over topics specific to community c
$\boldsymbol{\phi}_z$	Multinomial distribution over words specific to topic z
$\eta_{c,c'z}$	Probability of community c diffusing community c' on topic z
$\boldsymbol{\nu}$	The parameters for modeling individual diffusion preference
α, β, ρ	Dirichlet priors

topic popularity factor is missing in [13]. Last, but not the least, for community visualization, although a lot of efforts have been devoted to community detection, only a few of them further visualize the results to facilitate the deep analysis and semantic interpretation. In [7], the authors propose a community detection and visualization model, which differentiates the inner nodes and the border nodes for visualizing the interactions between communities. While their objective is to design a layout algorithm for clearly displaying the communities and their interactions, we focus on demonstrating the topic-aware user interaction strengths among the communities.

3. JOINT PROFILING AND DETECTION

In the following, we first define some key notions; then we formulate the joint community profiling and detection problem. Table 1 summarizes the notations used in this paper.

DEFINITION 1. *A **social graph** is $\mathcal{G} = (U, D, F, E)$, where $u \in U$ is a user and $d \in D$ is a user published document. There are two types of links in $\mathcal{G}$. $F_{uv} \in F$ is a* friendship link *from user u to user v; $E_{ij} \in E$ is a* diffusion link *from document i to document j. Both types of links are directed.*

For a Twitter network, $D_u \subset D$ is the set of tweets posted by user u; F_{uv} represents that user u follows user v; E_{ij} represents that tweet i is a retweet of tweet j. For a DBLP network, D_u is the set of papers published by author u; F_{uv} represents that author u co-authors with author v; E_{ij} represents that paper i cites paper j.

To enable content modeling, we first define topic.

DEFINITION 2. *A **topic** $z \in Z$ is a $|W|$-dimensional multinomial distribution $\boldsymbol{\phi}_z$ over words, where each dimension $\phi_{z,w}$ is the probability of a word $w \in W$ belonging to z.*

Then, we define the community membership, as well as our community content profile and diffusion profile.

DEFINITION 3. *A user u's **community membership** is a $|C|$-dimensional multinomial distribution $\boldsymbol{\pi}_u$, where each dimension $\pi_{u,c}$ is the probability of u belonging to community c, $\forall c \in C$.*

DEFINITION 4. *The **content profile** of community c is a $|Z|$-dimensional multinomial distribution $\boldsymbol{\theta}_c$ over topics, where each dimension $\theta_{c,z}$ is the probability of c discussing topic z.*

DEFINITION 5. *The **diffusion profile** of community c is a $|C| \times |Z|$-dimensional matrix $\boldsymbol{\eta}_c$, where each entry $\eta_{c,c'z}$ is the probability of c diffusing another community c' on topic z.*

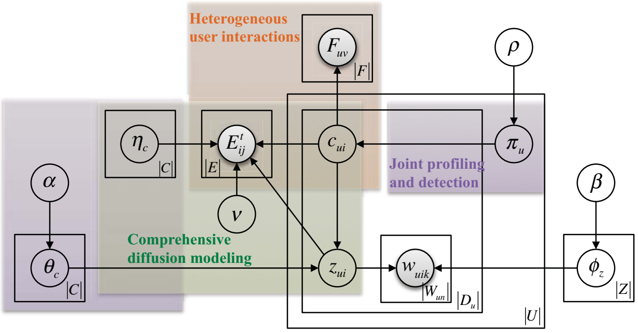


Figure 2: Graphical model of CPD.

Take community c_1 in Fig. 1 as an example. As c_1's users publish more content on z_1 and z_2, the resulting θ_{c_1,z_1} and θ_{c_1,z_2} are bigger. Besides, as c_1's users often retweet/cite themselves on z_1, the resulting η_{c_1,c_1z_1} is big. As motivated in Sect. 1, we formalize a joint profiling and detection problem to solve in this paper.

PROBLEM 1. *Given a social graph $\mathcal{G} = (U, D, F, E)$, the task of **joint community profiling and detection** is to infer: 1) each user u's community membership $\boldsymbol{\pi}_u$, $\forall u \in U$; 2) each community c's content profile $\boldsymbol{\theta}_c$ and diffusion profile $\boldsymbol{\eta}_c$, $\forall c \in C$.*

3.1 Model Design

Next, we concretize our model design w.r.t. the three technical challenges for community profiling as discussed in Sect. 1. We will later evaluate how well we address each challenge in Sect. 6.2.

Profile-aware generative model. Community detection aims to infer a community membership assignment $\boldsymbol{\pi}_u$ for each user u based on the friendship links F_{uv}'s. Community profiling aims to infer a content profile $\boldsymbol{\theta}_c$ and a diffusion profile $\boldsymbol{\eta}_c$ for each community c based on its member users' published content D_u's and diffusion links E^t_{ij}'s. We can reinforce profiling and detection, by letting them leverage each other's data. As a result, we wish to infer a set of community-level latent variables, including $\boldsymbol{\pi}_u$'s, $\boldsymbol{\theta}_c$'s and $\boldsymbol{\eta}_c$'s, together from all the observations (D, F, E).

Since joint profiling and detection is an unsupervised task, we adopt a generative framework for our CPD model. We design CPD as a graphical model in Fig. 2, where we use communities to explain all user observations. Firstly, we consider a user u to publish a document d_{ui} of topic z, due to her community assignment c_{ui} and the community content profile $\boldsymbol{\theta}_{c_{ui}}$. E.g., an author publishes a paper on deep learning, because she is from the machine learning (ML) community, which studies deep learning. As we deal with short documents (e.g., tweets in Twitter and paper titles in DBLP) and a short document is likely to be about one single topic [15], we assign one single topic to each document in CPD. Secondly, we consider a user u to publish a document d_{ui} of topic z, which diffuses another user v's document d_{vj}, due to both users' community assignments c_{ui} and c_{vj}, as well as the community diffusion profile $\eta_{c_{ui},c_{vj}z}$. E.g., an author u publishes a paper on software repositories, and cites another author v's paper on deep learning, because u is from the software engineering (SE) community, v is from the ML community, and SE community tends to cite papers on deep learning from the ML community. Finally, we consider a user u to form a friendship link with a user v, due to their similar community memberships $\boldsymbol{\pi}_u$ and $\boldsymbol{\pi}_v$. E.g., an author u is a co-author of another author v, because they are both from the ML community.

Addressing data heterogeneity. We model the friendship links F and the diffusion links E differently. Conventionally, a good community needs to have low conductance, which means the friendship links should be denser inside a community than outside a community. Specifically, we define the probability of having a friendship link between two users u and v as a sigmoid function, parameterized by their community membership similarity:

$$P(F_{uv}=1)=\sigma(\hat{\boldsymbol{\pi}}_u^T\hat{\boldsymbol{\pi}}_v), \tag{2}$$

where $\hat{\boldsymbol{\pi}}_u = [\hat{\pi}_{u,1},...,\hat{\pi}_{u,|C|}]^T$ is an estimation of $\boldsymbol{\pi}_u$ based on the aggregation of u's community assignments. In other words, we use $\hat{\boldsymbol{\pi}}_u$ and $\hat{\boldsymbol{\pi}}_v$, instead of $\boldsymbol{\pi}_u$ and $\boldsymbol{\pi}_v$, to generate the F_{uv}'s in Fig. 2. Such a design is motivated by [4, 5] to simplify the inference. $\sigma(x)=1/(1+e^{-x})$ is a sigmoid function. The more similar $\hat{\boldsymbol{\pi}}_u$ and $\hat{\boldsymbol{\pi}}_v$ are, the more likely F_{uv} exists. In other words, F_{uv} is large if u and v are from the same communities. This naturally enforces denser friendship links within a community than across communities, thus leading to low conductance. In contrast with the friendship links, the inter-community diffusion is not necessarily "weak" [10]. In fact, the community-level diffusion strengths vary over topics, which breaks the assumption of having to maintain the low conductance within a community. We need to resort to a different modeling of diffusion links, as we discuss next.

Accommodating nonconformity. Different factors can account for a diffusion decision. Take Twitter as an example; user u is likely to retweet v's tweet d_{vj} as her i-th tweet d_{ui} at time t if: 1) the community-level diffusion strength between c_{ui} (the community u belongs to when she generates document d_{ui}) and c_{vj} on topic z_{vj} is strong; 2) the topic z_{vj} of d_{vj} is trending at time t; 3) u has an individual preference to retweet from v. These factors show three typical perspectives to make a diffusion decision: community perspective (if a community is more likely to retweet another community), content perspective (if a topic is more popular at the time) and user perspective (if a user is more likely to retweet another user). Next, we characterize the three typical factors.

• *Community diffusion preference*: we consider a user u to diffuse another user v on topic z, if the communities of u and v are both interested in z and they often diffuse each other on z. Denote $s \in \{0,1\}$ as an indicator for a diffusion link in E to happen. Then, the probability of having a diffusion $s=1$ from u to v on z is

$$\begin{aligned} p(s=1,z|u,v) &\overset{1}{=} \sum_c\sum_{c'} p(s=1|c,c',z)p(z|c)p(z|c')p(c'|v) \\ &\overset{2}{\propto} \sum\nolimits_c \sum\nolimits_{c'} \eta_{c,c'z}\hat{\theta}_{c,z}\hat{\pi}_{u,c}\hat{\theta}_{c',z}\hat{\pi}_{v,c'}, \end{aligned} \tag{3}$$

where at step 1 we expand $p(s=1,z|u,v)$ by introducing the community membership $p(c|u)$ and $p(c'|v)$, the communities' interests on the topic $p(z|c)$ and $p(z|c')$, as well as the topic-sensitive community diffusion probability $p(s=1|c,c',z)$. At step 2, we estimate $p(s=1|c,c',z)$ with $\eta_{c,c'z}$, the probability of c retweeting/citing c' on z. Besides, we estimate $p(c|u)$ with $\hat{\pi}_{u,c}$, which is the empirical probability of community c being assigned to user u; similarly we estimate $p(c'|v)$ with $\hat{\pi}_{v,c'}$. Finally we estimate $p(z|c)$ with $\hat{\theta}_{c,z}$, which is the empirical probability of topic z assigned to the documents from c; similarly we estimate $p(z|c')$ with $\hat{\theta}_{c',z}$. Denote $\hat{\boldsymbol{\theta}}_{\cdot,z}=[\hat{\theta}_{1,z},...,\hat{\theta}_{|C|,z}]^T$ and $\bar{\boldsymbol{\eta}}=vec([\boldsymbol{\eta}_1,...,\boldsymbol{\eta}_{|C|}])$, where $vec(\mathbf{A})$ concatenates the row vectors in a matrix $\mathbf{A}$ to a vector. For a diffusion between d_{ui} and d_{vj}, which shares the same topic z, we denote $\bar{\mathbf{c}}_{ij}=vec((\hat{\boldsymbol{\pi}}_u\hat{\boldsymbol{\pi}}_v^T)\circ(\hat{\boldsymbol{\theta}}_{\cdot,z}\hat{\boldsymbol{\theta}}_{\cdot,z}^T))$, where $\circ$ is an element-wise product. Then Eq. 3 becomes $\bar{\mathbf{c}}_{ij}^T\bar{\boldsymbol{\eta}}$.

• *Topic popularity*: we model the popularity of a topic at a specific timestamp t as the count of topic z at t, which is denoted as n_z^t.

• *Individual preference*: we model user u's preference to diffuse information from user v with a linear function $\boldsymbol{\nu}^T\mathbf{f}_{uv}$, where $\boldsymbol{\nu}$ is a parameter, $\mathbf{f}_{uv}$ is a feature vector for u and v. Take Twitter as an example; we consider two features for u: 1) *user popularity*, which is defined as the number of u's followers divided by that of her followees $\frac{|Followers(u)|}{|Followees(u)|}$; 2) *user activeness*, which is defined as the number of u's retweets divided by that of her tweets $\frac{|Retweets(u)|}{|Tweets(u)|}$. We extract v's features and concatenate them with u's as $\mathbf{f}_{uv}$.

In order to systematically combine the three diffusion factors, we introduce a sigmoid function to define the probability of document d_{vj} diffusing document d_{ui} of topic z at timestamp t as:

$$p(E_{ij}^t=1|u,v,z,t)=\sigma(\bar{\mathbf{c}}_{ij}^T\bar{\boldsymbol{\eta}}+n_z^t+\boldsymbol{\nu}^T\mathbf{f}_{uv}). \tag{4}$$

We learn the parameters $\bar{\boldsymbol{\eta}}$ and $\boldsymbol{\nu}$, so that we know how much each factor contributes in the diffusion.

Generative process. We summarize the CPD model's generative process below. Denote $\vec{\mathbf{1}}_{\ell\times 1}$ as an all-one vector of length ℓ.

1. For each topic $z \in Z$, draw its word distribution from a Dirichlet prior parameterized by β: $\boldsymbol{\phi}_z|\beta \sim Dir(\beta\vec{\mathbf{1}}_{|W|\times 1})$;
2. For each community $c \in C$, draw its topic distribution from a Dirichlet prior parameterized by α: $\boldsymbol{\theta}_c|\alpha \sim Dir(\alpha\vec{\mathbf{1}}_{|Z|\times 1})$;
3. For each user $u \in U$
   (a) Draw her community distribution $\boldsymbol{\pi}_u|\rho \sim Dir(\rho\vec{\mathbf{1}}_{|C|\times 1})$;
   (b) For the i-th document d_{ui} of user u
      i. Draw a community assignment $c_{ui}|\boldsymbol{\pi} \sim Multi(\boldsymbol{\pi}_u)$, by u's multinomial community distribution $\boldsymbol{\pi}_u$;
      ii. Draw a topic $z_{ui}|\mathbf{c},\boldsymbol{\theta} \sim Multi(\boldsymbol{\theta}_{c_{ui}})$, by c_{ui}'s multinomial topic distribution $\boldsymbol{\theta}_c$;
      iii. Draw each word $w_{uik}|\mathbf{z},\boldsymbol{\phi} \sim Multi(\boldsymbol{\phi}_{z_{ui}})$, $\forall k = 1,...,|W_{ui}|$, by z_{ui}'s multinomial word distribution;
   (c) For each friendship link from user u to v, draw $F_{uv}|\boldsymbol{\pi} \sim Ber(\sigma(\hat{\boldsymbol{\pi}}_u^T\hat{\boldsymbol{\pi}}_v))$ by a Bernolli distribution (Eq. 2);
   (d) For each diffusion link E_{ij}^t from document d_{ui} to document d_{vj} at time t, draw $E_{ij}^t|C,\boldsymbol{\eta},Z,\boldsymbol{\nu},\mathbf{f} \sim Ber(\sigma(\bar{\mathbf{c}}_{ij}^T\bar{\boldsymbol{\eta}}+n_z^t+\boldsymbol{\nu}^T\mathbf{f}_{uv}))$ by a Bernolli distribution (Eq. 4);

4. SCALABLE MODEL INFERENCE

We develop a scalable inference algorithm for CPD. We aim to infer the topic assignment and community assignment latent variables $\{\mathbf{Z},\mathbf{C}\}$ from the observations $\{W,F,E\}$, where W is the words in D. We use collapsed Gibbs sampling [5, 13, 27] for the inference. We also estimate the variational parameters $\{\boldsymbol{\pi},\boldsymbol{\theta},\boldsymbol{\phi}\}$ and the model parameters $\{\boldsymbol{\nu},\boldsymbol{\eta}\}$ by variational Expectation Maximization (EM) [4, 5]. We later parallelize our inference algorithm.

4.1 Collapsed Gibbs Sampling

To derive the Gibbs sampler, we start with computing the collapsed posterior distribution of our model:

$$\begin{aligned} &p(W,F,E,C,Z,\mathbf{f},\boldsymbol{\nu},\bar{\boldsymbol{\eta}}|\rho,\alpha,\beta) \\ &=p(C|\rho)p(Z|C,\alpha)p(W|Z,\beta)p(F|C)p(E|C,\bar{\boldsymbol{\eta}},Z,\boldsymbol{\nu},\mathbf{f}), \end{aligned} \tag{5}$$

where $p(F|C)$ (abbreviated as $p(F)$ in the following) is the probability for the friendship links F generated by the communities C; $p(E|C,\bar{\boldsymbol{\eta}},Z,\boldsymbol{\nu},\mathbf{f})$ (abbreviated as $p(E)$) is the probability for the diffusion links E generated by the communities C. We follow [4] to model observed links only in Eq. 5; i.e., we define $p(F)=\prod_{(u,v)\in F}P(F_{uv}=1)$ (Eq. 2) and $p(E)=\prod_{(i,j)\in E}p(E_{ij}^t=1)$ (Eq. 4), where t is the timestamp of the diffusion link (i,j).

In the generative process of CPD model, both $P(F_{uv}=1)$ (step 3.c) and $p(E^t_{ij}=1)$ (steps 3.d) are modeled with sigmoid functions $\sigma(\cdot)$. Bayesian inference with sigmoid function is known as hard, because it is analytically inconvenient to construct a Gibbs sampler for the sigmoid function [24]. We are motivated by the data augmentation approach [5], which introduces Pólya-Gamma random variables to derive an exact mixture representation of the sigmoid function for easier inference. Hence we introduce two Pólya-Gamma variables $\boldsymbol{\lambda}$ and $\boldsymbol{\delta}$ for $p(F)$ and $p(E)$ respectively. Define $\psi(a,b)=e^{\frac{a-ba^2}{2}}$. After some derivations [3], we obtain the augmented probabilities for F and E as

$$p(F,\boldsymbol{\lambda})=\prod\nolimits_{(u,v)\in F}\psi(\hat{\boldsymbol{\pi}}_u^T\hat{\boldsymbol{\pi}}_v,\lambda_{uv})p(\lambda_{uv}|1,0), \tag{6}$$

$$p(E,\boldsymbol{\delta})=\prod_{(i,j)\in E}\psi(\bar{\mathbf{c}}_{ij}^T\bar{\boldsymbol{\eta}}+n_z^t+\boldsymbol{\nu}^T\mathbf{f}_{uv},\delta_{ij})p(\delta_{ij}|1,0). \tag{7}$$

Finally we infer $\mathbf{Z}$ and $\mathbf{C}$, together with $\boldsymbol{\lambda}$ and $\boldsymbol{\delta}$. We next summarize how we sample these variables (details are in [3]).

- **For $\mathbf{Z}$**: the probability of assigning topic z to d_{ui} is

$$\begin{aligned}&p(z_{ui}=z|C,Z_{\neg\{ui\}},W,F,E,\mathbf{f},\boldsymbol{\nu},\boldsymbol{\eta},\boldsymbol{\lambda},\boldsymbol{\delta})\\ &\propto\frac{n^z_{c,\neg\{ui\}}+\alpha}{n^{(\cdot)}_{c,\neg\{ui\}}+|Z|\alpha}\frac{\prod_{w=1}^{|W|}\prod_{i=1}^{n^w_{ui}}(n^w_{z,\neg\{ui\}}+\beta+i-1)}{\prod_{j=1}^{n^{(\cdot)}_{ui}}(n^{(\cdot)}_{z,\neg\{ui\}}+|W|\beta+j-1)}\\ &\quad\times\prod\nolimits_{v\in\Lambda_u}\psi(\hat{\boldsymbol{\pi}}_u^T\hat{\boldsymbol{\pi}}_v,\lambda_{uv}|C_{\neg\{ui\}})\\ &\quad\times\prod\nolimits_{j\in\Lambda_i}\psi(\bar{\mathbf{c}}_{ij}^T\bar{\boldsymbol{\eta}}+n_z^t+\boldsymbol{\nu}^T\mathbf{f}_{uv},\delta_{ij}|C_{\neg\{ui\}},Z_{\neg\{ui\}}),\end{aligned} \tag{8}$$

where $\Lambda_u=\{v|(u,v)\in F \text{ or } (v,u)\in F\}$ is user u's neighbors in F. $\Lambda_i=\{j|(i,j)\in E \text{ or } (j,i)\in E\}$ is document i's neighbors in E. $n^z_{c,\neg\{ui\}}$ and $n^{(\cdot)}_{c,\neg\{ui\}}$ denote the number of times that topic z is assigned to community c and that any topic is assigned to c, excluding document d_{ui}. Similarly, $n^w_{z,\neg\{ui\}}$ and $n^{(\cdot)}_{z,\neg\{ui\}}$ are the count that word w is assigned to topic z and that any word is assigned to z, excluding d_{ui}. n^w_{ui} and $n^{(\cdot)}_{ui}$ are the number of times that word w occurs in d_{ui} and the number of words in d_{ui}.

- **For $\mathbf{C}$**: the probability of assigning community c to u at d_{ui} is

$$\begin{aligned}&p(c_{ui}=c|C_{\neg\{ui\}},Z,W,F,E,\mathbf{f},\boldsymbol{\nu},\bar{\boldsymbol{\eta}},\boldsymbol{\lambda},\boldsymbol{\delta})\\ &=\frac{n^c_{u,\neg\{ui\}}+\rho}{n^{(\cdot)}_{u,\neg\{ui\}}+|C|\rho}\frac{n^z_{c,\neg\{ui\}}+\alpha}{n^{(\cdot)}_{c,\neg\{ui\}}+|Z|\alpha}\prod\nolimits_{v\in\Lambda_u}\psi(\hat{\boldsymbol{\pi}}_u^T\hat{\boldsymbol{\pi}}_v,\lambda_{uv}|C_{\neg\{ui\}})\\ &\quad\times\prod\nolimits_{j\in\Lambda_i}\psi(\bar{\mathbf{c}}_{ij}^T\bar{\boldsymbol{\eta}}+n_z^t+\boldsymbol{\nu}^T\mathbf{f}_{uv},\delta_{ij}|C_{\neg\{ui\}},Z_{\neg\{ui\}}),\end{aligned} \tag{9}$$

where $n^c_{u,\neg\{ui\}}$ and $n^{(\cdot)}_{u,\neg\{ui\}}$ are the number of user u's documents assigned to c and u's total document count, both excluding d_{ui}.

- **For $\boldsymbol{\lambda}$**: the conditional distribution of $\boldsymbol{\lambda}$ is Pólya-Gamma, i.e.,

$$\begin{aligned}&p(\lambda_{uv}|W,F,E,C,Z,\mathbf{f},\boldsymbol{\nu},\boldsymbol{\eta},\boldsymbol{\delta})\\ &\propto e^{\frac{-\lambda_{uv}(\hat{\boldsymbol{\pi}}_u^T\hat{\boldsymbol{\pi}}_v)^2}{2}}p(\lambda_{uv}|1,0)=PG(1,\hat{\boldsymbol{\pi}}_u^T\hat{\boldsymbol{\pi}}_v).\end{aligned} \tag{10}$$

We efficiently sample λ_{uv} by an alternate exponentially tilted Jacobi distribution [24].

- **For $\boldsymbol{\delta}$**: the conditional distribution of $\boldsymbol{\delta}$ is also Pólya-Gamma,

$$\begin{aligned}&p(\delta_{ij}|W,F,E,C,Z,\mathbf{f},\boldsymbol{\nu},\boldsymbol{\eta},\boldsymbol{\delta})\\ &\propto e^{\frac{-\delta_{ij}(\bar{\mathbf{c}}_{ij}^T\bar{\boldsymbol{\eta}}+n_z^t+\boldsymbol{\nu}^T\mathbf{f}_{uv})^2}{2}}p(\delta_{ij}|1,0)\\ &=PG(1,\bar{\mathbf{c}}_{ij}^T\bar{\boldsymbol{\eta}}+n_z^t+\boldsymbol{\nu}^T\mathbf{f}_{uv}).\end{aligned} \tag{11}$$

4.2 Model Parameter Estimation

We use variational EM to iteratively estimate the variational parameters $\{\boldsymbol{\pi},\boldsymbol{\theta},\boldsymbol{\phi}\}$ and the model parameters $\{\boldsymbol{\nu},\boldsymbol{\eta}\}$:

Algorithm 1 Scalable inference for CPD

```
Input: Users $U$, docs $D$, friendship links $F$, diffusion links $E$;
Output: Topic assignments $Z$, community assignments $C$, model
    parameters $\boldsymbol{\nu}$ and $\boldsymbol{\eta}$;
 1: Initialize $\boldsymbol{\nu}$, $\boldsymbol{\eta}$, $\alpha$, $\beta$, $\rho$;
 2: for iter = 1:$T_1$ do
 3:   for each user $u \in U$ do
 4:     for each document $d_{ui} \in D_u$ do
 5:       Sample a topic label $z_{ui}$ according to Eq. 8;
 6:       Sample a community label $c_{ui}$ according to Eq. 9;
 7:   for each friendship link $(u,v) \in F$ do
 8:     Sample augmented variable $\lambda_{uv}$ according to Eq. 10;
 9:   for each diffusion link $(i,j) \in E$ do
10:     Sample augmented variable $\delta_{ij}$ according to Eq. 11;
11:   for each diffusion link $(i,j) \in E$ do
12:     Update $\eta_{c_{ui},c_{vj}z_{ui}}$ by aggregating $c_{ui}$'s, $c_{vj}$'s and $z_{ui}$'s;
13:   for subiter = 1:$T_2$ do
14:     Gradient descent for $\boldsymbol{\nu}$ over the diffusion links $E$.
```

1. (E-step) Use the samples of collapsed Gibbs sampling to estimate the parameters $\boldsymbol{\pi}$, $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, given $\boldsymbol{\nu}$, $\boldsymbol{\eta}$.

2. (M-step) Optimize $\boldsymbol{\nu}$ and $\boldsymbol{\eta}$ by maximizing Eq. 5, given the parameters $\boldsymbol{\pi}$, $\boldsymbol{\theta}$, $\boldsymbol{\phi}$ estimated in the E-step.

In the E-step, the Gibbs sampler iteratively draws samples of $\mathbf{Z}$, $\mathbf{C}$, $\boldsymbol{\lambda}$ and $\boldsymbol{\delta}$ by Eqs. 8–11. Based on the samples, we estimate: $\pi_{u,c}=\frac{n_u^c+\rho}{n_u^{(\cdot)}+|C|\rho}$, $\theta_{c,z}=\frac{n_c^z+\alpha}{n_c^{(\cdot)}+|Z|\alpha}$ and $\phi_{z,w}=\frac{n_z^w+\beta}{n_z^{(\cdot)}+|W|\beta}$. In the M-step, we first estimate $\eta_{c,c'z}$'s by aggregating the community and topic assignments w.r.t all the documents, based on the last iteration of sampling. Then we estimate $\boldsymbol{\nu}$ by maximizing Eq. 5 with all other variables fixed– this is essentially fitting a logistic regression function; to solve it, we randomly sample the same amount of non-observed diffusion links as negative instances for optimization. As α and ρ are used to sample the $\boldsymbol{\pi}_u$'s and $\boldsymbol{\theta}_c$'s, we follow the convention [11] to set their values as 50 divided by $\boldsymbol{\pi}_u$'s dimension and $\boldsymbol{\theta}_c$'s dimension respectively, i.e., $\alpha=50/|Z|$, $\rho=50/|C|$. As β is used to sample the word distribution $\boldsymbol{\phi}_z$'s and the number of words is large, we follow [11] again to set $\beta=0.1$.

4.3 Scalability

We summarize our inference algorithm in Alg. 1. In steps 3–10, we take an E-step for collapsed Gibbs sampling. In steps 11-14, we take an M-step for training the model parameters.

Time complexity. In steps 4–6, as we compute the community assignments and topic assignments for each document of each user, it takes $O(|D|\times|C|+|W|\times|Z|)$. In steps 7–8, as we compute $\hat{\boldsymbol{\pi}}_u^T\hat{\boldsymbol{\pi}}_v$ for each friendship link, it takes $O(|C|\times|F|)$. In steps 9–10, as we compute $(\bar{\mathbf{c}}_{ij}^T\bar{\boldsymbol{\eta}}+n_z^t+\boldsymbol{\nu}^T\mathbf{f}_{uv})$ for each diffusion link, it takes $|C|^2\times|E|$. In steps 11–12, as we aggregate the community assignments and topic assignments for each diffusion link, it takes $O(|E|)$. In steps 13–14, as we compute gradients for ν over all the diffusion links, it takes $O(|E|\times T_2)$. In total, for T_1 iterations, the overall complexity is $O((|D|\times|C|+|W|\times|Z|+|C|\times|F|+|C|^2\times|E|+|E|+|E|\times T_2)\times T_1)$. As we can see, Alg. 1's time complexity is linear to the data size (i.e., $|D|$, $|F|$, and $|E|$).

Parallelization. We consider multithread parallelization of Alg. 1 and leave multi-machine parallelization as future work. In our variational EM algorithm, we find the E-step takes much longer time than the M-step, because: 1) the E-step's collapsed Gibbs sampling has to be done iteratively over all the observations, including documents (thus words), friendship links and diffusion links; 2) the M-step's model parameter estimation is comparatively much easier, since optimizing $\boldsymbol{\nu}$ is basically solving logistic regression on the diffusion links (and the same amount of negative links) and $\boldsymbol{\eta}$ is estimated by simply aggregating the community and topic assignments. Thus, in this paper we focus on parallelizing the E-step.

• *Segmenting data to reduce inter-dependency*. Recall in Sect. 4.1, the sampling requires computing: 1) three counters, including the community-topic counter n_c^z, the topic-word counter n_z^w, the user-community counter n_u^c; 2) two link probabilities, including the friendship one $\psi(\hat{\boldsymbol{\pi}}_u^T \hat{\boldsymbol{\pi}}_v, \lambda_{uv})$ and the diffusion one $\psi(\bar{\mathbf{c}}_{ij}^T \bar{\boldsymbol{\eta}} + n_z^t + \boldsymbol{\nu}^T \mathbf{f}_{uv}, \delta_{ij})$. Among these computations, both topic and community assignments are applied to documents (thus their users), the friendship link probability is applied to users, and the diffusion link probability is applied to two documents (thus their users). Therefore, except n_z^w, the vast majority of computations are done on users and documents. This motivates us to segment the data by users and documents, so that different CPU cores can work with little inter-dependency. It may be also possible to consider words for data segment, but it is not obvious and we leave it for future work. Considering that a user often has many documents (especially in Twitter), we design two guidelines to segment the data: 1) we keep a user's documents in the same data segment, because otherwise there are likely many conflicting updates about the same user from multiple threads; 2) we prefer keeping the same-topic documents in the same data segment, because it helps to reduce the conflicting updates about the same topic from multiple threads. Overall, we first run LDA [2] on all users' documents with $|Z|$ topics; then we partition the users into $|Z|$ segments, based on each user's most frequently assigned topic in her documents. In each segment, each user has her documents, related friendship links and diffusion links.

• *Distributing workload to avoid data skewness*. We aim to distribute the $|Z|$ data segments to M threads, such that the workload on each thread is balanced. Note that M is set as the number of physical CPU cores in this work. Our approach is to first estimate the workload of each data segment, and then cast this segment allocation task as solving M standard *0-1 knapsack problems*[2]. Denote the i-th data segment's workload as $o_i \in \mathbb{R}^+$, thus the total workload is $O = \sum_{i=1}^M o_i$. Denote a binary indicator as $x_i \in \{0, 1\}$. Then for each thread, we solve the objective function: $\max \ \sum_{i=1}^M o_i x_i, \quad \text{s.t.} \quad \sum_{i=1}^M o_i x_i \leq \frac{1}{M} O$, which tries to find a subset of the data segments to have as close to $\frac{O}{M}$ workload as possible. One can fine tune this objective function in practice to best allocate data segments for even workload among the threads. We estimate each workload o_i as follows. First, we estimate the average processing time for each document and link, based on a serial implementation of the sampling algorithm over all data. Then, based on the number of documents and links a user has, we estimate the average workload of processing that user. Finally, we sum up the average workload of all users in the i-th data segment as o_i.

5. APPLICATIONS

We concretize how to enable the following three applications based on five CPD outputs, including: 1) the community assignment for users $\pi_{u,c}$'s; 2) the community content profile $\theta_{c,z}$'s; 3) the community diffusion profile $\eta_{c,c'z}$'s; 4) words' topic assignment $\phi_{w,z}$'s; 5) the individual diffusion preference parameter $\boldsymbol{\nu}$.

Community-aware diffusion. Given input of a document d_{vj} published by user v, we output the probability that another user u will publish a document d_{ui} to retweet or cite d_{vj} at timestamp t as

$$p(E_{ij}^t = 1 | u, v, d_{vj}, t) \overset{1}{=} \textstyle\sum_z p(E_{ij}^t = 1 | u, v, z, t) p(z | d_{vj})$$
$$\overset{2}{=} \sum_z \sigma(\sum_c \sum_{c'} \pi_{u,c} \theta_{c,z} \eta_{c,c'z} \pi_{v,c'} \theta_{c',z} + n_z^t + \boldsymbol{\nu}^T \mathbf{f}_{uv}) p(z | d_{vj}). \tag{12}$$

where at step 1 we expand $p(E_{ij}^t = 1 | u, v, d_{vj}, t)$ by the topics of d_{vj}. At step 2, we plug in the definition of $p(E_{ij}^t = 1 | u, v, z, t)$ by Eq. 4. As we can see, Eq. 12 comprehensively models the diffusion by taking the community assignments $\boldsymbol{\pi}$, the community profiles $\boldsymbol{\theta}$ and $\boldsymbol{\eta}$, and the individual diffusion preference $\boldsymbol{\nu}$ into account.

Profile-driven community ranking. Given input of a query $q \in W^k$ ($\forall k \geq 1$), we output the ranking of communities based on their probabilities to diffuse information about q. Denote the probability of a community c to generate a diffusion link $s = 1$ of query q as

$$p(s = 1 | c, q) \overset{1}{=} \textstyle\sum_z \sum_{c'} p(s = 1 | c, c', z) p(z | q, c') p(c' | q)$$
$$\overset{2}{\propto} \textstyle\sum_z \sum_{c'} \eta_{c,c'z} p(z | q, c') \overset{3}{\propto} \sum_z \sum_{c'} \eta_{c,c'z} \theta_{c'z} \prod_{w \in q} \phi_{z,w}, \tag{13}$$

where at step 1 we expand $p(s = 1 | c, q)$ by the community diffusion profile $p(s = 1 | c, c', z)$, the topic assignment for q in a community $p(z | q, c')$ and the probability that q is from that community $p(c' | q)$. At step 2 we plug in the definition of $p(s = 1 | c, c', z) \propto \eta_{c,c'z}$ and consider q can come from any community with $p(c' | q)$ uniformly. At step 3, we estimate the probability $p(z | q, c')$ in a similar way as Eq. 8. We skip the details but explain the rational of this estimation: $p(z | q, c')$ is proportional to the probability of community c' generating topic z (i.e., captured by $\theta_{c',z}$) and the probability of q belonging to topic z (i.e., captured by $\prod_{w \in q} \phi_{z,w}$).

Profile-driven community visualization. We can visualize each community's content profile and its diffusion profile, as Fig. 1(b) shows. In particular, we are interested in the diffusion visualization, as it is new. In our experiments, we visualize how a community interacts with the others in two typical settings: 1) *diffusion on a specific topic*, where we use $\eta_{c,c'z}$ as the diffusion strength from c to c' under topic z; 2) *diffusion with topic aggregation*, where we use $\sum_z \eta_{c,c'z}$ as the diffusion strength from c to c'.

6. EXPERIMENTS

We test CPD with two large-scale real-world data sets. We design experiments to: 1) evaluate how well we address each challenge listed in Sect. 1; 2) evaluate CPD's performance, by comparing with the state-of-the-art baselines in different applications.

6.1 Set Up

All experiments are conducted on Linux computers equipped with Intel(R) 3.50GHz CPUs and 16GB RAMs. We do 10-fold cross validation. We report average scores for all experiments and significant results when necessary. Our code is available online[3].

Data sets. We use two public data sets: Twitter [18] and DBLP [29]. The Twitter data set was collected in May 2011. The DBLP data set contains the publications indexed by DBLP[4] from 1936 to 2010. All tweets and paper titles are pre-processed by removing stop words, stemming and POS tagging[5], and only nouns, verbs and hashtags are kept. After that, we removed the documents with less than two words, and then removed the users with no document. Table 2 summarizes the statistics of our data sets after pre-processing.

[2] https://en.wikipedia.org/wiki/Knapsack_problem

[3] https://bitbucket.org/vwz/vldb2017-cpd

[4] http://dblp.uni-trier.de/

[5] http://nlp.stanford.edu/software/tagger.shtml

Table 2: Data set statistics.

	#(user)	#(friend. link)	#(diff. link)	#(doc.)	#(word)
Twitter	137,325	3,589,811	992,522	39,952,379	2,316,020
DBLP	916,907	3,063,186	10,210,652	4,121,213	330,334

Table 3: Differences with baselines.

Methods	**Data**			**Diffusion factors**			**Tasks**			
	text	friend link	diff link	indi	comm	topic	topic extra	comm detect	diff pred	comm profile
PMTLM [38]	●		●			●	●		●	
WTM [32]	●	●	●	●					●	
CRM [13]		●	●	●	●			●	●	
COLD [15]	●		●		●		●	●	●	
Ours	●	●	●	●	●	●	●	●	●	●

Baselines. We choose baselines based on three guidelines: 1) they are the state of the art to model heterogeneous user observations at the data level; 2) they model diffusion prediction at the task level; 3) preferably they model community. Finally, we choose four baselines below, and list our differences with them in Table 3.

• *Poisson Mixed-Topic Link Model (PMTLM)* [38]. It models document network based on document's topic assignment. We adapt PMTLM for community detection and friendship link prediction comparison, by aggregating topic allocations of a user's documents as her community membership. We also compare with PMTLM on diffusion prediction, as it also models the document links.

• *Whom to Mention (WTM)* [32]. It models the user diffusion links with user content and friendship. It does not model community. We compare with WTM on diffusion prediction.

• *Community Role Model (CRM)* [13]. It models friendship links and diffusion links based on the user's community assignment and role assignment together. We compare with CRM on community detection, friendship link prediction and diffusion prediction.

• *COmmunity Level Diffusion (COLD)* [15]. It models the content and diffusion links based on communities. Thus it is the closest work to ours. But it models neither friendship links in community detection, nor non-community factor in diffusion prediction. We compare with COLD on community detection, friendship link prediction and diffusion prediction. As COLD has community diffusion strength, we also compare it on community ranking.

In addition to the above *existing* baselines, we design more baselines to validate our advantages over a straightforward community profiling approach of "first detecting communities, then aggregating each community's user observations". Specifically, we adopt two state-of-the-art algorithms, CRM [13] and COLD [15], to detect communities. Denote the derived probabilities of user u belonging to a community c as $\pi^*_{u,c}$. We then aggregate the user data in each community as its profiles. To get aggregated content profile, we first run LDA [2] on all documents with $|Z|$ topics, and for each document d_{ui}, we get its $|Z|$-dimensional multinomial topic distribution as $\boldsymbol{\theta}^*_{d_{u,i}}$. Denote community c's aggregated content profile as $\boldsymbol{\theta}^*_c$. We have $\boldsymbol{\theta}^*_c = \sum_{u=1}^{|U|} \pi^*_{u,c} \sum_{i=1}^{|D_u|} \frac{\boldsymbol{\theta}^*_{d_{u,i}}}{|D_u|}$. To get aggregated diffusion profile, we aggregate each diffusion link between $d_{u,i}$ and $d_{v,j}$ in E w.r.t. their users' communities on a topic z. Denote the aggregated diffusion profile $\eta^*_{c,c'z}$ as the probability of community c diffusing community c' on z. We have $\eta^*_{c,c'z} \propto \sum_{(i,j)\in E} \pi^*_{u,c}\pi^*_{v,c'}\theta^*_{d_{u,i},z}\theta^*_{d_{v,j},z}$. In all, we obtain the following two more baselines, which implement the straightforward "first detection, then aggregation" profiling approach.

• *CRM+Agg*. It uses CRM [13] to detect communities; then it uses the above two equations to calculate the aggregated community content profiles $\boldsymbol{\theta}^*_c$ and diffusion profiles $\eta^*_{c,c'z}$.

• *COLD+Agg*. It detect communities by COLD [15], then similarly uses the above two equations to get content and diffusion profiles.

We compare with both CRM+Agg and COLD+Agg in diffusion link prediction and community ranking.

Evaluation. Since we jointly profile and detect communities, we will evaluate the quality of both community detection and profiles.

• *Detection quality*. We consider two ways to evaluate the resulting communities: 1) how dense they are; 2) how well they can be used to explain the friendship link observations. For 1), we use *conductance* [16, 17] as the metric. As our community assignment is probabilistic, we follow [15] to let each user belong to her top five communities in conductance evaluation (and also later community ranking evaluation). The smaller conductance is, the better. For 2), we follow [15] to design a *link prediction* task, where we use Eq. 2 to predict whether a friendship link is observed based on two users' communities. As there is no predefined threshold for link prediction, we use *AUC* (Area Under the receiver operating characteristic Curve) [15, 13] as the metric. In the 10-fold cross validation, each time we use 10% of the positive links and sample the same amount of negative links to calculate AUC. The higher AUC is, the better.

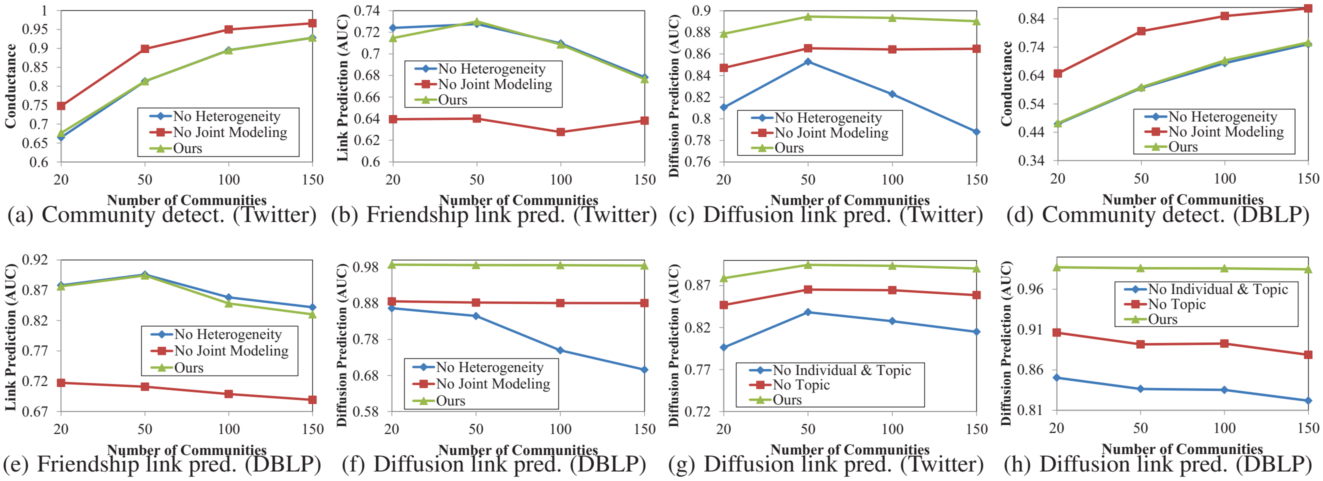
• *Profile quality*. Due to the lack of ground truth, we evaluate the community profiles' quality through the applications in Sect. 5. For community-aware diffusion, we again use *AUC* as the evaluation metric because there is no predefined threshold. For profile-driven community ranking, as the communities detected by different algorithms are not directly comparable, we evaluate its performance in terms of each community's users– given a query q, we check how many users in the top K communities (denoted as C_{topK}) truly retweet (or cite) about q. Then naturally we compute *precision* and *recall*. Denote the users who mention q in their retweets (or citation paper titles) as U^*_q. Denote the users belonging to any community of C_{topK} as U_K. The precision of C_{topK} for q is $P(K,q) = |U^*_q \cap U_K|/|U_K|$, and the recall is $R(K,q) = |U^*_q \cap U_K|/|U^*_q|$. We define *mean average precision* (MAP) over all the queries (Q) as $MAP@K = \sum_q(\sum_{i=1}^K P(i,q)/K)/|Q|$ and *mean average recall* (MAR) as $MAR@K = \sum_q(\sum_{i=1}^K R(i,q)/K)/|Q|$. Finally, *mean average F1* is $MAF@K = \frac{2\times MAP@K\times MAR@K}{MAP@K+MAR@K}$. The higher MAF is, the better. In addition, as the content profile is based on topics, we adopt one extra widely used metric (*perplexity*) in topic modeling [2] to evaluate its quality, which measures how well it generates the user content observations, and we calculate perplexity the same as in [15]. The lower perplexity is, the better.

6.2 Model Design

We want to evaluate how well we address each community profiling challenge as introduced in Sect. 1. To achieve this goal, we design some baselines based on the degenerated versions of CPD, for validating the advantages of our model design. We compare CPD with these baselines, and evaluate the quality of detected communities and profiles through three tasks: community detection, friendship link prediction and diffusion link prediction.

• *Modeling the inter-dependency with community detection*. We design a baseline "no joint modeling", where we first detect communities only from the friendship links through a generative model by Eq. 2, then we extract the profiles through a generative model as in CPD except having the communities fixed. As shown in Fig. 3(a)–3(f), ours is always better than "no joint modeling".

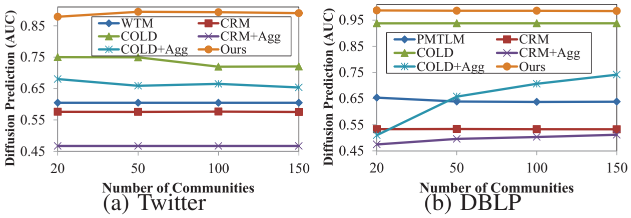
• *Addressing the heterogeneity of social observations*. We design a



(a) Community detect. (Twitter) (b) Friendship link pred. (Twitter) (c) Diffusion link pred. (Twitter) (d) Community detect. (DBLP)

(e) Friendship link pred. (DBLP) (f) Diffusion link pred. (DBLP) (g) Diffusion link pred. (Twitter) (h) Diffusion link pred. (DBLP)

Figure 3: Study of our model design.



(a) Twitter (b) DBLP

Figure 4: Results of community-aware diffusion.

baseline “no heterogeneity”, where we adapt CPD to model friendship and diffusion links identically by Eq. 2, but keep the other parts of CPD modeling unchanged. As shown in Fig. 3(a) - 3(f), ours is better than “no heterogeneity” on diffusion prediction, and comparable with it on the other two tasks. This implies: 1) diffusion links and friendship links are different, and diffusion links require more sophisticated modeling than friendship links; 2) the two types of links are correlated; diffusion links do not significantly change the community structure once the friendship links are given.

- *Accommodate the nonconformity of user behaviors*. We design two baselines: 1) “no individual & topic”, where we exclude the individual factor and topic factor from Eq. 4 in CPD; 2) “no topic”, where we exclude only the topic factor from Eq. 4 in CPD. As shown in Fig. 3(g) and 3(h), the individual factor contributes 4.8% and 6.8% absolute AUC improvement on Twitter and DBLP respectively; the topic factor contributes another 3.6% and 10.5% absolute AUC improvement on each data set.

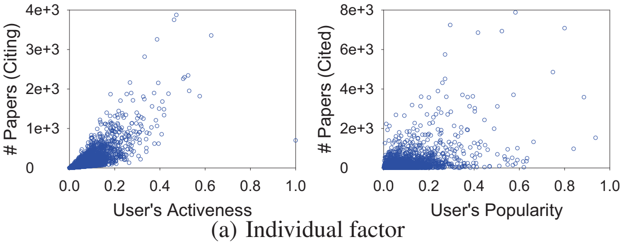
In all, we conclude that our model design well addresses the three challenges in community profiling.

6.3 Comparison with Baselines

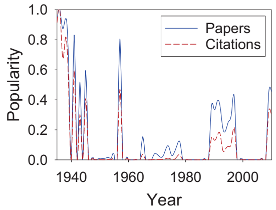
We evaluate CPD and the baselines on various applications.

6.3.1 *Community-aware Diffusion*

Quantitative analysis. In Fig. 4, we summarize the result comparison with the baselines introduced in Sect. 6.1. PMLTM is not applicable to Twitter, since it is designed solely for citation network–it predicts a citation based on the similarity between two documents, but in Twitter a tweet and its retweet are almost identical. As



(a) Individual factor



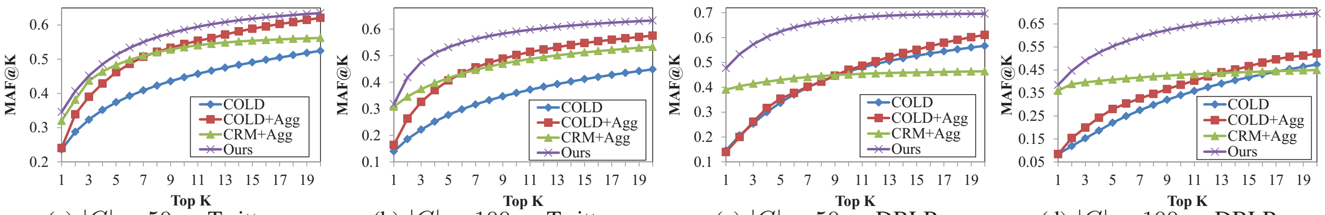
Top Five Topics That c_{18} Cites c_{32}		Top Five Topics That c_{32} Cites c_{18}	
Topic	*Diffusion Strength*	*Topic*	*Diffusion Strength*
T_{22}	0.0383	T_{22}	0.0225
T_8	0.0014	T_{47}	0.0006
T_{49}	0.0005	T_8	0.0001
T_{47}	0.0004	T_9	0.00002
T_{39}	0.0001	T_{39}	0.000007

(b) Topic factor (c) Community factor

Figure 5: Community-aware diffusion case study.

shown in Fig. 4, our model consistently outperforms all the baselines, thanks to: 1) our modeling various diffusion factors and heterogeneous user links, in contrast with the baselines in Table 3; 2) our joint detection and profiling, in contrast with the two “first detection then aggregation” baselines. When $|C| = 100$, we achieve 24.2%–91.6% and 5.1%–108.0% relative AUC improvements than the baselines in Twitter and DBLP, respectively. The improvements are statistically significant over the 10-fold cross validation results, with student’s t-test one-tailed p-value $p < 0.01$.

Case study. We examine the three diffusion factors in Eq. 4 with the DBLP data. Firstly, in Fig. 5(a) we plot the number of papers a user cites w.r.t. her *activeness*, and the number of citations a user has w.r.t. her *popularity*. User activeness and popularity are defined in Sect. 3.1. Generally, the more active a user is (i.e., publishing more papers), the more papers she cites; besides, the more popular a user is (i.e., a more established researcher), the more citations her papers get. This observation supports our design of modeling both user activeness and popularity as the individual factors in diffusion.



(a) $|C| = 50$ on Twitter (b) $|C| = 100$ on Twitter (c) $|C| = 50$ on DBLP (d) $|C| = 100$ on DBLP

Figure 6: Results of profile-driven community ranking.

Table 4: Top four words in each topic.

Topic	Word Distribution (listed by "word:probability")
T_{22}	network:0.059, wireless:0.050, sensor:0.046, routing:0.038
T_{49}	network:0.042, performance:0.037, traffic:0.031, routing:0.028
T_{47}	service:0.056, web:0.028, mobile:0.025, management:0.024
T_8	security:0.031, key:0.028, authentication:0.027, protocol:0.020
T_9	code:0.061, algorithm:0.032, function:0.028, linear:0.027
T_0	design:0.049, circuit:0.034, power:0.027, cmos:0.017
T_{44}	parallel:0.05, performance:0.036, memory:0.03, architecture:0.02
T_{46}	analysis:0.061, reliability:0.029, optical:0.024, design:0.021

Table 5: Top three communities ranked for query "router".

K	AP@K	AR@K	AF@K	Topic Distribution
1	0.919	0.327	0.483	T_{22}:0.976, T_{49}:0.013, T_{47}:0.006
2	0.900	0.424	0.576	T_8:0.988, T_{22}:0.004, T_9:0.003
3	0.891	0.528	0.663	T_0:0.977, T_{44}:0.008, T_{46}:0.005

In Fig. 5(b), we plot the number of papers and citations w.r.t. a specific topic ("parallel performance memory") over the years. As we can see, there is a high correlation between the number of papers and that of citations over time– if a topic is popular (i.e., it has many papers), then it is more likely to be cited (i.e., it appears in many citations). This observation supports our design of modeling the topic factor in Sect. 3.1.

Finally, in Fig. 5(c) we list the diffusion between two example communities : c_{18} and c_{32} (i.e., the top 2 communities for query "router" in profile-drive community ranking (Sect. 6.3.2)). As we can see, c_{18} and c_{32} tend to cite each other on topic T_{22} ("network" as shown in Table 4). Besides, c_{18} tends to cite c_{32} on T_8 ("security"), whereas c_{32} tends to cite c_{18} on T_{47} ("service"). This shows that each community has a preference to diffuse others on certain topics. Thus it is necessary to model community factor in diffusion.

6.3.2 *Profile-driven Community Ranking*

For community ranking, we follow several guidelines to choose queries: 1) it should be easy to assess whether a retweet or a citation contains a query, thus we choose single terms (i.e., either hashtags or words) as queries; 2) a query has to be meaningful– since words are noisy, we choose hashtags as queries in Twitter; DBLP has no hashtag, thus we choose words as queries, but we remove the top 1,000 frequent words; 3) a query has to appear with sufficient frequency in retweets or citations, thus we choose hashtags (Twitter) and words (DBLP) with frequency larger than 100. In the end, we get 5,680 queries in Twitter and 27,479 queries in DBLP.

Given each query q, we rank the detected communities by Eq. 13, and then return the top K communities(for $K = 1, ..., 20$).

Quantitative analysis. Fig. 6 compares CPD with the baselines that support community-level content and diffusion modeling, including COLD, COLD+Agg and CRM+Agg. As we can see, CPD consistently outperforms all the baselines; when $|C| = 100$ and $K = 5$, we achieve 27.6%–92.0% (Twitter) and 35.4%–150.8% (DBLP) relatively MAF improvements than the baselines. All improvements are statistically significant over the 10-fold cross validation results, with student's t-test one-tailed p-value $p < 0.01$. Note that CPD is better than COLD+Agg and CRM+Agg, again showing the advantage of joint detection and profiling. Besides, we observe that CPD's MAF@K starts to converge earlier than others. It means we find more relevant users in the top K communities.

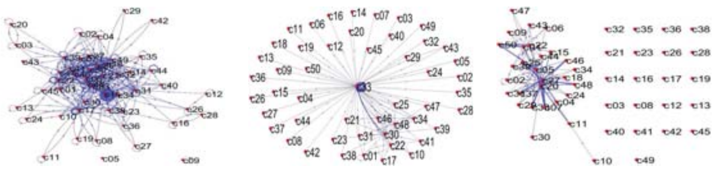
We further tested community ranking with different subsets of queries. We divided the queries by their occurrence frequency in the corpus. We equally split the whole frequency range into five intervals and tested community ranking with the subset of queries whose frequency falls within each interval. We observed: 1) the results show similar trends that CPD consistently outperforms the baselines; 2) the absolute MAF@K values are not sensitive to different query subsets. Due to space limit, we skip these plots.

Case study. We further examine the communities ranked by CPD for a specific query. Table 5 lists the top three communities that are most likely to cite papers about "router". AP@K is the average precision of the ranked top K communities; similarly, AR@K is the average recall and AF@K is the average F1. AF@K increases as K increases, which is consistent with the trend observed in Fig. 6. Besides, according to Table 4, the top 3 communities to cite "router" are all reasonably the networking communities: "network wireless sensor", "security key authentication" and "circuits design".

6.3.3 *Profile-driven Community Visualization*

In Fig. 7, we visualize the DBLP community diffusion under three cases: the aggregation of all topics, a general topic and a specialized topic, respectively. Denote the detected 50 communities as c_{01}–c_{50}. For each directed edge between two communities c and c', the width indicates the diffusion strength. In Fig. 7(a), the strength is an aggregated value over all topics ($\sum_z \eta_{cc'z}$); in Fig. 7(b) and 7(c) the strength is $\eta_{c,c'z}$ for two typical types of topics. We skip the edges with strengths below average for simpler visualization.

We can make several interesting observations from Fig. 7. Firstly, Fig. 7(a) shows that under topic aggregation, the communities often diffuse a lot within themselves. This coincides with our definition of "community" as a group of users who share similar diffusion behavior– in this case, the same community users often diffuse information to each other. Secondly, Fig. 7(a) shows that some communities are more "open" than the others. E.g., c_{48} ("data database search") and c_{33} ("web information analysis") are more open research communities, which diffuse information from a majority of communities. In contrast, c_{09} ("neural control system") appears as a more closed research community, which hardly diffuses information with other communities. Such a visualization enables us to assess the openness of a research community. Finally, we find that, the diffusion behaviors vary w.r.t. different kinds of topics. E.g.,

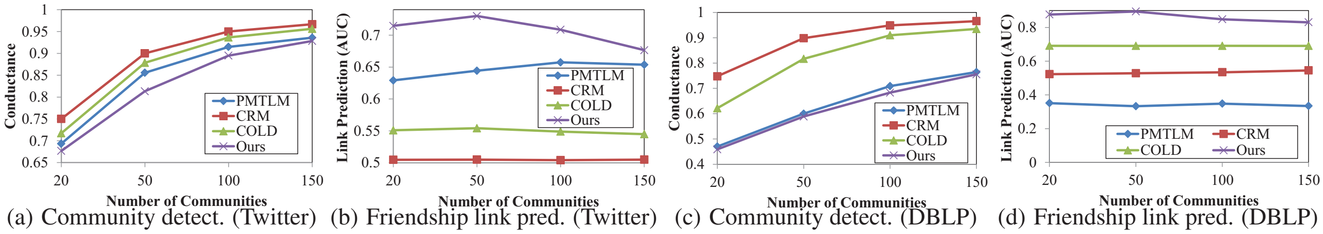


(a) Aggregated topic (b) A general topic (c) A specialized topic

Figure 7: Results of profile-drive community visualization of diffusion on

Dataset	Algorithms	Number of Communities			
		20	50	100	150
Twitter	COLD+Agg	825943.4	694741.7	515997.0	427181.1
	CRM+Agg	826737.5	695400.6	516730.9	427761.2
	Ours	5117.0	3992.1	3801.4	3849.7
DBLP	COLD+Agg	61348.6	47179.0	39983.4	36922.7
	CRM+Agg	61901.9	47480.5	41289.1	37018.3
	Ours	1153.0	982.0	875.0	885.0

Figure 8: Perplexity comparison.



(a) Community detect. (Twitter) (b) Friendship link pred. (Twitter) (c) Community detect. (DBLP) (d) Friendship link pred. (DBLP)

Figure 9: Results of community detection.

Fig. 7(b) shows the diffusion on a very general topic ("web, information, search, semantic"), which is discussed and diffused by many research communities. In contrast, Fig. 7(c) shows the diffusion on a very specialized topic ("transmission, gbs, trail, video"), which is of interest to only a few communities such as c_{25} ("distributed performance computing") and c_{27} ("reliability device design"). This visualization reveals the topic generality and is helpful to researchers in choosing research topics.

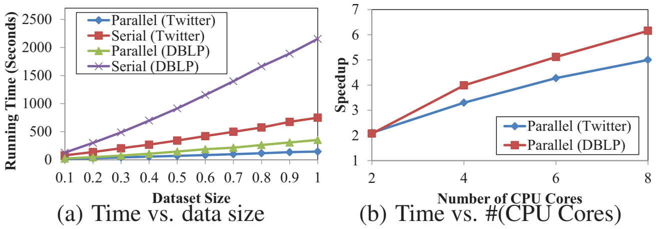
6.3.4 *Quality of Community and Content Profile*

In addition to the three applications, we also conduct experiments to evaluate the quality of communities and content profiles. In Fig. 9, we show our model consistently outperforms the baselines in terms of community quality. Because COLD+Agg and CRM+Agg use the detection of COLD and CRM respectively, we do not include them in comparison again. When $|C| = 100$, we achieve: 1) 2.2%–5.8% (Twitter) and 3.5%–27.8% (DBLP) relative conductance improvements; 2) 7.8%–40.6% (Twitter) and 22.8%–143.5% (DBLP) relative AUC improvements. All the improvements are significant with p-values $p < 0.01$. In general, we are better than COLD and PMLTM, as they do not model the friendship links in community detection; we are better than CRM, as it does not enforce dense friendship links in a community.

In Fig. 8, we compare with COLD+Agg and CRM+Agg in terms of perplexity. Obviously, CPD achieves the lowest perplexity, meaning that our content profiles can best explain the user data. This supports our argument of joint modeling, as motivated in Eq. 1.

6.4 Scalability

In Fig. 10(a), we first show that our training time (per iteration, Alg. 1's steps 3–10) scales linearly to the data size. Each value p (e.g., $p = 0.1$) in the x-axis of Fig. 10(a) indicates that we randomly sample $(p \times 100)$ percents of the total documents, friendship links and diffusion links for experiments. We repeat ten times and report the average training time. We set $|C| = |Z| = 150$. Different $|C|$ and $|Z|$ can change the absolute training time, but they do not change the linearity of our training time to the data size. Moreover, we also show that our multithread parallelization achieves up to 4.5× (in Twitter)and 5.7× (in DBLP) speedup over the serial implementation, by using eight CPU cores.



(a) Time vs. data size (b) Time vs. #(CPU Cores)

Figure 10: Results of scalability.

In Fig. 10(b), we plot the speedup with different numbers of CPU cores in parallelization. Generally, the speedup increases as using more CPU cores. We observe that the speedup for DBLP is bigger than that of Twitter. That maybe because compared with Twitter, DBLP users tend to have less diverse topics. This greatly reduces the inter-dependency between *data segments* (defined in Sect. 4.3). In [3], we also plot the real processing time of each CPU core, and show our parallelization design achieves good workload balancing.

7. CONCLUSION

In this paper, we study a novel problem of community profiling, whose goal is to characterize each community with both internal profile and external profile. The difficulty of community profiling is largely overlooked, which motivates us to propose a CPD model. It identifies and addresses three key challenges, including the inter-dependency with community detection, the heterogeneity of social observations and the nonconformity of user behaviors. We develop a scalable inference algorithm for CPD, which scales linearly with the data size, and further parallelize it with multithreading. We evaluate CPD in terms of the quality of both community detection and profile, and verify that our model design well addresses the three challenges. We further show that CPD outperforms the state-of-the-art baselines in various tasks, including community detection, friendship link prediction, community-aware diffusion, profile-driven community ranking and content profile evaluation. In future, we plan to explore other user information for defining the profiles, such as user attributes, and user sentiments.

Acknowledgment

We thank the support of: National Natural Science Foundation of China (No. 61502418), Zhejiang Provincial Natural Science Foundation (No. LQ14F020002), Research Grant for Human-centered Cyber-physical Systems Programme at Advanced Digital Sciences Center from Singapore A*STAR, and NSF Grant IIS 16-19302. This work is partially supported by DP170103954 and FT130101530. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the funding agencies.

8. REFERENCES

[1] S. Basu Roy, L. V. Lakshmanan, and R. Liu. From group recommendations to group formation. In *SIGMOD*, pages 1603–1616, 2015.

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

[3] H. Cai, V. W. Zheng, F. Zhu, K. C. Chang, and Z. Huang. From community detection to community profiling. *CoRR*, abs/1701.04528:1–14, 2017.

[4] J. Chang and D. M. Blei. Hierarchical relational models for document networks. *ANN APPL STAT*, 4(1):124–150, 2010.

[5] N. Chen, J. Zhu, F. Xia, and B. Zhang. Generalized relational topic models with data augmentation. In *IJCAI*, pages 1273–1279, 2013.

[6] W.-Y. Chen, D. Zhang, and E. Y. Chang. Combinational collaborative filtering for personalized community recommendation. In *KDD*, pages 115–123, 2008.

[7] J. D. Cruz, C. Bothorel, and F. Poulet. Community detection and visualization in social networks: Integrating structural and semantic information. *ACM Trans. Intell. Syst. Technol.*, 5(1):11:1–11:26, 2014.

[8] N. Du, Y. Liang, M. Balcan, and L. Song. Influence function learning in information diffusion networks. In *ICML*, pages 2016–2024, 2014.

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

[10] M. Granovetter. The strength of weak ties: a network theory revisited. *Sociological Theory*, 1(6):211–233, 1983.

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

[12] X. Han, L. Wang, R. Farahbakhsh, A. Cuevas Rumin, R. Cuevas Rumin, N. Crespi, and L. He. CSD: a multi-user similarity metric for community recommendation in online social networks. *Expert Sys. with App.*, 53:14 – 26, 2016.

[13] Y. Han and J. Tang. Probabilistic community and role model for social networks. In *KDD*, pages 407–416, 2015.

[14] L. He, C.-T. Lu, J. Ma, J. Cao, L. Shen, and P. S. Yu. Joint community and structural hole spanner detection via harmonic modularity. In *KDD*, pages 875–884, 2016.

[15] Z. Hu, J. Yao, B. Cui, and E. Xing. Community level diffusion extraction. In *SIGMOD*, pages 1555–1569, 2015.

[16] K. Kloster and D. F. Gleich. Heat kernel based community detection. In *KDD*, pages 1386–1395, 2014.

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

[18] R. Li, S. Wang, H. Deng, R. Wang, and K. C.-C. Chang. Towards social user profiling: unified and discriminative influence model for inferring home locations. In *KDD*, pages 1023–1031, 2012.

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

[20] Y.-R. Lin, J. Sun, P. Castro, R. Konuru, H. Sundaram, and A. Kelliher. Metafac: Community discovery via relational hypergraph factorization. In *KDD*, pages 527–536, 2009.

[21] B. Lucier, J. Oren, and Y. Singer. Influence at scale: Distributed computation of complex contagion in networks. In *KDD*, pages 735–744, 2015.

[22] J. J. McAuley and J. Leskovec. Learning to discover social circles in ego networks. In *NIPS*, pages 548–556, 2012.

[23] E. Ntoutsi, K. Stefanidis, K. Nørvåg, and H.-P. Kriegel. Fast group recommendations by applying user clustering. In *ER*, pages 126–140, 2012.

[24] N. G. Polson, J. G. Scott, and J. Windle. Bayesian inference for logistic models using pólya-gamma latent variables. *J AM STAT ASSOC*, pages 1339–1349, 2013.

[25] I. Ronen, I. Guy, E. Kravi, and M. Barnea. Recommending social media content to community owners. In *SIGIR*, pages 243–252, 2014.

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

[27] M. Sachan, A. Dubey, S. Srivastava, E. P. Xing, and E. Hovy. Spatial compactness meets topical consistency: Jointly modeling links and content for community detection. In *WSDM*, pages 503–512, 2014.

[28] Y. Sun, C. C. Aggarwal, and J. Han. Relation strength-aware clustering of heterogeneous information networks with incomplete attributes. *PVLDB*, 5(5):394–405, Jan. 2012.

[29] J. Tang, A. C. Fong, B. Wang, and J. Zhang. A unified probabilistic framework for name disambiguation in digital library. *IEEE TKDE*, 24(6):975–987, 2012.

[30] L. Tang, X. Wang, and H. Liu. Uncovering groups via heterogeneous interaction analysis. In *ICDM*, pages 503–512, 2009.

[31] P. Wadhwa and M. Bhatia. Community detection approaches in real world networks: A survey and classification. *Int. J. Virtual Communities Soc. Netw.*, 6(1):35–51, 2014.

[32] 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 *WWW*, pages 1331–1340, 2013.

[33] M. Wang, C. Wang, J. X. Yu, and J. Zhang. Community detection in social networks: An in-depth benchmarking study with a procedure-oriented framework. *PVLDB*, 8(10):998–1009, June 2015.

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

[35] Z. Xu, Y. Ke, Y. Wang, H. Cheng, and J. Cheng. A model-based approach to attributed graph clustering. In *SIGMOD*, pages 505–516, 2012.

[36] J. Yang, J. J. McAuley, and J. Leskovec. Community detection in networks with node attributes. In *ICDM*, pages 1151–1156, 2013.

[37] Y. Zhou, H. Cheng, and J. X. Yu. Graph clustering based on structural/attribute similarities. *PVLDB*, 2(1):718–729, 2009.

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