TECHNICAL REPORT

TR-CS-NMSU-2014-1-13

Chuan Hu
Huiping Cao

Department of Computer Science
New Mexico State University

January 13, 2014

Detecting Influence Relationships from Graphs

Chuan Hu[1] Huiping Cao[2]
[1]chu@cs.nmsu.edu, [2]hcao@cs.nmsu.edu
Dept. of Computer Science, New Mexico State University, U.S.A.

Abstract

Graphs have been widely used to represent objects and connections in applications such as the Web, social networks, and citation networks. Mining influence relationships from graphs has gained interests in recent years because providing descriptive influence information about the object connections in graphs can facilitate graph exploration, graph search, and connection recommendations. In this paper, we study the problem of detecting influence aspects, on which objects are connected, and influence degree (or influence strength), with which one graph node influences another graph node on a given aspect.

We propose two generative Aspect Influence Models, OAIM and LAIM, to detect both influence aspects and influence degrees. These models utilize the topological structure of the graphs, the text content associated with objects, and the context in which the objects are connected. We compare these two models with one baseline approach which considers only the text content associated with objects. The empirical studies on citation graphs and networks of users from Twitter show that our models can discover more effective results than the baseline approach.

Keywords: graph, influence aspect, influence degree, probabilistic generative model, Gibbs sampling

1 Introduction

Graphs have been widely used to represent objects and interactions between objects in many applications such as the Web, social networks, and citation networks. The connections among objects (i.e., edges between graph nodes) in graphs capture that one object has influences over other objects. However, these connections are not equally important. For instance, strong and weak ties often exist in social networks [22]. Most existing graphs do not directly capture and represent how strong and on what aspects the influence relationships are.

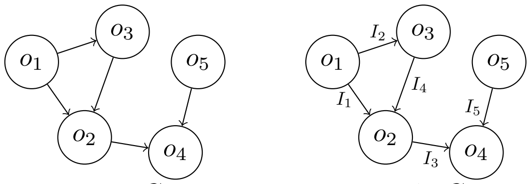
Let us use *influence aspect* to denote on which area (or dimension) that one object influences another object and the *influence degree* to denote how strong the influence is. Mining and providing influence relationships with influence aspects and influence degrees to describe object connections can greatly improve the use of graphs in the process of graph exploration (e.g., [15, 23, 26]), graph search (e.g., [16]) , and connection recommendations based on object relationships in graphs (e.g., [21, 24]). We give two examples to show the need for mining influence degrees and aspects from graphs.

Motivating Example 1: Research articles often cite each other to show their connections. The citation relationships among articles can provide a researcher helpful information to find articles related to what (s)he is interested in. However, given a research article p, there may be hundreds or thousands of research papers citing it. Although all these articles are somehow influenced by the article p, only very few of them are highly influenced by it regarding one aspect (e.g., methodology). Being able to discover on which aspect and how strong one article influences the others can greatly help a researcher when (s)he explores or searches publication collections.

Motivating Example 2: People in a social network are connected with and have influence to each other. However, even for directly connected people, their influence to each other is different. Discovering the strength of connections and the connection types can help identify strong ties (i.e., people with high influence relationship). Stronger influence relationship can provide more information for the recommendation. For instance, if a user named Alice is influenced by her friend Sarah more on *entertainment* aspect and influenced more by her friend Bob on *study* aspect, recommendations to Alice about which movie to watch should use more movie preference and movie list from Sarah.

In the literature, much effort has been put to discover influences on different types of graphs. But most of these techniques [4, 15, 23, 26] focus on detecting influence degrees.

Our work is different in that we detect both *influence aspects and influence degrees at the aspect level* from one graph by utilizing the topological structure of graphs, the text content associated with graph nodes, and the context in which the graph nodes are connected. In particular, influence aspects denote how two objects are connected and influence degrees at the aspect level describe how strongly two graph nodes are connected on the given aspect.



(a) G_s (b) G_t

I_1	I_2	I_3	I_4	I_5
b:0.6	p:0.5	p:0.6	e:0.7	m:0.7
p:0.3	b:0.26	e:0.3	p:0.2	b:0.2
m:0.95	e:0.14	b:0.1	b:0.1	p:0.1
e:0.05	m:0.1	m:0.0	m:0.0	e:0.0

(c) Influence vectors for G_t

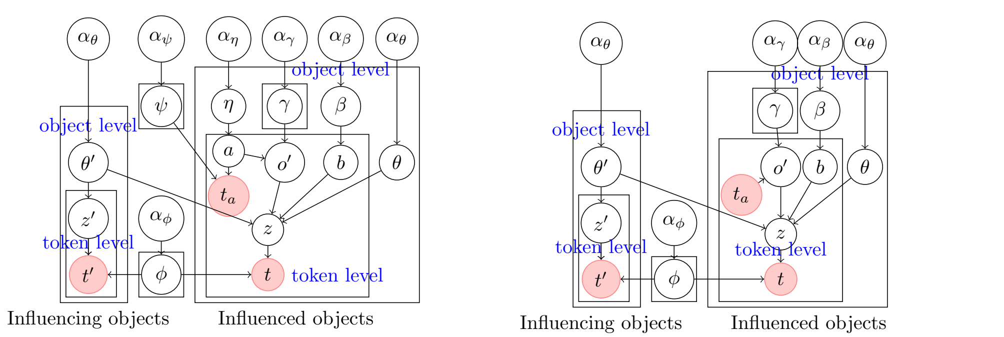
Figure 1: Graphs G_s (without influence relationships) and imaginary G_t (with influence relationships): the influence aspects that can be annotated on the edges are: b (ackground), e (xperiment), m (ethodology), p (roblem)

Three major challenges need to be addressed to discover such influence information from graphs. The first challenge is to properly define, represent, and model influence aspects and aspect-level influence degrees. The second challenge is to objectively measure the effectiveness of the discovered results. The third challenge is to make full use of both the text and structure information of graphs in the discovery process because they all carry informative knowledge.

We propose two novel Aspect Influence (AI) models to capture the influence aspects and influence degrees at the aspect level by utilizing both the text and structure information in a graph. In particular, we model the text information of graph nodes using latent topic states. We model the graph edges (structure) by associating with them latent influence aspects and aspect-level degrees. With such modeling, from the perspective of generating a graph node, the content of a graph node can be generated either by introducing novel information or by inheriting information (tokens, topics) from the connected graph nodes on the influence aspects. When the latent influence degree from one node o' to another node o on a given aspect a is higher, more tokens of o are drawn from o' on aspect a. Our contributions are as follows.

- We formally define the problem of discovering aspect level influence relationships, which consist of influence aspects and influence degrees on specific aspects, among graph nodes.
- We propose two generative probabilistic models, Latent Aspect Influence Model (LAIM) and Observed Aspect Influence Model (OAIM). These two models describe the generative process of graph node contents by considering both text and structure information in graphs.
- We implement a blocking Gibbs sampling approach to learn these two models.
- We perform extensive experiments using real data sets (citation network, Twitter) to show the effectiveness of the proposed approaches. Quantitative comparisons of LAIM and OAIM with one baseline approach show that introducing influence aspects can provide more comprehensive influence relationships among graph nodes.

This paper is organized as follows. Section 2 formally defines the problem and related notations. Sections 3 and 4 explain in detail our models and the proposed approach to learn the models and to derive the influence relationships using the models. Section 6 reviews the related work in this area. Section 5 shows our experimental results on the proposed approaches. Finally, Section 7 concludes this paper and shows possible future directions.



(a) Latent Aspect Influence Model (LAIM) (b) Observed Aspect Influence Model (OAIM)

Figure 2: Aspect Influence Models

2 Problem formulation

Our study takes as input a graph $G = (V, E)$, where V is the set of graph objects o and E is a set of directed edges $o' \to o$. Each graph node (or an object) o is associated with descriptive information. For instance, in a citation network of research articles, one research article is a graph object o and it contains a list of words; In social networks, a user is a graph object, which is associated with descriptive information. We use *object profile* to denote the descriptive information associated with each graph object and formally define it as follows.

Definition 2.1 (Object profile) *The profile of an object o is a sequence of tokens: $t_{o,1}, t_{o,2}, \cdots, t_{o,T(o)}$, where $T(o)$ is the number of tokens in the profile of o and $t_{o,pos}$ is the observed value of this object at position pos.*

We use D and T to represent the total number of objects in graph G and the total number of distinct tokens in all the object profiles respectively. We also use $\mathcal{D}$ to denote the set of tokens in the profiles of all the graph objects.

A directed edge $o' \to o$ in G denotes that object o' influences object o. One object may influence another object differently on different aspects (or dimensions) as described in the motivating examples in Section 1. Such aspects may be explicitly denoted (i.e., observed) in the graph. E.g., in citation networks, the observed aspect terms can be background, problem, methodology, etc. In social networks, the observed aspect terms can be politics, music, movie, etc. Let $\mathcal{A}$ be the set of observed aspect terms and T_a be the total number of distinct observed aspect terms in $\mathcal{A}$.

Definition 2.2 (Influence aspect) *One observed influence aspect a is an observed term from $\mathcal{A}$. One latent influence aspect a is a probabilistic distribution among all the T_a distinct observed aspect terms.*

We use A to denote the total number of latent influence aspects. The definition of the latent influence aspect is similar to the topic definition in topic modeling [2], where each latent topic is a distribution over the observed tokens that are used to describe objects. However, they are different in that the observed aspect terms for a latent aspect is from $\mathcal{A}$ while the observed tokens for a latent topic is from $\mathcal{D}$, and $\mathcal{A} \neq \mathcal{D}$.

Definition 2.3 (Aspect-level influence degree) *When an object o' influences another object o on an aspect a, we use $I(o' \xrightarrow{a} o)$, which is a number in the range of $[0, 1]$, to denote the influence degree (or influence strength) that o' has over o on aspect a.*

Influence is directional and influence degree is not symmetric, i.e., $I(o' \xrightarrow{a} o)$ does not equal to $I(o \xrightarrow{a} o')$ generally.

Definition 2.4 (Research Problem) *Given a graph G_s=(V, E), our problem is to discover an annotation function f to annotate G_s to G_t = $(V, E, \mathcal{A}, W)$ where V and E are the same to that in G_s, $\mathcal{A}$ and W are annotations on E such that $f(e, a) = w$ where $e \in E$, $a \in \mathcal{A}$, $w \in W$, and $w \in (0.0, 1.0]$.*

For example, given a graph G_s in Figure 1(a), the discovery algorithm is to output another graph G_t which is associated with vectors of influence aspects and influence degrees on those aspects.

3 Aspect influence models

Objects' profiles and their influence relationships are governed by three major factors. The first factor is the set of latent topic states associated with each object. An object's latent topic states determine the internal theme of this object. Two objects with similar internal themes tend to influence each other more compared with objects that do not share internal themes. E.g., an article with theme "bioinformatics" is more likely influenced by articles with theme "biology" than by articles with theme "politics". This factor has been taken into consideration in research of topic modeling [2]. The second factor is the set of links that connect the objects (i.e., edges between graph nodes) [4, 15]. The explicitly linked objects have obvious influence relationships, so the model should take the explicit links into consideration. The third factor is the context in which the object links appear [13]. Such context can reflect the influence aspect. For example, the link connecting one article to another article appears in the context of "experimental results" is more likely reflecting influence aspect "experiment" or "problem definition", but is less likely reflecting influence aspect "entertainment". Our solution framework is to learn these governing factors and derive the influence relationships from these governing factors. In our approaches, we adopt the definition of latent topic states which are probabilistic distributions over tokens (or words). We define two types of influence aspects in our models as discussed in Sections 3.1 and 3.2.

In our framework, the object collections can be viewed as a set of profiles generated by a generative process. The parameters used in the aspect influence models are listed in Table 1.

Symbol	meaning
o, o'	an influenced object and an influencing object
t, t'	tokens for o and o' respectively
t_a	observed aspect term for a token
b	latent boolean variable
a	latent aspect variable
z, z'	latent topic states for o and o' respectively
$T(o)$	the number of tokens in the profile of object o
$R(o)$	the set of objects that influence o
D	the total number of objects
$\mathcal{D}$	the set of all the observed tokens in object profiles
T	the total number of distinct tokens
T_a	the total number of distinct aspect terms
Z	the total number of latent topic states
A	the total number of latent influence aspects
$\mathcal{A}$	the set of all the observed aspect terms
$\vec{\alpha}_\phi$	Dirichlet prior parameters =$(\alpha_\phi, \cdots, \alpha_\phi)$ for t
$\vec{\alpha}_\theta$	Dirichlet prior parameters =$(\alpha_\theta, \cdots, \alpha_\theta)$ for z
$\vec{\alpha}_\eta$	Dirichlet prior parameters =$(\alpha_\eta, \cdots, \alpha_\eta)$ for a
$\vec{\alpha}_\psi$	Dirichlet prior parameters =$(\alpha_\psi, \cdots, \alpha_\psi)$ for t_a
$\vec{\alpha}_\gamma$	Dirichlet prior parameters =$(\alpha_\gamma, \cdots, \alpha_\gamma)$ for o'
$\vec{\alpha}_\beta$	Beta prior parameter=$(\alpha_{old}, \alpha_{new})$ for b
ϕ	a $Z \times T$ matrix
θ, θ'	a $D \times Z$ matrix
γ	a $D \times T_a \times L(D)$ matrix, where $L(D) = max_o(R(o))$
η	a $D \times A$ matrix
β	a $D \times 2$ matrix
ψ	a $A \times T_a$ matrix

Table 1: Parameters for aspect influence models

3.1 Latent Aspect Influence Model (LAIM)

The creation of an object profile can be envisioned as being generated from several foundamental ideas. From the perspective of generating an object profile, when an object is not influenced by any other objects, its tokens' generation is only governed by their latent topics, which is not the focus of this paper. The Latent Aspect Influence Model (LAIM) is introduced mainly to govern the generation of profile tokens for objects that are influenced by other objects. When an object o is influenced by another object o', then part of o's idea (i.e., profile) is new, but the other part of its idea is borrowed from o'. When o borrows idea from o', the tokens will be affected by both the major idea of o' (i.e., topics of o') and the influence aspect from o' to o. Inspired by the generation of tokens in the topic modeling [2], where each profile token is assumed to be associated with a latent topic and is draw from a topic-token distribution, Latent Aspect Influence Model (LAIM) assumes that every profile token of o is not only associated with a latent topic from o', but also associated with an influencing aspect from its influencing objects. Thus, its generation is controlled by both the (observed/latent) influence aspect and the influencing object's latent topic.

Figure 2(a) shows the aspect influence model which models the influence aspect as a latent state a. We denote this model as Latent Aspect Influence Model (LAIM). In LAIM, there are two types of observed variables, object-profile tokens t and influence-aspect terms t_a. In LAIM, we incorporate two types of objects: the objects that explicitly influence other objects (i.e., graph nodes with outgoing edges) and the objects that are explicitly influenced by other objects (i.e., graph nodes with incoming edges).

For objects that explicitly influence other objects, we model that their profile tokens t' arise from latent topic states z' by using LDA model. The topic mixture θ' is used to represent the probabilistic distribution of a latent state over all the possible latent states. The prior of θ' is generated using a Dirichlet hyper-parameter $\vec{\alpha}_\theta = (\alpha_\theta, \cdots, \alpha_\theta)$.

For objects that can be influenced by other objects (i.e., graph nodes with incoming edges), we model their profiles by considering their own contents and their links from other objects. The profile of an object o is formed with both new information and inherited information from other objects that influence o. An object o's latent topic z (which is used to draw token t) can be drawn from its own topic mixture θ or its influencing objects' topic mixture θ'. This choice is modeled with a binary variable b which follows a Bernoulli distribution. The Bernoulli distribution's parameter β is learned by the model, given an asymmetric Beta prior $\vec{\alpha}_\beta = (\alpha_{old}, \alpha_{new})$.

1. For every latent topic state z, draw $\vec{\phi}_z \sim dirichlet(\vec{\alpha}_\phi)$
2. For every latent aspect a, draw $\vec{\psi}_a \sim dirichlet(\vec{\alpha}_\psi)$
3. For every object o that **can influence other objects**
   - (a) Draw $\vec{\theta}'_o \sim dirichlet(\vec{\alpha}_\theta)$
   - (b) For each position pos in the profile of object o
     - i. Generate z' with $p(z'|o; \vec{\theta'}_o) \sim multi(\vec{\theta'}_o)$
     - ii. Generate t' with $p(t'|z'; \vec{\phi}_{z'}) \sim multi(\vec{\phi}_{z'})$
4. For each object o that **can be influenced by other objects**
   - (a) Draw $\vec{\theta}_o \sim dirichlet(\vec{\alpha}_\theta)$
   - (b) Draw $\vec{\eta}_o \sim dirichlet(\vec{\alpha}_\eta)$
   - (c) For each latent aspect a, draw $\vec{\gamma}_{o,a} \sim dirichlet(\vec{\alpha}_\gamma)$
   - (d) Draw the proportion between newly generated values and values generated from its influencing object $\beta_o \sim beta(\alpha_{old}, \alpha_{new})$
   - (e) For each position pos of the profile for object o, generate its token from either an influencing object or innovative state.
     - i. Draw a coin b with $p(b|o) \sim bernoulli(\beta_o)$
     - ii. if $b = 0$,
       - A. Draw z with $p(z|o; \vec{\theta}_o) \sim multi(\vec{\theta}_o)$
       - B. Generate t with $p(t|z, \vec{\phi}_z) \sim multi(\vec{\phi}_z)$
     - iii. if $b = 1$, draw an influence aspect a, an influencing object o', and a topic state z' for o from o'
       - A. Draw a with $p(a|o, \vec{\eta}_o) \sim multi(\vec{\eta}_o)$
       - B. Generate t_a with $p(t_a|a, \vec{\psi}_a) \sim multi(\vec{\psi}_a)$
       - C. Draw an interacting object o' with $I(o' \rightarrow o|a) \sim multi(\vec{\gamma}_{o,a})$
       - D. Draw z=z' with $p(z'|o', \vec{\theta'}_{o'}) \sim multi(\vec{\theta'}_{o'})$
       - E. Generate t = t' with $p(t'|z', \vec{\phi}_{z'}) \sim multi(\vec{\phi}_{z'})$

Figure 3: Generative process for LAIM: z', z, t', t, b, a, o' represent $z'_{o,pos}$, $z_{o,pos}$, $t'_{o,pos}$, $t_{o,pos}$, $b_{o,pos}$, $a_{o,pos}$, $o'_{o,pos}$ respectively

Based on the above described ideas, the generative process of object profiles and their interactions are outlined in Figure 3. This generative process takes as input a graph G_s=(V, E) with connected objects in V and outputs an annotated graph G_t=$(V, E, \mathcal{A}, W)$. The first two steps of the generative process generates the latent topic to token distribution mixture ϕ, a $Z \times T$ matrix, and latent aspect to observed aspect term distribution mixture ψ. In particular, $\vec{\phi}_z$ captures the distribution of a given latent topic state z on all the observed tokens. $\vec{\psi}_a$ represents the distribution of a given latent influence aspect a on all the observed aspect terms t_a.

When an object o can influence other objects, we generate its latent topic states z' and tokens t' (Step 3).

When an object o can be influenced by other objects, we generate its profile either from its influencing object o' or from its own latent topic mixture (Step 4). When its profile tokens are generated with novel topics, we generate its latent topic state z and token t (Step 4(e)ii). On the other hand, when an object o can be influenced by other objects, its profile tokens are generated based on a latent aspect a from an influencing objects o'. In this case, the latent aspect needs to be drawn first. Then, the influencing object o' is drawn from the mixture component $\vec{\gamma}_{o,a}$ given the generated latent aspect a. Given the generated o', the generative process next generates the latent topic state z' (from θ') and token t' (Step 4(e)iii).

Note that an object can influence other objects and be influenced by others at the same time (e.g., o_2 and o_3 in Figure 1). In other words, an object has both incoming and outgoing edges. To model such objects, we use the similar idea as that in [4, 13, 19] to keep two copies of the object profiles: one copy is generated with Step 3 and the other copy is generated with Step 4. The generative process leverages these two procedures together.

3.2 OAIM model

The LAIM models an aspect as a latent state a and uses it to decide how o' affects the generation of o. Each a is represented as a distribution over the observed influence aspect terms. This model is complicated. With the intuition that the models with more variables are harder and less efficient to learn compared with models with less variables, we propose a second generative probabilistic model by treating observed influence aspect terms t_a as the influencing aspects. Figure 2(b) shows this probabilistic model, which is denoted as OAIM. This model contains less variables. Intuitively, it takes less time to learn OAIM than to learn LAIM given the same input. However, OAIM's results may not be as effective as LAIM. We compare these two different models in our experiments.

4 Model learning via Gibbs Sampling

We utilize the Gibbs sampling approach to learn the model parameters and latent states that control the generative process. Gibbs sampling [7] allows to learn a model by iteratively updating each latent variable when fixing the remaining variables. The latent variables in the generative process of Figure 3 are b, a, o', z', z.

To facilitate the sampling process, we need to keep counts of tokens assigned to different configurations. The update equations of LAIM and OAIM for the Gibbs sampling could be computed in constant time with the count cache.

For objects that can be influenced by other objects, we use $N_{dv_1,\cdots,dv_i}[v_1,\cdots,v_i]$ to denote the number of tokens assigned to the configuration with $dv_1 = v_1$, $dv_i = v_i$ during the sampling process. For example, $N_{o,a,o',b}[1,3,2,1]$ denotes the total number of tokens that are assigned to object o_1 and that come from object o_2 on aspect a_3 when the latent variable b is 1. For objects that can influence others, the count cache is denoted as N'. The detailed meaning of each count cache is given as follows.

For each object o that belongs to the set of *influencing objects*, we need to keep the following counts. The subscript "$_{b=i}$" denotes that the count is only used when $b = i$ where i can be 0 or 1.

- $N'_{o,z}$: the times to sample latent topic state z for object o.
- N'_o: the times to sample for object o.
  Here $N'_o(o_i)$=$\sum_{z_i=1}^{Z} N'_{o,z}(o_i, z_i)$.
- $N'_{z,t}$: the times to sample latent topic state z for token t for all the objects in the influencing object set.
- N'_z: the times to sample latent topic state z for all the objects in the influencing object set.
  Here, $N'_z(z_i)$=$\sum_{t_i=1}^{T} N'_{z,t}(z_i, t_i)$.
- $N'_{o',z',b=1}$: the times to sample latent topic state z' using object o *only when* o is used as an influencing object o' to generate latent states for other objects.
- $N'_{o',b=1}$: the times to sample object o *only when* o is used as an influencing object o' to generate latent states for other objects.
  Here, $N'_{o',b=1}(o_i, 1)$=$\sum_{z_i=1}^{Z} N'_{o',z',b=1}(o_i, z_i, 1)$.

For each object o which belongs to the set of *influenced objects*, we need to keep the following counts.

- $N_{o,z,b=0}$: the times to sample latent topic state z for object o. This happens only when b=0.
- $N_{o,o',b=1}$: the times to use o' to draw the tokens for o. This happens only when b=1.
- $N_{o,b}$: the times to sample different b values for object o. Here, $N_{o,b}(o_i, 0) = \sum_{z_i=1}^{Z} N_{o,z,b=0}(o_i, z_i, 0)$,
  $N_{o,b}(o_i, 1) = \sum_{o'_j \in R(o_i)} N_{o,o',b=1}(o_i, o'_j, 1)$
- N_o: the times to sample object o.
  Here $N_o(o_i)$ = $\sum_{b_i=0}^{1} N_{o,b}(o_i, b_i)$.
- $N_{o,a,o',b=1}$: the times to sample the latent influencing aspect a using influencing object o'.
  $N_{o,o',b=1}(o_i, o'_j, 1)$ = $\sum_{a_k=1}^{A} N_{o,a,o',b=1}(o_i, a_k, o'_j, 1)$.
- $N_{o,a,b=1}$: the times to sample aspect a for object o.
  $N_{o,a,b=1}(o_i, a_k, 1)$
  =$\sum_{o'_j \in R(o_i)} N_{o,a,o',b=1}(o_i, a_k, o'_j, 1)$.
- $N_{a,t_a,o',b=1}$: the times to sample for observed aspect t_a using influencing object o' .
- $N_{a,t_a,b=1}$: the times to sample for observed aspect t_a.
  $N_{o,t_a,b=1}(o_i, t_{a_k}, 1)$
  =$\sum_{o'_j \in R(o_i)} N_{o,t_a,o',b=1}(o_i, t_{a_k}, o'_j, 1)$.

(1) $$p(b_{pos}=0|\vec{t},\vec{t'},\vec{z},\vec{z'},\vec{o'},\vec{a},\vec{b}_{\neg pos}) \propto \rho_{o,z,b0} \cdot \frac{N_{o,b}(o,0)+\alpha_{new}-1}{N_o(o)+\alpha_{old}+\alpha_{new}-1}$$

(2) $$p(b_{pos}=1|\vec{t},\vec{t'},\vec{z},\vec{z'},\vec{o'},\vec{a},\vec{b}_{\neg pos}) \propto \rho_{o=o',b1} \cdot \frac{N_{o,b}(o,1)+\alpha_{old}-1}{N_o(o)+\alpha_{old}+\alpha_{new}-1}$$

(3) $$p(a_{pos}|\vec{t},\vec{t'},\vec{z},\vec{z'},\vec{o'},\vec{a}_{\neg pos},\vec{t_a},\vec{b}) \propto \frac{N_{o,a,o',b}(o,a_{pos},o'_{pos},1)+\alpha_\gamma-1}{N_{o,a,b}(o,a_{pos},1)+|R(o)|\alpha_\gamma-1} \cdot \frac{N_{a,t_a,b}(a_{pos},t_{a_{pos}},1)+\alpha_\psi-1}{N_{a,b}(a_{pos},1)+T_a\alpha_\psi-1} \cdot \frac{N_{o,a,b}(o,a_{pos},1)+\alpha_\eta-1}{N_{o,b}(o,1)+A\alpha_\eta-1}$$

(4) $$p(o'_{pos}|\vec{t},\vec{t'},\vec{z},\vec{z'},\vec{o'}_{\neg pos},\vec{a},\vec{b}) \propto \rho_{o=o',b1} \cdot \frac{N_{o,a,o',b}(o,a_{pos},o'_{pos},1)+\alpha_\gamma-1}{N_{o,a,b}(o,a_{pos},1)+|R(o)|\alpha_\gamma-1}$$

(5) $$p(z_{pos}|\vec{t},\vec{t'},z_{\vec{\neg pos}},\vec{z'},\vec{o'},\vec{a},\vec{b}_{\neg pos},b_{pos}=0) \propto \rho_{z,t}\cdot\rho_{o,z,b0}$$

(6) $$p(z_{pos}|\vec{t},\vec{t'},\vec{z}_{\neg pos},\vec{z'},\vec{o'},\vec{a},\vec{b}_{\neg pos},b_{pos}=1) \propto \rho_{z,t}\cdot\rho_{o=o',b1}$$

Given the following count ratio ρ,

$$\rho_{o=o',b1} = \frac{N'_{o',z',b=1}(o'_{pos},z_{pos},1)+N'_{o,z}(o'_{pos},z_{pos})+\alpha_\theta-1}{N'_{o',b=1}(o'_{pos},1)+N'_o(o'_{pos})+Z\alpha_\theta-1}$$
$$\rho_{z,t} = \frac{N_{z,t}(z_{pos},t_{pos})+N'_{z,t}(z_{pos},t_{pos})+\alpha_\phi-1}{N_z(z_{pos})+N'_z(z_{pos})+T\alpha_\phi-1}$$
$$\rho_{o,z,b0} = \frac{N_{o,z,b}(o,z_{pos},0)+\alpha_\theta-1}{N_{o,b}(o,0)+Z\alpha_\theta-1}$$

- $N_{a,b=1}$: the times to sample a.
  $N_{a,b=1}(a_i,1)=\sum_{t_a=1}^{T_a} N_{a,t_a,b=1}(a_i,t_a,1)$.
- $N_{z,t}$: the times to sample t for latent state z.
- N_z: the times to sample latent state z.
  $N_z(z_i)=\sum_{t_i=1}^{T} N_{z,t}(z_i,t_i)$.

Example 4.1 *Given the graph in Figure 1 with objects o_1,o_2,o_3,o_4,o_5. The set of influencing objects is $S'=\{o_1,o_2,o_3,o_5\}$ and the set of influenced objects is $S=\{o_2,o_3,o_4\}$. For every object in S', we need to maintain all N' counts. For every object in S, we need to maintain all the N counts.*

The Gibbs sampling algorithm needs to sample the latent variables using update equations. The update equations used to learn the aspect influence models are given in Equations 1 – 6. We show the detailed derivation of Equation 3 in Section 9.1. The other equations can be derived analogously.

In each iteration i, to generate the profile tokens for object o, o's tokens come from other objects that influence o when b is 1. Let $T'(a)$ be the total number of tokens coming from other objects for influence aspect a, and let $T'(o',a)$ be the total number of tokens coming from a specific influencing object o' on aspect a. Then, the ratio $\frac{T'(o',a)}{T'(a)}$ can estimate the strength that o' influences o on aspect a, i.e., $I(o' \xrightarrow{a} o)$.

Since the expected value of any parameter variables can be approximated by averaging over all the samples [15] after the sampling chain converges (i.e., after the burn-in phase), the influence that o_k has over o_j on aspect a can be estimated as follows.

(7) $$I(o_k \xrightarrow{a} o_j) = \frac{1}{m}\cdot\sum_{i=1}^{m}\frac{N_{o,a,o',b}[o_j,a,o_k,1]^{(i)}+\alpha_\gamma}{N_{o,a,b}[o_j,a,1]^{(i)}+|R(o_j)|\alpha_\gamma}$$

where m is the total number of iterations of the sampling chain after converge, and the superscript $^{(i)}$ denotes the i-th iteration.

OAIM. In learning OAIM, the update Equations 1, 2, 5, 6 are still used. However, Equation 3 is not needed for OAIM because OAIM does not model latent influence aspect a anymore. In addition, Equation 4 is changed to Equation 8.

(8) $$p(o'_{pos}|\vec{t},\vec{t'},\vec{z},\vec{z'},\vec{o'}_{\neg pos},\vec{t_a},\vec{b}) \propto \rho_{o=o',b1} \cdot \frac{N_{o,t_a,o',b}(o,t_a,o'_{pos},1)+\alpha_\gamma-1}{N_{o,t_a,b}(o,t_a,1)+|R(o)|\alpha_\gamma-1}$$

To implement the Gibbs sampling algorithm, we adopt the *blocking Gibbs sampling* strategy [12], which samples several variables together as a block to speed up the Gibbs sampling process for complex probabilistic models. In particular, to sample token t_{pos} for influenced objects, all the latent variables associated with position pos (i.e., b_{pos}, a_{pos} (for *LAIM*), o'_{pos}, z_{pos}) are sampled together. To sample token t_{pos} for influencing objects, the latent variable that we need to sample is z'_{pos}.

Time Complexity.

Let M be the total number of iterations before the sampling process converges.

For influencing objects, the sampling procedure is the same for both the LAIM and OAIM models. In each sampling iteration, the sampling process samples a new latent state $z_{pos} \in \{1, \ldots, Z\}$ for each token t_{pos} in every object's profile. So the time complexity for sampling influencing objects in both LAIM and OAIM is $O(M \cdot D \cdot \bar{T}(o) \cdot Z)$, where $\bar{T}(o)$ is the average number of tokens in object o.

For the influenced objects, the sampling process in each iteration draws a new block of latent variables b_{pos}, a_{pos}(LAIM), o'_{pos}, z_{pos} for each token t_{pos} in every object. Since we adopt the blocked Gibbs sampling strategy, the candidate size for new sample at each position is the product of the numbers for each variable's values (or states). Thus, the time complexity of sampling influenced objects is $O(M \cdot D \cdot \bar{T}(o) \cdot Z \cdot \bar{R}(o) \cdot A)$ for LAIM and is $O(M \cdot D \cdot \bar{T}(o) \cdot Z \cdot \bar{R}(o))$ for OAIM, where $\bar{R}(o)$ is the average number of objects that influence the object o.

5 Experiments

All the three approaches are implemented using Java and run on a workstation with Intel(R) i7-2600 Quad Core CPU @ 3.40GHz and 16G memory, running OpenSUSE.

Baseline approach. Since no other works exist to model object influences at the aspect level, we implement a simple intuitive approach to calculate the influence relationships at the aspect level to compare with LAIM and OAIM. This approach, denoted as *Baseline* approach, calculates the influence relationships based on the contents of object profiles. It first calculates the normalized TF-IDF value for each token in all the objects' profiles. Let $tfidf(t_{pos}|o)$ be the normalized tf-idf value of a token at position pos in the profile of object o. Let $\vec{tfidf}_{o'}$ be the tf-idf vector for an influencing object o' and $\vec{tfidf}_{o,a}$ be the tf-idf vector for an influenced object o in the context related to aspect a. The influence from an object o' to object o on aspect a can be estimated as $I(o' \xrightarrow{a} o)$=$cosine(\vec{tfidf}_{o'}, \vec{tfidf}_{o,a})$. This value leverages the influence from o' to all the tokens in o within the context of aspect a. The baseline approach utilizes one parameter $\lambda \in (0, 1)$ to capture the probability of sampling new tokens for an object o. Accordingly, $1 - \lambda$ is the probability of sampling a token for an object o from other influencing objects o'.

Data sets. We used two real data sets, *CiteMisc* and *Twitter5000*, to test our proposed approaches. (1) *CiteMisc* data set contains forty six research articles, which we collected on eight major research topics, including association rule mining, clustering, frequent pattern mining, keyword search over graphs, topic modeling, information retrieval, spatial database query processing, XML database queries. To collect this data set, we first chose eight articles, each of which is from one topic, as seeds. Then, for each of these articles o, we find the set of the articles $\{o'\}$ such that o' is directly cited by o (i.e., o' appears in the reference list of o) or o' is indirectly cited by o. We manually label these research articles with eight observed influence aspects: abstract, background, problem definition, methodology, experiment introduction, experiment data, experiment comparison, experiment analysis. The total number of distinct tokens, T, is $\sim$1000 and each article cites (i.e., influenced by) $\sim$5 articles on average. (2) *Twitter5000* data set contains 5000 twitter users and their recent 200 tweets. The total number of tokens is $\sim 5M$ and the number of distinct tokens is $\sim 15K$. Each twitter user in this data set follows (i.e., implicitly is influenced by) $\sim$56 users on average. This data set embeds $\sim 2K$ observed aspects, which are the categories tagged to the users. For scalability tests, we partition this data set to five smaller data sets with 500, 1000, 1500, 2000, 5000 users. The corresponding approximate numbers of total tokens are $0.5M$, $1.1M$, $1.6M$, $2.2M$ and $4.8M$.

The burn-in value for Gibbs sampling process is generally set to be half of the total number of iterations [6]. In our experiment, the burn-in value in different settings is set to 100, which is almost half of the total number of iterations. The hyper-parameters in LAIM and OAIM are set as $\alpha_\phi = 0.01$, $\alpha_\theta = 0.1$, $\alpha_\gamma = 0.1$, $\alpha_\psi = 0.1$, $\alpha_\eta = 0.1$, $\alpha_\beta = 0.5$. As suggested in [9], each vector parameter (e.g., $\vec{\alpha}_\phi$) is set as a symmetric prior. I.e., the scalar values in the parameter vector are the same. Second, we set the values of these parameters according to the conclusion from [8]. We also note that previous research [25] shows that these hyper-parameters has strong impact on model inference and how to set up and estimate hyper-parameters is not easy and not throughly investigated [9]. In our future work, we will perform

$$p(\vec{t^o}) = \prod_{j=pos}^{T(o)} \left(\lambda \cdot p(t_{pos}|o) + (1-\lambda) \sum_{o' \in R(o)} I(o' \xrightarrow{a} o) p(t_{pos}|o') \right) \tag{9}$$

where $p(t_{pos}|o)$ is the normalized tf-idf value for t_{pos}.

$$p(\vec{t'^o}) = \prod_{i=1}^{T(o)} \left(\sum_{z'=1}^{Z} p(t'_i|z_i = z') \cdot p(z_i = z'|o) \right) = \prod_{t'=1}^{T(o)} \left(\sum_{z'=1}^{Z} \phi_{z',t'} \cdot \theta'_{o,z'} \right) \tag{10}$$

$$\begin{aligned} p(\vec{t^o})_{LAIM} = &\prod_{i=1}^{T(o)} \Bigg(p(b_i = 0) \cdot \Big(\sum_{z=1}^{Z} p(t_i|z_i = z) \cdot p(z_i = z|o) \Big) \\ &+ p(b_i = 1) \cdot \Big(\sum_{a=1}^{A} \sum_{o' \in R(o)} \sum_{z'=1}^{Z} p(a|o) \cdot p(o'|o,a) \cdot p(z_i = z'|o') \cdot p(t_i|z_i = z') \Big) \Bigg) \\ = &\prod_{t=1}^{T(o)} \Bigg(\beta_{o,new} \cdot \Big(\sum_{z=1}^{Z} \phi_{z,t} \cdot \theta_{o,z} \Big) + \beta_{o,old} \cdot \Big(\sum_{a=1}^{A} \sum_{o' \in R(o)} \sum_{z'=1}^{Z} \eta_{o,a} \cdot \gamma_{o,a,o'} \cdot \theta'_{o',z'} \cdot \phi_{z',t} \Big) \Bigg) \end{aligned} \tag{11}$$

$$\begin{aligned} p(\vec{t^o})_{OAIM} = &\Pi_{i=1}^{T(o)} \Bigg(p(b_i = 0) \cdot \Big(\sum_{z=1}^{Z} p(t_i|z_i = z) \cdot p(z_i = z|o) \Big) \\ &+ p(b_i = 1) \cdot \Big(\sum_{o' \in R(o)} \sum_{z'=1}^{Z} p(o'|o, a = t_{a_i}) \cdot p(z_i = z'|o') \cdot p(t_i|z_i = z') \Big) \Bigg) \\ = &\Pi_{t=1}^{T(o)} \Bigg(\beta_{o,new} \cdot \Big(\sum_{z=1}^{Z} \phi_{z,t} \cdot \theta_{o,z} \Big) + \beta_{o,old} \cdot \Big(\sum_{o' \in R(o)} \sum_{z'=1}^{Z} \gamma_{o,t_{a_i},o'} \cdot \theta'_{o',z'} \cdot \phi_{z',t} \Big) \Bigg) \end{aligned} \tag{12}$$

Figure 4: Predictive likelihood functions

thorough synthetic analysis about the effect of these hyper-parameters over our influence models. The final output parameter result is the average of that in each iteration after burn-in period but before converge.

5.1 Effectiveness Measurements

Both subjective and objective measurements are used to evaluate the results. For subjective evaluation, we did case studies on two small data sets (Section 5.2.1). We also used two objective measures *Precision*$@K$ and *predictive log-likelihood* to show the effectiveness (Sections 5.2.2 and 5.2.3 respectively).

Log-Likelihood. The predictive log-likelihood is calculated on the tokens in the profiles of both influencing and influenced objects to compare the generalization capabilities of the baseline approach and our models, LAIM and OAIM. For the baseline approach, the likelihood for all the tokens in object o is calculated as Equation 9.

For LAIM and OAIM, we adopt a similar approach as that in [13,20] to calculate the predictive log-likelihood by extending the Gibbs sampling process over the unseen data. For LAIM, the Gibbs sampling process already learned $\Pi_{LAIM} = (\phi, \theta, \theta', \gamma, \eta, \beta, \psi)$ where these parameters are explained in detail in Table 1. With D_{new} new objects, we learn $\Delta(\theta)$=$(\vec{\theta}_{D+1}, \cdots, \vec{\theta}_{D+D_{new}})$, $\Delta(\theta')$=$(\vec{\theta'}_{D+1}, \cdots, \vec{\theta'}_{D+D_{new}})$, $\Delta(\gamma)$=$(\gamma_{D+1}, \cdots, \gamma_{D+D_{new}})$. The initialization of $\Delta(\theta)$, $\Delta(\theta')$ and $\Delta(\gamma)$ are the same as that of initializing θ, θ' and γ in the initial Gibbs Sampling. Then, the Gibbs sampling continues to leverage the new objects until the sampling chains converge. Given the newly learned parameters $\Pi_{LAIM}^{D+D_{new}}$, the predictive likelihood of a new influencing object o, denoted as $p(\vec{t'^o})$, can be expressed as Equation 10. The predictive likelihood of a new influenced object o, $p(\vec{t^o})_{LAIM}$, can be represented in detail as Equation 11.

For OAIM, the Gibbs sampling process already learned $\Pi_{OAIM} = (\phi, \theta, \theta', \gamma, \beta)$. In the continued Gibbs Sampling, the update equations are the same as those in Section 4 except that any count with new object o or o' is initialized to zero. The likelihood of a new influencing object o, $p(\vec{t'^o})$ is still calculated using Equation 10. The predictive likelihood of a new influenced object o is denoted as $p(\vec{t^o})_{OAIM}$ and is detailed as Equation 12.

User u	Top-2 aspects	Top-2 influencing users
elisefoley	Election Day/Night 2012	TeamRomney, samyoungman
	National Security	marcambinder, rozen
dcharlesReuters	national politics	WSJ, Lis Smith
	US politics	WSJ, HuffPostPol
aterkel	thinkprocess-staff	brainstelter, ThinkProgress
	self_created	David_Ingram, jaketapper
ReutersPolitics	Reporters	ZekeJMiller, mattspetalnick
	self_created	NewsHour, ReutersWorld
WestWingReport	self_created	JimPethokoukis, BillGates
	Journalism	mitchellreports, nytimes

Table 2: Top-2 Influence relationship discovered by OAIM on *Twitter500* data set

5.2 Result analysis

5.2.1 Aspect influence case studies

The first set of experiments is to show the effectiveness of our proposed approaches through case studies. To *subjectively* verify that the models learned meaningful results, we show a very few number of objects and their most influencing objects on given influence aspects for the *Twitter500* data set. We also did case studies on the *CiteMisc* data set and get meaningful results.

For the *Twitter5000* data set, we extract a small number of famous users and analyze the aspects on which they are influenced. These famous users are used because it is easier to subjectively verify the findings from our models. Table 2 shows a few number of results from our OAIM model. For instance, a user $elisefoley$, who is a reporter for a newspaper Huffington Post is influenced on the aspect *Election Day/Night 2012* by the user $TeamRomney$ the most. This information shows that this reporter's interest is greatly influenced by Romney's team in the election. In addition, $elisefoley$ is influenced on the aspect *National Security* by another user *marcambinder*, who is inside the Government Secrecy Industry. Similarly, a user *dcharlesReuters*, who is Reuters correspondent in Washington, covering homeland security issues, is influenced on the aspect of *national politics* by the *WSJ* (i.e., Wall Street Journal) the most.

id	Article	Top-2 aspects	Top-1 influencing article
p_1	Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and its Applications	Background	BIRCH: An Efficient Data Clustering Method for Very Large Databases
		Solution	A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise
p_2	Unsupervised Prediction of Citation Influences	Background	The Author-Topic Model for Authors and Documents
		Solution	Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks
p_3	Scalable Algorithms for Association Mining	Background	Set-Oriented Mining for Association Rules in Relational Databases
		Solution	Sampling Large Databases for Association Rules
p_4	Identifying Meaningful Return Information for XML Keyword Search	Background	Effective Keyword Search in Relational Databases
		Solution	Efficient Keyword Search for Smallest LCAs in XML Databases
p_5	Effective Keyword Search in Relational Databases	Background	ObjectRank: Authority-Based Keyword Search in Databases
		Solution	DISCOVER: Keyword Search in Relational Databases
p_6	Frequent Patterns without Candidate Generation	Background	Mining Association Rules with Item Constraints
		Solution	A Tree Projection Algorithm For Generation of Frequent Itemsets

Table 3: Top-1 influencing articles discovered by OAIM for the six seed articles in the *CiteMisc* data set

Table 3 shows the results of OAIM model on *CiteMisc* data set with the parameter setting Z=10. There are eight observed aspects *abstract, background, problem definition, methodology, experiment introduction, experiment data, experiment comparison, experiment analysis*. We chose 6 influenced articles and show the most influencing (i.e. top-1) articles for each of them on two aspects, *background* and *solution*, which OAIM discovers from the *CiteMisc* data set. For all these listed articles, the most influencing articles and the influence aspects are meaningful. They can give us further information. For instance, the article p_3 *Scalable Algorithms for Association Mining* is most influenced by the article *Set-Oriented Mining for Association Rules in Relational Databases* on the *Background* aspect because they all study the association rule mining problem. But when it comes to the *Solution* aspect, the article *Sampling Large Databases for Association Rules* influences p_3 more because they all work on scalable large data processing. We observe that OAIM model is returning us meaningful results from several Twitter users and CiteMisc dataset which we can verify subjectively.

To give a concrete idea about the results of *LAIM*, we also show a small number of users and their most influencing users on two latent influence aspects in Table 4. For instance, user *dcharlesReuters* is most influenced by *GStephanopoulos* (a chief political correspondent in ABC) on latent aspect A_8. As shown in Table 4 (b), A_8 has higher probability on observed aspect terms related to news. This makes sense because *dcharlesReuters* is a correspondent and his twitter account activity shows that he is influenced more on *news* aspect.

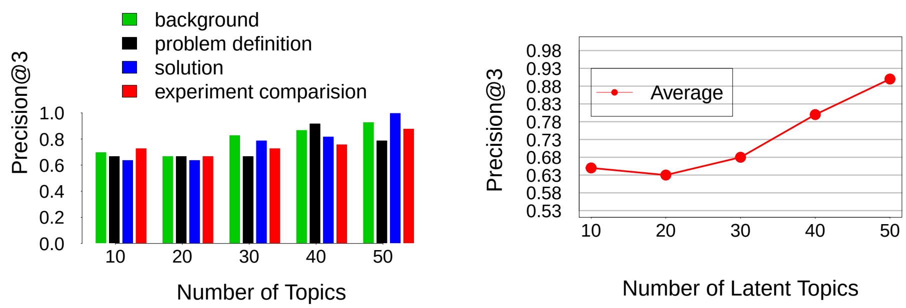
A user u	Aspect	Users followed by u
dcharlesReuters	latent aspect 8	GStephanopoulos
WestWingReport	latent aspect 3	brianstelter
rickklein	latent aspect 4	GStephanopoulos

(a) Three users with the top-1 most influencing latent aspects

A_3 (economy)	A_4 (foreign)	A_8 (news)
WSJ	NGB	National Journal on Twitter
OFA-States	Foreign Office on Twitter	Debate Watch 2012
WSJ staff	FCO Ministers	HotlineOnCall
Debate Watch 2012	London2012	newzhoundz
HotlineOnCall	Sochi 2014	Election Night

(b) Latent aspects A_3, A_4, and A_8

Table 4: Top-1 Influence relationships discovered by LAIM on *Twitter500* data set. (There are overall A (=10) latent influence aspects. Three latent influence aspects, A_3, A_4, and A_8, are shown for demonstration purpose.)



(a) Individual aspects (b) Average

Figure 5: Precision@3 for *CiteMisc* (OAIM model)

5.2.2 Precision@K

We also measure the precision of the top-K results on *CiteMisc* data set by manually creating the top-3 influencing objects for eight influencing aspects as ground truth. Then, we calculate the precision@3 of the results discovered by our OAIM model on different aspects. The precision on four aspects (other aspects show similar trends) and the average precision are plotted in Figure 5. The results show that our model can find results with high precision for different numbers of latent topics, and the precision increases with the number of latent topics. The precision@3 values of baseline model are: 0.54 (background), 0.48 (problem definition), 0.41 (solution), 0.57 (experiment comparison). These values are smaller than those learned from OAIM.

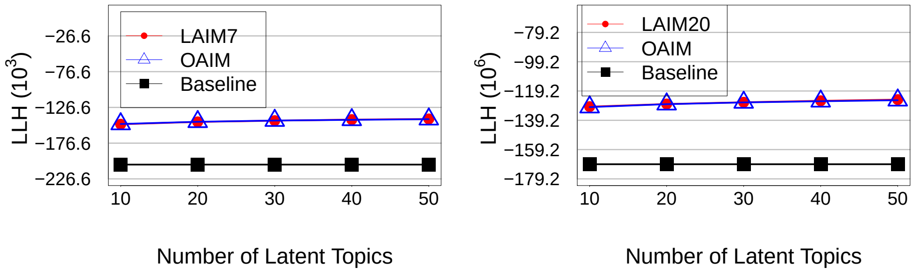
5.2.3 Log-likelihood measurements

The log-likelihood values of one experimental setting over one data set (Figures 6, 7, 8) are calculated. The reported log-likelihood is the average of log-likelihood of the objects in the test set. In these figures, LAIMx means that the number of latent aspects A is x.

We first compare the baseline approach with our LAIM and OAIM models and show the results in Figure 6. For *CiteMisc*, T_a=7. For *Twitter2000*, T_a=$\sim 2K$. On both *CiteMisc* and *Twitter2000*, OAIM and LAIM get much better log-likelihood than the baseline approach. This shows that the simple baseline approach is not as effective as our Aspect Influence Models.

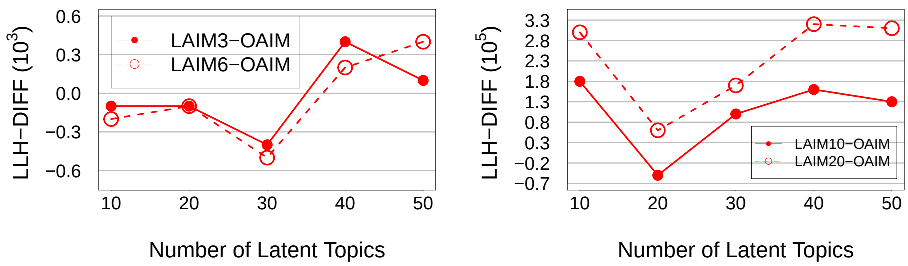
The second set of experiments is to compare the LAIM and OAIM models by running them on both *CiteMisc* and Twitter500 data sets. For this set of tests, we plot the difference of log-likelihood values (LLH-DIFF) between LAIM and OAIM. Figure 7 shows the results when we vary the number of latent topic states (Z). When *LLH-DIFF* is smaller than zero, it means that OAIM gets bigger log-likelihood value than LAIM, and OAIM performs better than LAIM accordingly. On the other hand, a positive *LLH-DIFF* value means that OAIM is performing worse than LAIM. Figure 7 shows that when the number of latent topic states Z is smaller, OAIM performs slightly better than LAIM (with negative *LLH-DIFF* value). But, when Z is bigger, LAIM performs slightly better than OAIM (with positive *LLH-DIFF* value).

The third set of experiments is to evaluate LAIM's performance when the number of latent aspects, A, changes. Figure 8 plots the results for both data sets when we vary A. The performance of LAIM improves with the increase of A when A is small. We can observe that LAIM's log-likelihood values on unseen data increase as A increases from 2 to 5 in Figure 8(a), and as A increases from 10 to 100 in Figure 8(b). This is because latent variables form a set



(a) *CiteMisc* (b) *Twitter2000*

Figure 6: Compare baseline approach with OAIM and LAIM



(a) *CiteMisc* (b) *Twitter2000*

Figure 7: Comparing OAIM and LAIM: log-likelihood difference vs. Z

of features extracted from observed variables; a model using more latent variables loses less information, thus shows higher likelihood in the testing samples. But LAIM's performance stabilizes or slightly decreases once it reaches a point. For the *CiteMisc* data set, which has eight observed aspects, LAIM's performance is the best when A is 5. For *Twitter2000* data set, which has $\sim 2K$ observed aspects, the best performance is achieved when A is 50. As the latent variables become more, the sample matrix is sparser, which decrease the predication capability of the model.

5.2.4 Efficiency

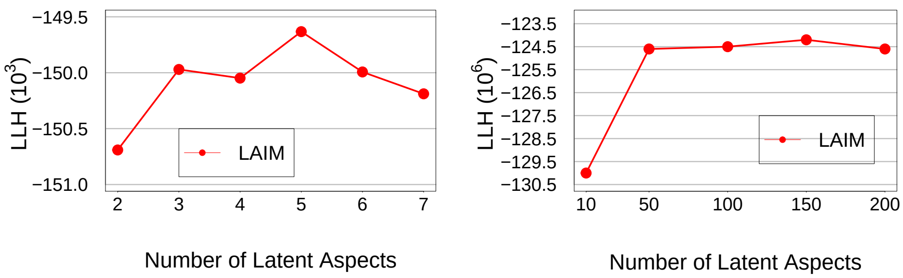
We compare the running time for the two models LAIM and OAIM. The results from LAIM10, LAIM20, and OAIM on *Twitter2000* data set with $\sim 2M$ tokens (Figure 9(a)) show that OAIM is much faster than LAIMx no matter whether the number of latent aspects A is 10 or 20. This result is obvious because OAIM does not need to sample latent aspect states. Furthermore, the running time for LAIM10 is less than that of LAIM20 confirming the fact that sampling more latent states needs more time. We also show how these two models scale with the size of graphs with five data sets containing 500, 1000, 1500, 2000, and 5000 users whose token numbers vary from $\sim 0.5M$ to $\sim 5M$. Figure 9(b) shows that both models scale linearly to the graph token numbers. This result is consistent with our analysis in Section 4.

Our experiments also show that both LAIM and OAIM converge, whose measurement is the $\hat{R}$ [6] of the parameters that we learn. The number of iterations before convergence for LAIM and OAIM on different sets of parameters and different size of iterations are similar, varying in the range of $[150, 167]$. As future work, we will analyze the convergence of both models theoretically and experimentally with more comprehensive parameter settings.

6 Related Work

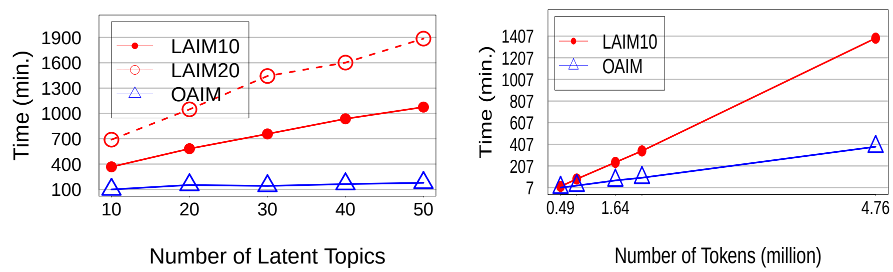
Topic models such as PLSI [10], Latent Dirichlet Allocation (LDA) model [2], and CTM [1] are designed to model single document sets. In topic models, every object is associated with a multinomial distribution over latent topics, each of which is a distribution over the observed tokens that are used to describe objects. Then, every token (or word) in an object description should be associated with several latent topics and can be virtually treated as being generated from a mixture of latent topics.

People have put effort to discover associations (i.e., connections) among graph nodes. The first group of works that are highly related to our research targets to discover influence degrees among objects. Dietz et al. [4] is one of the first few articles that leverage both the text and links in a citation graph into a probabilistic model to infer influence strength between articles by utilizing the LDA topic model. Liu et al. [15] proposed a very similar generative graphical



(a) *CiteMisc* (b) Twitter2000

Figure 8: LAIM log-likelihood vs. A ($Z = 10$)



(a) Time vs. Z (b) Time vs. graph size

Figure 9: Comparing running time of LAIM and OAIM

model to infer both direct and indirect top-level influence strengths between objects from heterogeneous graphs (e.g., citation graphs and co-authorship graphs). However, all these works do not consider influence aspects.

The second group of works are to identify the relationship types among objects in graphs. Diehl et al. [3] introduced the problem of identifying the types of relationships between communication parties from their communication content (e.g., messages), where the types of relationships come from a predefined set. The focus of the analysis is on the communication content associated with graph edges. Our work is different in that we focus on analyzing the content associated to graph nodes by also leveraging the structure of graphs. The work in [26] discussed techniques to detect one specific type of influence relationships, adviser-advisee relationship, from publication networks. Tang et al. [22] presented a framework to learn the social relationship types for edges in a target graph by utilizing other graphs (source graphs) with already labeled relationships. The relationship types in [22] are similar to influence aspects in our work. However, our work differs from [22] in that we do not assume the existence of known relationships from any graphs, which is given in [22].

There are other works that discover different types of influence relationships in graphs [18, 27]. Kataria et al. [13] study the problem of predicting the existence of citation relationships among documents. Their method utilizes the concept of citation contexts, which are the words occurring close to where a citation happens in the citing documents. [21] and [24] introduced generative process approaches to do social recommendations for the possible creation of new graph edges. Miao et al. [17] presented Latent Association Analysis (LAA) model to discover the association patterns in bipartite graphs. And the mining result is a correlation degree function between a source document and its corresponding target documents. [11] utilized cascade Poisson processes to learn the influence strength among users by analyzing special event sequences which consist of information about when users adopt products. [14] designed online algorithms to calculate influence strength among users by looking at a continuous stream of tweets only once.

7 Conclusions and Future Works

In this paper, we study the problem of detecting influence relationships at an aspect level from graphs. In particular, these influence relationships capture on which aspect (influence aspect) and how strong (influence degree) that one object influences another. We design two probabilistic models, OAIM and LAIM, to capture and represent these influence relationships. We developed block Gibbs sampling algorithms to learn the generative models. We show through case studies that these two models can generate meaningful results and extracts aspects. To objectively measure the effectiveness of the proposed models, we calculate log-likelihood of unseen data for the two models and a baseline approach. Extensive experimental results show that the proposed models can effectively discover influence relationships from graphs. Future works include further analyzing the choice of different parameter values and performing more subjective evaluations by letting different users give feedback on citation graphs.

8 Acknowledgements

We thank Mr. Kemin Chao for his help in collecting the *CiteMisc* data set.

9 Appendix

9.1 Derivation for Gibbs sampling update equation

We exemplify the Gibbs Sampling derivation for LAIM through the update Equation 3 for $p(a_{pos}|\vec{t}, \vec{t'}, \vec{z}, \vec{z'}, \vec{o'}, \vec{a}_{\neg pos}, \vec{t_a}, \vec{b})$. Equations for the other variables are derived analogously. Let $\Sigma = \{\vec{\alpha}_\phi, \vec{\alpha}_\theta, \vec{\alpha}_\eta, \vec{\alpha}_\psi, \vec{\alpha}_\gamma, \vec{\alpha}_\beta\}$. To start with, given an object o, the joint distribution of all the variables is represented as: $p(\vec{t}, \vec{t'}, \vec{z}, \vec{z'}, \vec{o'}, \vec{a}, \vec{t_a}, \vec{b}|\Sigma)$. It is rewritten as Equation 13 based on the LAIM graphical model.

$$
\begin{aligned}
&p(\vec{t}, \vec{t'}, \vec{z}, \vec{z'}, \vec{o'}, \vec{a}, \vec{t_a}, \vec{b}|\Sigma)\\
=&p(\vec{t}, \vec{t'}|\vec{z}, \vec{z'}; \vec{\alpha}_\phi) \cdot p(\vec{z}, \vec{z'}|\vec{o'}, \vec{b}; \vec{\alpha}_\theta)\\
&\cdot p(\vec{o'}|\vec{a}; \vec{\alpha}_\gamma) \cdot p(\vec{t}_a|\vec{a}; \vec{\alpha}_\psi) \cdot p(\vec{a}|\vec{\alpha}_\eta) \cdot p(\vec{b}|\vec{\alpha}_\beta)
\end{aligned} \tag{13}
$$

Since LAIM uses only conjugate priors, Equation 13 can be rewritten as Equation 14 (Rao-Blackwellized [5] version) by integrating out the multinomial distributions ϕ, θ, θ', γ, ψ, η, and β.

$$
\begin{aligned}
&p(\vec{t}, \vec{t'}, \vec{z}, \vec{z'}, \vec{o'}, \vec{a}, \vec{t_a}, \vec{b}|\Sigma)\\
=&\int p(\vec{t}, \vec{t'}|\vec{z}, \vec{z'}, \phi) \cdot p(\phi|\vec{\alpha}_\phi)\, d\phi\\
&\cdot \int p(\vec{z}, \vec{z'}|\vec{o'}, \vec{b}, \theta, \theta') \cdot p(\theta|\vec{\alpha}_\theta) \cdot p(\theta'|\vec{\alpha}_\theta)\, d\theta\theta'\\
&\cdot \int p(\vec{o'}|\vec{a}, \gamma) \cdot p(\gamma|\vec{\alpha}_\gamma)\, d\gamma \cdot \int p(\vec{t}_a|\vec{a}, \psi) \cdot p(\psi|\vec{\alpha}_\psi)\, d\psi\\
&\cdot \int p(\vec{a}|\vec{b}, \eta) \cdot p(\eta|\vec{\alpha}_\eta)\, d\eta \cdot \int p(\vec{b}|\beta) \cdot p(\beta|\alpha_{old}, \alpha_{new})\, d\beta
\end{aligned} \tag{14}
$$

$$
\begin{aligned}
&p(a_{pos}|\vec{t}, \vec{t'}, \vec{z}, \vec{z'}, \vec{o'}, \vec{a}_{\neg pos}, \vec{t}_a, \vec{b})\\
=&\frac{p(\vec{t}, \vec{t'}, \vec{z}, \vec{z'}, \vec{o'}, \vec{a}_{\neg pos}, a_{pos}, \vec{t}_a, \vec{b}|\Sigma)}{p(\vec{t}, \vec{t'}, \vec{z}, \vec{z'}, \vec{o'}, \vec{a}_{\neg pos}, \vec{t}_a, \vec{b}|\Sigma)}\\
=&\frac{\int p(\vec{o'}|\vec{a}, \gamma) \cdot p(\gamma|\vec{\alpha}_\gamma) d\gamma}{\int p(\vec{o'}|\vec{a}_{\neg pos}, \gamma) \cdot p(\gamma|\vec{\alpha}_\gamma) d\gamma} \cdot \frac{\int p(\vec{t}_a|\vec{a}, \psi) \cdot p(\psi|\vec{\alpha}_\psi)\, d\psi}{\int p(\vec{t}_a|\vec{a}_{\neg pos}, \psi) \cdot p(\psi|\vec{\alpha}_\psi)\, d\psi} \cdot \frac{\int p(\vec{a}|\vec{b}, \eta) \cdot p(\eta|\vec{\alpha}_\eta) d\eta}{\int p(\vec{a}_{\neg pos}|\vec{b}, \eta) \cdot p(\eta|\vec{\alpha}_\eta) d\eta}
\end{aligned} \tag{15}
$$

Now, we show the derivation of $p(a_{pos}|\vec{t}, \vec{t'}, \vec{z}, \vec{z'}, \vec{o'}, \vec{a}_{\neg pos}, \vec{t_a}, \vec{b})$. When we need to sample a at position pos (i.e., sample a_{pos}), it implies that the sample for b at that position b_{pos} must be one. Then, $p(a_{pos}|\vec{t}, \vec{t'}, \vec{z}, \vec{z'}, \vec{o'}, \vec{a}_{\neg pos}, \vec{t_a}, \vec{b})$ is equal to $p(a_{pos}|\vec{t}, \vec{t'}, \vec{z}, \vec{z'}, \vec{o'}, \vec{a}_{\neg pos}, \vec{t_a}, \vec{b}_{\neg pos}, b_{pos} = 1)$. The conditional of a_{pos} is obtained by dividing the joint distribution of all variables by the joint with all variables but a_{pos} (denoted by $\vec{a}_{\neg pos}$). Using Equation 14, we get Equation 15.

Let $\vec{\alpha} = (\alpha_1, \cdots, \alpha_N)$ and $\Delta(\vec{\alpha}) = \frac{\prod_{i=1}^{N} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{N} \alpha_i)}$, where $\Gamma(n) = \Gamma(n-1) \cdot (n-1)$. The integrals in Equation 15 can be derived as Equations 16, 18, and 17. After applying Equations 16, 18, and 17 to Equation 15, we can get Equation 19.

$$
\begin{aligned}
&\int p(\vec{o'}|\vec{a}, \gamma) \cdot p(\gamma|\vec{\alpha}_\gamma)\, d\gamma\\
=&\int \prod_{a=1}^{A} \Big(\prod_{o' \in R(o)} \gamma_{o,a,o'}^{N_{o,a,o',b^{(o,a,o',1)}}} \cdot \frac{1}{\Delta(\vec{\alpha}_\gamma)} \prod_{o' \in R(o)} \gamma_{o,a,o'}^{\alpha_\gamma - 1} d\vec{\gamma}_{o,a} \Big)\\
=&\int \prod_{a=1}^{A} \frac{1}{\Delta(\vec{\alpha}_\gamma)} \prod_{o' \in R(o)} \gamma_{o,a,o'}^{N_{o,a,o',b^{(o,a,o',1)}} + \alpha_\gamma - 1} d\vec{\gamma}_{o,a}\\
=&\prod_{a=1}^{A} \frac{\Delta(\vec{N}_{o,a,o',b=1} + \vec{\alpha}_\gamma)}{\Delta(\vec{\alpha}_\gamma)}
\end{aligned} \tag{16}
$$

where $\vec{\gamma}_{o,a}$ is a vector with length $R(o)$.

$$\begin{aligned}&\int p(\vec{t}_a|\vec{a},\psi)\cdot p(\psi|\vec{\alpha}_\psi)\,d\psi\\ &=\int\prod_{t_a=1}^{T_a}\psi_{a,t_a}^{N_{a,t_a,b}(a,t_a,1)}\cdot\frac{1}{\Delta(\vec{\alpha}_\psi)}\prod_{t_a=1}^{T_a}\psi_{a,t_a}^{\alpha_\psi-1}d\psi=\frac{\Delta(\vec{N}_{a,t_a,b=1}+\vec{\alpha}_\psi)}{\Delta(\vec{\alpha}_\psi)}\end{aligned}\tag{17}$$

$$\begin{aligned}&\int p(\vec{a}|\vec{b},\eta)\cdot p(\eta|\vec{\alpha}_\eta)\,d\eta\\ &=\int\prod_{a=1}^{A}\eta_{o,a}^{N_{o,a,b}(o,a,1)}\cdot\frac{1}{\Delta(\vec{\alpha}_\eta)}\prod_{a=1}^{A}\eta_{o,a}^{\alpha_\eta-1}d\eta\\ &=\int\frac{1}{\Delta(\vec{\alpha}_\eta)}\cdot\prod_{a=1}^{A}\eta_{o,a}^{N_{o,a,b}(o,a,1)+\alpha_\eta-1}d\eta=\frac{\Delta(\vec{N}_{o,a,b=1}+\vec{\alpha}_\eta)}{\Delta(\vec{\alpha}_\eta)}\end{aligned}\tag{18}$$

$$\begin{aligned}&p(a_{pos}|\vec{t},\vec{t'},\vec{z},\vec{z'},\vec{o'},\vec{a}_{\neg pos},\vec{t}_a,\vec{b})\\ &\propto\frac{\prod_{a=1}^{A}\frac{\Delta(\vec{N}_{o,a,o',b=1}+\vec{\alpha}_\gamma)}{\Delta(\vec{\alpha}_\gamma)}}{\prod_{a=1}^{A}\frac{\Delta(\vec{N}_{o,a_{\neg pos},o',b=1}+\vec{\alpha}_\gamma)}{\Delta(\vec{\alpha}_\gamma)}}\cdot\frac{\frac{\Delta(\vec{N}_{a,t_a,b=1}+\vec{\alpha}_\psi)}{\Delta(\vec{\alpha}_\psi)}}{\frac{\Delta(\vec{N}_{a,t_{a_{\neg pos}},b=1}+\vec{\alpha}_\psi)}{\Delta(\vec{\alpha}_\psi)}}\cdot\frac{\frac{\Delta(\vec{N}_{o,a,b=1}+\vec{\alpha}_\eta)}{\Delta(\vec{\alpha}_\eta)}}{\frac{\Delta(\vec{N}_{o,a_{\neg pos},b=1}+\vec{\alpha}_\eta)}{\Delta(\vec{\alpha}_\eta)}}\\ &=\frac{\Delta(\vec{N}_{o,a=a_{pos},o',b=1}+\vec{\alpha}_\gamma)}{\Delta(\vec{N}_{o,a_{\neg pos},o',b=1}+\vec{\alpha}_\gamma)}\cdot\frac{\Delta(\vec{N}_{a,t_a,b=1}+\vec{\alpha}_\psi)}{\Delta(\vec{N}_{a,t_{a_{\neg pos}},b=1}+\vec{\alpha}_\psi)}\cdot\frac{\Delta(\vec{N}_{o,a,b=1}+\vec{\alpha}_\eta)}{\Delta(\vec{N}_{o,a_{\neg pos},b=1}+\vec{\alpha}_\eta)}\end{aligned}\tag{19}$$

The three components in Equation 19 can be simplified by canceling the common factors in the Δ function. We utilize the the Kronecker delta function $\delta(x,y)$ which is 1 iff $x=y$, and is 0 otherwise.

$$\begin{aligned}&\frac{\Delta(\vec{N}_{o,a,b=1}+\vec{\alpha}_\eta)}{\Delta(\vec{N}_{o,a_{\neg pos},b=1}+\vec{\alpha}_\eta)}=\frac{\frac{\prod_{a=1}^{A}\Gamma(N_{o,a,b}(o,a,1)+\alpha_\eta)}{\Gamma(\sum_{a=1}^{A}(N_{o,a,b}(o,a,1)+\alpha_\eta))}}{\frac{\prod_{a=1}^{A}\Gamma(N_{o,a,b}(o,a,1)+\alpha_\eta-\delta(a_{pos},a))}{\Gamma(\sum_{a=1}^{A}(N_{o,a,b}(o,a,1)+\alpha_\eta-\delta(a_{pos},a)))}}\\ &=\frac{N_{o,a,b}(o,a_{pos},1)+\alpha_\eta-1}{N_{o,b}(o,1)+A\alpha_\eta-1}\end{aligned}\tag{20}$$

We can apply a similar procedure to simply other components in Equation 19 with the Δ functions and get the final update equation shown as Equation 21.

$$\begin{aligned}&p(a_{pos}|\vec{t},\vec{t'},\vec{z},\vec{z'},\vec{o'},\vec{a}_{\neg pos},\vec{t}_a,\vec{b})\\ &=\frac{N_{o,a,o',b}(o,a_{pos},o'_{pos},1)+\alpha_\gamma-1}{N_{o,a,b}(o,a_{pos},1)+|R(o)|\alpha_\gamma-1}\cdot\frac{N_{a,t_a,b}(a_{pos},t_{a_{pos}},1)+\alpha_\psi-1}{N_{a,b}(a_{pos},1)+T_a\alpha_\psi-1}\cdot\frac{N_{o,a,b}(o,a_{pos},1)+\alpha_\eta-1}{N_{o,b}(o,1)+A\alpha_\eta-1}\end{aligned}\tag{21}$$

References

[1] D. M. Blei and J. D. Lafferty. Correlated topic models. In *NIPS*, 2005.

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

[3] C. P. Diehl, G. Namata, and L. Getoor. Relationship identification for social network discovery. In *AAAI*, 2007.

[4] L. Dietz, S. Bickel, and T. Scheffer. Unsupervised prediction of citation influences. In *ICML*, pages 233–240, 2007.

[5] A. Doucet, N. de Freitas, K. P. Murphy, and S. J. Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In *Proc. of Conf. in Uncertaity in Artificial Intelligence (UAI)*, pages 176–183, 2000.

[6] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis, Second Edition (Chapman & Hall/CRC Texts in Statistical Science)*. Chapman and Hall/CRC, 2 edition, July 2003.

[7] W. Gilks, S. Richardson, and D. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman & Hall, 1996.

[8] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl. 1):5228–5235, April 2004.

[9] G. Heinrich. Parameter estimation for text analysis. Technical report, 2004.

[10] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR*, pages 50–57, 1999.

[11] T. Iwata, A. Shah, and Z. Ghahramani. Discovering latent influence in online social activities via shared cascade poisson processes. In *KDD*, pages 266–274, 2013.

[12] C. S. Jensen, U. Kjærulff, and A. Kong. Blocking gibbs sampling in very large probabilistic expert systems. *Int. J. Hum.-Comput. Stud.*, 42(6):647–666, 1995.

[13] S. Kataria, P. Mitra, and S. Bhatia. Utilizing context in generative bayesian models for linked corpus. In *AAAI*, 2010.

[14] K. Kutzkov, A. Bifet, F. Bonchi, and A. Gionis. Strip: stream learning of influence probabilities. In *KDD*, pages 275–283, 2013.

[15] L. Liu, J. Tang, J. Han, M. Jiang, and S. Yang. Mining topic-level influence in heterogeneous networks. In *CIKM*, 2010.

[16] Y. Liu, H. Cao, Y. Hao, P. Han, and X. Zeng. Discovering context-aware influential objects. In *SDM*, 2012.

[17] G. Miao, Z. Guan, L. E. Moser, X. Yan, S. Tao, N. Anerousis, and J. Sun. Latent association analysis of document pairs. In *KDD*, pages 1415–1423, 2012.

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

[19] R. Nallapati, A. Ahmed, E. P. Xing, and W. W. Cohen. Joint latent topic models for text and citations. In *KDD*, 2008.

[20] M. Rosen-Zvi, T. L. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *UAI*, 2004.

[21] Y. Shen and R. Jin. Learning personal + social latent factor model for social recommendation. In *KDD*, 2012.

[22] J. Tang, T. Lou, and J. M. Kleinberg. Inferring social ties across heterogenous networks. In *WSDM*, 2012.

[23] J. Tang, J. Sun, C. Wang, and Z. Yang. Social influence analysis in large-scale networks. In *KDD*, 2009.

[24] J. Tang, S. Wu, J. Sun, and H. Su. Cross-domain collaboration recommendation. In *KDD*, pages 1285–1293, 2012.

[25] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101, 2004.

[26] C. Wang, J. Han, Y. Jia, J. Tang, D. Zhang, Y. Yu, and J. Guo. Mining advisor-advisee relationships from research publication networks. In *KDD*, pages 203–212, 2010.

[27] G. Wang, Q. Hu, and P. S. Yu. Influence and similarity on heterogeneous networks. In *CIKM*, pages 1462–1466, 2012.