Scalable Text and Link Analysis with Mixed-Topic Link Models

Yaojia Zhu
University of New Mexico
yaojia.zhu@gmail.com

Xiaoran Yan
University of New Mexico
everyxt@gmail.com

Lise Getoor
University of Maryland
getoor@cs.umd.edu

Cristopher Moore
Santa Fe Institute
moore@santafe.edu

ABSTRACT

Many data sets contain rich information about objects, as well as pairwise relations between them. For instance, in networks of websites, scientific papers, and other documents, each node has content consisting of a collection of words, as well as hyperlinks or citations to other nodes. In order to perform inference on such data sets, and make predictions and recommendations, it is useful to have models that are able to capture the processes which generate the text at each node and the links between them. In this paper, we combine classic ideas in topic modeling with a variant of the mixed-membership block model recently developed in the statistical physics community. The resulting model has the advantage that its parameters, including the mixture of topics of each document and the resulting overlapping communities, can be inferred with a simple and scalable expectation-maximization algorithm. We test our model on three data sets, performing unsupervised topic classification and link prediction. For both tasks, our model outperforms several existing state-of-the-art methods, achieving higher accuracy with significantly less computation, analyzing a data set with 1.3 million words and 44 thousand links in a few minutes.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; I.2 [**ARTIFICIAL INTELLIGENCE**]: Learning

Keywords

Document classification; Topic modeling; Link prediction; Stochastic block model

1. INTRODUCTION

Many modern data sets contain not only rich information about each object, but also pairwise relationships between them, forming networks where each object is a node and links represent the relationships. In document networks, for example, each node is a document containing a sequence of words, and the links between nodes are citations or hyperlinks. Both the content of the documents and the topology of the links between them are meaningful.

Over the past few years, two disparate communities have been approaching these data sets from different points of view. In the data mining community, the goal has been to augment traditional approaches to learning and data mining by including relations between objects [15, 23, 33]: for instance, using the links between documents to help us label them by topic. In the network community, including its subset in statistical physics, the goal has been to augment traditional community structure algorithms such as the stochastic block model [14, 19, 30] by taking node attributes into account: for instance, to use the content of documents, rather than just the topological links between them, to help us understand their community structure.

In the original stochastic block model, each node has a discrete label, assigning it to one of k communities. These labels, and the $k \times k$ matrix of probabilities with which a given pair of nodes with a given pair of labels have a link between them, can be inferred using Monte Carlo algorithms (e.g. [26]) or, more efficiently, with belief propagation [12, 11] or pseudolikelihood approaches [7]. However, in real networks communities often overlap, and a given node can belong to multiple communities. This led to the *mixed-membership* block model [1], where the goal is to infer, for each node v, a distribution or mixture of labels θ_v describing to what extent it belongs to each community. If we assume that links are assortative, i.e., that nodes are more likely to link to others in the same community, then the probability of a link between two nodes v and v' depends on some measure of similarity (say, the inner product) of θ_v and $\theta_{v'}$.

These mixed-membership block models fit nicely with classic ideas in topic modeling. In models such as Probabilistic Latent Semantic Analysis (PLSA) [18] and Latent Dirichlet Allocation (LDA) [4], each document d has a mixture θ_d of topics. Each topic corresponds in turn to a probability distribution over words, and each word in d is generated independently from the resulting mixture of distributions. If we think of θ_d as both the mixture of topics for generating words and the mixture of communities for generating links, then we can infer $\{\theta_d\}$ jointly from the documents' content and the presence or absence of links between them.

KDD'13, August 11–14, 2013, Chicago, Illinois, USA.
Copyright 2013 ACM 978-1-4503-2174-7/13/08 ...$15.00.

There are many possible such models, and we are far from the first to think along these lines. Our innovation is to take as our starting point a particular mixed-membership block model recently developed in the physics community [2], which we call the BKN model. It differs from the mixed-membership stochastic block model (MMSB) of [1] in several ways:

1. The BKN model treats the community membership mixtures θ_d directly as parameters to be inferred. In contrast, MMSB treats θ_d as hidden variables generated by a Dirichlet distribution, and infers the hyperparameters of that distribution. The situation between PLSA and LDA is similar; PLSA infers the topic mixtures θ_d, while LDA generates them from a Dirichlet distribution.
2. The MMSB model generates each link according to a Bernoulli distribution, with an extra parameter for sparsity. Instead, BKN treats the links as a random multigraph, where the number of links $A_{dd'}$ between each pair of nodes is Poisson-distributed. As a result, the derivatives of the log-likelihood with respect to θ_d and the other parameters are particularly simple.

These two factors make it possible to fit the BKN model using an efficient and exact expectation-maximization (EM) algorithm, making its inference highly scalable. The BKN model has another advantage as well:

3. The BKN model is *degree-corrected*, in that it takes the observed degrees of the nodes into account when computing the expected number of edges between them. Thus it recognizes that two documents that have very different degrees might in fact have the same mix of topics; one may simply be more popular than the other.

In our work, we use a slight variant of the BKN model to generate the links, and we use PLSA to generate the text. We present an EM algorithm for inferring the topic mixtures and other parameters. (While we do not impose a Dirichlet prior on the topic mixtures, it is easy to add a corresponding term to the update equations.) Our algorithm is scalable in the sense that each iteration takes $O(K(N+M+R))$ time for networks with K topics, N documents, and M links, where R is the sum over documents of the number of distinct words appearing in each one. In practice, our EM algorithm converges within a small number of iterations, making the total running time linear in the size of the corpus.

Our model can be used for a variety of learning and generalization tasks, including document classification or link prediction. For document classification, we can obtain hard labels for each document by taking its most-likely topic with respect to θ_d, and optionally improve these labels further with local search. For link prediction, we train the model using a subset of the links, and then ask it to rank the remaining pairs of documents according to the probability of a link between them. For each task we determine the optimal relative weight of the content vs. the link information.

We performed experiments on three real-world data sets, with thousands of documents and millions of words. Our results show that our algorithm is more accurate, and considerably faster, than previous techniques for both document classification and link prediction.

The rest of the paper is organized as follows. Section 2 describes our generative model, and compares it with related models in the literature. Section 3 gives our EM algorithm and analyzes its running time. Section 4 contains our experimental results for document classification and link prediction, comparing our accuracy and running time with other techniques. In Section 5, we conclude, and offer some directions for further work.

2. OUR MODEL AND PREVIOUS WORK

In this section, we give our proposed model, which we call the *Poisson mixed-topic link model* (PMTLM) and its degree-corrected variant PMTLM-DC.

2.1 The Generative Model

Consider a network of N documents. Each document d has a fixed length L_d, and consists of a string of words $w_{d\ell}$ for $1 \le \ell \le L_d$, where $1 \le w_{d\ell} \le W$ where W is the number of distinct words. In addition, each pair of documents d, d' has an integer number of links connecting them, giving an adjacency matrix $A_{dd'}$. There are K topics, which play the dual role of the overlapping communities in the network.

Our model generates both the content $\{w_{d\ell}\}$ and the links $\{A_{dd'}\}$ as follows. We generate the content using the PLSA model [18]. Each topic z is associated with a probability distribution β_z over words, and each document has a probability distribution θ_d over topics. For each document $1 \le d \le N$ and each $1 \le \ell \le L_d$, we independently choose a topic $z = z_{d\ell} \sim \text{Multi}(\theta_d)$, and choose the word $w_{d\ell} \sim \text{Multi}(\beta_z)$. Thus the total probability that $w_{d\ell}$ is a given word w is

$$\Pr[w_{d\ell} = w] = \sum_{z=1}^{K} \theta_{dz} \beta_{zw} \, . \tag{1}$$

We assume that the number of topics K is fixed. The distributions β_z and θ_d are parameters to be inferred.

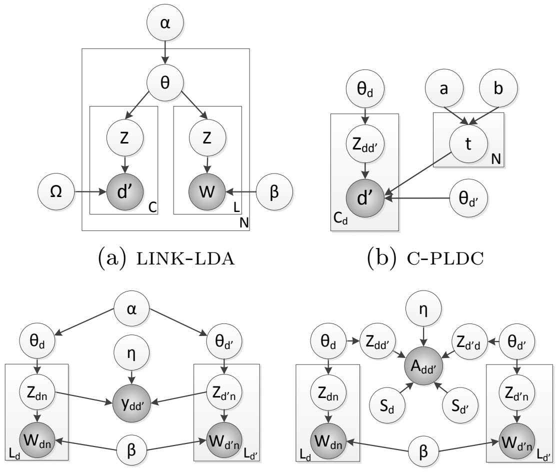
We generate the links using a version of the Ball-Karrer-Newman (BKN) model [2]. Each topic z is associated with a link density η_z. For each pair of documents d, d' and each topic z, we independently generate a number of links which is Poisson-distributed with mean $\theta_{dz}\theta_{d'z}\eta_z$. Since the sum of independent Poisson variables is Poisson, the total number of links between d and d' is distributed as

$$A_{dd'} \sim \text{Poi}\left(\sum_z \theta_{dz} \theta_{d'z} \eta_z \right) \, . \tag{2}$$

Since $A_{dd'}$ can exceed 1, this gives a random multigraph. In the data sets we study below, $A_{dd'}$ is 1 or 0 depending on whether d cites d', giving a simple graph. On the other hand, in the sparse case the event that $A_{dd'} > 1$ has low probability in our model. Moreover, the fact that $A_{dd'}$ is Poisson-distributed rather than Bernoulli makes the derivatives of the likelihood with respect to the parameters θ_{dz} and η_z very simple, allowing us to write down an efficient EM algorithm for inferring them.

This version of the model assumes that links are assortative, i.e., that links between documents only form to the extent that they belong to the same topic. One can easily generalize the model to include disassortative links as well, replacing η_z with a matrix $\eta_{zz'}$ that allows documents with distinct topics z, z' to link [2].

We also consider *degree-corrected* versions of this model, where in addition to its topic mixture θ_d, each document



(a) LINK-LDA (b) C-PLDC

(c) RTM (d) PMTLM-DC

Figure 1: Graphical models for link generation.

has a propensity S_d of forming links. In that case,

$$A_{dd'} \sim \text{Poi}\left(S_d S_{d'} \sum_z \theta_{dz}\theta_{d'z}\eta_z\right) . \tag{3}$$

We call this variant the *Poisson Mixed-Topic Link Model with Degree Correction* (PMTLM-DC).

2.2 Prior Work on Content–Link Models

Most models for document networks generate content using either PLSA [18], as we do, or LDA [4]. The distinction is that PLSA treats the document mixtures θ_d as parameters, while in LDA they are hidden variables, integrated over a Dirichlet distribution. As we show in Section 3, our approach gives a simple, exact EM algorithm, avoiding the need for sampling or variational methods. While we do not impose a Dirichlet prior on θ_d in this paper, it is easy to add a corresponding term to the update equations for the EM algorithm, with no loss of efficiency.

There are a variety of methods in the literature to generate links between documents. PHITS-PLSA [10], LINK-LDA [13] and LINK-PLSA-LDA [27] use the PHITS [9] model for link generation. PHITS treats each document as an additional term in the vocabulary, so two documents are similar if they link to the same documents. This is analogous to a mixture model for networks studied in [28]. In contrast, block models like ours treat documents as similar if they link to *similar* documents, as opposed to literally the same ones.

The PAIRWISE LINK-LDA model [27], like ours, generates the links with a mixed-topic block model, although as in MMSB [1] and LDA [4] it treats the θ_d as hidden variables integrated over a Dirichlet prior. They fit their model with a variational method that requires N^2 parameters, making it less scalable than our approach.

In the C-PLDC model [32], the link probability from d to d' is determined by their topic mixtures $\theta_d, \theta_{d'}$ and the popularity $t_{d'}$ of d', which is drawn from a Gamma distribution with hyperparameters a and b. Thus $t_{d'}$ plays a role similar to the degree-correcting parameter $S_{d'}$ in our model, although we correct for the degree of d as well. However, C-PLDC does not generate the content, but takes it as given.

The Relational Topic Model (RTM) [5, 6] assumes that the link probability between d and d' depends on the topics of the words appearing in their text. In contrast, our model uses the underlying topic mixtures θ_d to generate both the content and the links. Like our model, RTM defines the similarity of two topics as a weighted inner product of their topic mixtures: however, in RTM the probability of a link is a nonlinear function of this similarity, which can be logistic, exponential or normal, of this similarity.

Although it deals with a slightly different kind of dataset, our model is closest in spirit to the Latent Topic Hypertext Model (LTHM) [17]. This is a generative model for hypertext networks, where each link from d to d' is associated with a specific word w in d. If we sum over all words in d, the total number of links $A_{dd'}$ from d to d' that LTHM would generate follows a binomial distribution

$$A_{dd'} \sim \text{Bin}\left(L_d, \lambda_{d'} \sum_z \theta_{dz}\theta_{d'z}\right) , \tag{4}$$

where $\lambda_{d'}$ is, in our terms, a degree-correction parameter. When L_d is large this becomes a Poisson distribution with mean $L_d \lambda_{d'} \sum_z \theta_{dz}\theta_{d'z}$. Our model differs from this in two ways: our parameters η_z give a link density associated with each topic z, and our degree correction S_d does not assume that the number of links from d is proportional to its length.

We briefly mention several other approaches. The authors of [15] extend the probabilistic relational model (PRM) framework and proposed a unified generative model for both content and links in a relational structure. In [24], the authors proposed a link-based model that describes both node attributes and links. The HTM model [31] treats links as fixed rather than generating them, and only generates the text. Finally, the LMMG model [21] treats the appearance or absence of a word as a binary attribute of each document, and uses a logistic or exponential function of these attributes to determine the link probabilities.

In Section 4 below, we compare our model to PHITS-PLSA, LINK-LDA, C-PLDC, and RTM. Graphical models for the link generation components of these models, and ours, are shown in Figure 1.

3. A SCALABLE EM ALGORITHM

Here we describe an efficient Expectation-Maximization algorithm to find the maximum-likelihood estimates of the parameters of our model. Each update takes $O(K(N+M+R))$ time for a document network with K topics, N documents, and M links, where R is the sum over the documents of the number of distinct words in each one. Thus the running time per iteration is linear in the size of the corpus.

For simplicity we describe the algorithm for the simpler version of our model, PMTLM. The algorithm for the degree-corrected version, PMTLM-DC, is similar.

3.1 The likelihood

Let C_{dw} denote the number of times a word w appears in document d. From (1), the log-likelihood of d's content is

$$\begin{aligned}\mathcal{L}_d^{\text{content}} &= \log P(w_{d1}, \ldots, w_{dL_d} \,|\, \theta_d, \beta) \\ &= \sum_{w=1}^{W} C_{dw} \log\left(\sum_{z=1}^{K} \theta_{dz}\beta_{zw}\right) .\end{aligned} \tag{5}$$

Similarly, from (2), the log-likelihood for the links $A_{dd'}$ is

$$\begin{aligned}\mathcal{L}^{\text{links}} &= \log P(A \mid \theta, \eta) \\ &= \frac{1}{2} \sum_{dd'} A_{dd'} \log \left(\sum_{z} \theta_{dz} \theta_{d'z} \eta_z \right) \\ &\quad - \frac{1}{2} \sum_{dd'} \sum_{z} \theta_{dz} \theta_{d'z} \eta_z \, . \end{aligned} \tag{6}$$

We ignore the constant term $-\sum_{dd'} \log A_{dd'}!$ from the denominator of the Poisson distribution, since it has no bearing on the parameters.

3.2 Balancing Content and Links

While we can use the total likelihood $\sum_d \mathcal{L}_d^{\text{content}} + \mathcal{L}^{\text{links}}$ directly, in practice we can improve our performance significantly by better balancing the information in the content vs. that in the links. In particular, the log-likelihood $\mathcal{L}_d^{\text{content}}$ of each document is proportional to its length, while its contribution to $\mathcal{L}^{\text{links}}$ is proportional to its degree. Since a typical document has many more words than links, $\mathcal{L}^{\text{content}}$ tends to be much larger than $\mathcal{L}^{\text{links}}$.

Following [18], we can provide this balance in two ways. One is to normalize $\mathcal{L}^{\text{content}}$ by the length L_d, and another is to add a parameter α that reweights the relative contributions of the two terms $\mathcal{L}^{\text{content}}$ and $\mathcal{L}^{\text{links}}$. We then maximize

$$\mathcal{L} = \alpha \sum_{d} \frac{1}{L_d} \mathcal{L}_d^{\text{content}} + (1 - \alpha) \mathcal{L}^{\text{links}} \, . \tag{7}$$

Varying α from 0 to 1 lets us interpolate between two extremes: studying the document network purely in terms of its topology, or purely in terms of the documents' content. Indeed, we will see in Section 4 that the optimal value of α depends on which task we are performing: closer to 0 for link prediction, and closer to 1 for topic classification.

3.3 Update Equations and Running Time

We maximize $\mathcal{L}$ as a function of $\{\theta, \beta, \eta\}$ using an EM algorithm, very similar to the one introduced by [2] for overlapping community detection. We start with a standard trick to change the log of a sum into a sum of logs, writing

$$\begin{aligned}\mathcal{L}_d^{\text{content}} &\geq \sum_{w=1}^{W} C_{dw} \sum_{z=1}^{K} h_{dw}(z) \log \frac{\theta_{dz} \beta_{zw}}{h_{dw}(z)} \\ \mathcal{L}^{\text{links}} &\geq \frac{1}{2} \sum_{dd'} \sum_{z=1}^{K} A_{dd'} q_{dd'}(z) \log \frac{\theta_{dz} \theta_{d'z} \eta_z}{q_{dd'}(z)} \\ &\quad - \frac{1}{2} \sum_{dd'} \sum_{z=1}^{K} \theta_{dz} \theta_{d'z} \eta_z \, . \end{aligned} \tag{8}$$

Here $h_{dw}(z)$ is the probability that a given appearance of w in d is due to topic z, and $q_{dd'}(z)$ is the probability that a given link from d and d' is due to topic z. This lower bound holds with equality when

$$h_{dw}(z) = \frac{\theta_{dz} \beta_{zw}}{\sum_{z'} \theta_{dz'} \beta_{z'w}} \, , \; q_{dd'}(z) = \frac{\theta_{dz} \theta_{d'z} \eta_z}{\sum_{z'} \theta_{dz'} \theta_{d'z'} \eta_{z'}} \, , \tag{9}$$

giving us the E step of the algorithm.

For the M step, we derive update equations for the parameters $\{\theta, \beta, \eta\}$. By taking derivatives of the log-likelihood (7) (see the online version for details) we obtain

$$\eta_z = \frac{\sum_{dd'} A_{dd'} q_{dd'}(z)}{\left(\sum_d \theta_{dz}\right)^2} \tag{10}$$

$$\beta_{zw} = \frac{\sum_d (1/L_d) \, C_{dw} h_{dw}(z)}{\sum_d (1/L_d) \sum_{w'} C_{dw'} h_{dw'}(z)} \tag{11}$$

$$\theta_{dz} = \frac{(\alpha/L_d) \sum_w C_{dw} h_{dw}(z) + (1-\alpha) \sum_{d'} A_{dd'} q_{dd'}(z)}{\alpha + (1-\alpha)\kappa_d} \, . \tag{12}$$

Here $\kappa_d = \sum_{d'} A_{dd'}$ is the degree of document d.

To analyze the running time, let R_d denote the number of distinct words in document d, and let $R = \sum_d R_d$. Then only KR of the parameters $h_{dw}(z)$ are nonzero. Similarly, $q_{dd'}(z)$ only appears if $A_{dd'} \neq 0$, so in a network with M links only KM of the $q_{dd'}(z)$ are nonzero. The total number of nonzero terms appearing in (9)–(12), and hence the running time of the E and M steps, is thus $O(K(N{+}M{+}R))$.

As in [2], we can speed up the algorithm if θ is sparse, i.e. if many documents belong to fewer than K topics, so that many of the θ_{dz} are zero. According to (9), if $\theta_{dz} = 0$ then $h_{d\ell}(z) = q_{dd'}(z) = 0$, in which case (12) implies that $\theta_{dz} = 0$ for all future iterations. If we choose a threshold below which θ_{dz} is effectively zero, then as θ becomes sparser we can maintain just those $h_{d\ell}(z)$ and $q_{dd'}(z)$ where $\theta_{dz} \neq 0$. This in turn simplifies the updates for η and β in (10) and (11).

We note that the simplicity of our update equations comes from the fact that the $A_{dd'}$ is Poisson, and that its mean is a multilinear function of the parameters. Models where $A_{dd'}$ is Bernoulli-distributed with a more complicated link probability, such as a logistic function, have more complicated derivatives of the likelihood, and therefore more complicated update equations.

Note also that this EM algorithm is exact, in the sense that the maximum-likelihood estimators $\{\widehat{\theta}, \widehat{\beta}, \widehat{\eta}\}$ are fixed points of the update equations. This is because the E step (9) is exact, since the conditional distribution of topics associated with each word occurrence and each link is a product distribution, which we can describe exactly with h_{dw} and $q_{dd'}$. (There are typically multiple fixed points, so in practice we run our algorithm with many different initial conditions, and take the fixed point with the highest likelihood.)

This exactness is due to the fact that the topic mixtures θ_d are parameters to be inferred. In models such as LDA and MMSB where θ_d is a hidden variable integrated over a Dirichlet prior, the topics associated with each word and link have a complicated joint distribution that can only be approximated using sampling or variational methods. (To be fair, recent advances such as stochastic optimization based on network subsampling [16] have shown that approximate inference in these models can be carried out quite efficiently.)

On the other hand, in the context of finding communities in networks, models with Dirichlet priors have been observed to generalize more successfully than Poisson models such as BKN [16]. Happily, we can impose a Dirichlet prior on θ_d with no loss of efficiency, simply by including pseudocounts in the update equations—in essence adding additional words and links that are known to come from each topic. This lets us obtain a maximum a posteriori (MAP) estimate of an LDA-like model. We leave this as a direction for future work.

3.4 Discrete Labels and Local Search

Our model, like PLSA and the BKN model, lets us infer a soft classification—a mixture of topic labels or community memberships for each document. However, we often want to infer categorical labels, where each document d is assigned to a single topic $1 \leq z_d \leq K$. A natural way to do this is to let z_d be the most-likely label in the inferred mixture, $\hat{z}_d = \text{argmax}_z\, \theta_{dz}$. This is equivalent to rounding θ_d to a delta function, $\theta_{dz} = 1$ for $z = \hat{z}_d$ and 0 for $z \neq \hat{z}_d$.

If we wish, we can improve these discrete labels further using local search. If each document has just a single topic, the log-likelihood of our model is

$$\mathcal{L}_d^{\text{content}} = \sum_{w=1}^{W} C_{dw} \log \beta_{z_d w} \tag{13}$$

$$\mathcal{L}^{\text{links}} = \frac{1}{2} \sum_{dd'} A_{dd'} \log \eta_{z_d z_{d'}} \,. \tag{14}$$

Note that here η is a matrix, with off-diagonal entries that allow documents with different topics $z_d, z_{d'}$ to be linked. Otherwise, these discrete labels would cause the network to split into K separate components.

Let n_z denote the number of documents of topic z, let $L_z = \sum_{d:z_d=z} L_d$ be their total length, and let $C_{zw} = \sum_{d:z_d=z} C_{dw}$ be the total number of times w appears in them. Let $m_{zz'}$ denote the total number of links between documents of topics z and z', counting each link twice if $z = z'$. Then the MLEs for β and η are

$$\hat{\beta}_{zw} = \frac{C_{zw}}{L_z} \,, \; \hat{\eta}_{zz'} = \frac{m_{zz'}}{n_z n_{z'}} \,. \tag{15}$$

Applying these MLEs in (13) and (14) gives us a point estimate of the likelihood of a discrete topic assignment z_d, which we can normalize or reweight as discussed in Section 3.2 if we like. We can then maximize this likelihood using local search: for instance, using the Kernighan-Lin heuristic as in [20] or a Monte Carlo algorithm to find a local maximum of the likelihood in the vicinity of $\hat{z}$. Each step of these algorithms changes the label of a single document d, so we can update the values of n_z, L_z, C_{zw}, and $m_{zz'}$ and compute the new likelihood in $O(K + R_d + \kappa_d)$ time. In our experiments we used the KL heuristic, and found that for some data sets it noticeably improved the accuracy of our algorithm for the document classification task.

4. EXPERIMENTAL RESULTS

In this section we present empirical results on our model and our algorithm for unsupervised document classification and link prediction. We compare its accuracy and running time with those of several other methods, testing it on three real-world document citation networks.

4.1 Data Sets

The top portion of Table 1 lists the basic statistics for three real-world corpora [29]: Cora, Citeseer, and PubMed[1]. Cora and Citeseer contain papers in machine learning, with $K = 7$ topics for Cora and $K = 6$ for Citeseer. PubMed consists of medical research papers on $K = 3$ topics, namely three types of diabetes. All three corpora have ground-truth topic labels provided by human curators.

The data sets for the three corpora are slightly different. PubMed contains the number of times C_{dw} each word appeared in each document, while Cora and Citeseer record whether or not a word occurred at least once in the document. For Cora and Citeseer, we treat C_{dw} as 0 or 1.

4.2 Models and Implementations

We compare the Poisson Mixed-Topic Link Model (PMTLM) and its degree-corrected variant, denoted PMTLM-DC, with PHITS-PLSA, LINK-LDA, C-PLDC, and RTM (see Section 2.2). We used our own implementation of both PHITS-PLSA and RTM. For RTM, we implemented the variational EM algorithm given in [6]. The implementation is based on the LDA code available from the authors[2]. We also tried the code provided by J. Chang[3], which uses a Monte Carlo algorithm for the E step, but we found the variational algorithm works better on our data sets. While RTM includes a variety of link probability functions, we only used the sigmoid function. We also assume a symmetric Dirichlet prior. The results for LINK-LDA and C-PLDC are taken from [32].

Each E and M step of the variational algorithm for RTM performs multiple iterations until they converge on estimates for the posterior and the parameters [6]. This is quite different from our EM algorithm: since our E step is exact, we update the parameters only once in each iteration. Our convergence condition for the E step and for the entire EM algorithm are that the fractional increase of the log-likelihood between iterations is less than 10^{-6}; we performed a maximum of 50 iterations in each E step and a maximum of 500 EM iterations for the entire algorithm. To optimize the η parameters (see the graphical model in Section 2.2) RTM uses a tunable regularization parameter ρ, which can be thought of as the number of observed non-links. We tried various settings for ρ, namely $0.1M, 0.2M, 0.5M, M, 2M, 5M$ and $10M$ where M is the number of observed links, and tuned ρ separately for each data set and each task. We used gradient descent to optimize the η parameters in each M step.

As described in Section 3.2, for PMTLM, PMTLM-DC and PHITS-PLSA we vary the relative weight α of the likelihood of the content vs. the links, tuning α to its best possible value for each data set and each task. For the PubMed data set, we also normalized the content likelihood by the length of the documents.

4.3 Document Classification

4.3.1 Experimental Setting

For PMTLM, PMTLM-DC and PHITS-PLSA, we performed 500 independent runs of the EM algorithm, each with random initial values of the parameters and topic mixtures. For each run we iterated the EM algorithm up to 5000 times; we found that it typically converges in fewer iterations, with the criterion that the fractional increase of the log-likelihood for two successive iterations is less than 10^{-7}. Figure 2 shows that the log-likelihood as a function of the number of iterations are quite similar for all three data sets, even though these corpora have very different sizes. This indicates that even for large data sets, our algorithm converges within a small number of iterations, making its total running time linear in the size of the corpus.

[1] These data sets are available for download at http://www.cs.umd.edu/projects/linqs/projects/lbc/

[2] See http://www.cs.princeton.edu/~blei/lda-c/

[3] See http://www.cs.princeton.edu/~blei/lda/

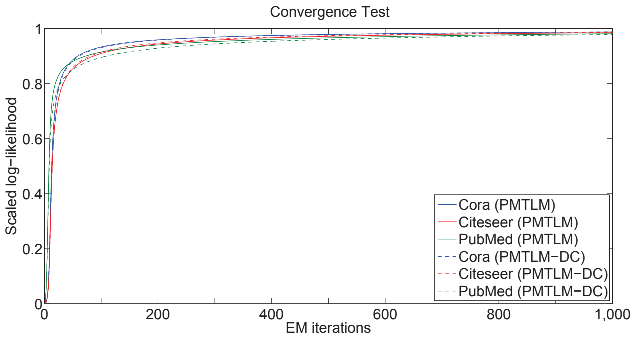


Figure 2: The average log-likelihood of the PMTLM and PMTLM-DC models as a function of the number of EM iterations, normalized so that 0 and 1 are the initial and final log-likelihood for 5000 EM iterations. Each points is the average over 100 independent runs. In both models and all three data sets, we approach 1 after just 1000 iterations, showing that the convergence time is roughly constant as a function of the size of the corpus.

		Cora	Citeseer	PubMed
Statistics	K	7	6	3
	N	2,708	3,312	19,717
	M	5,429	4,608	44,335
	W	1,433	3,703	4,209
	R	49,216	105,165	1,333,397
Time (sec)	EM (PLSA)	28	61	362
	EM (PHITS-PLSA)	40	67	445
	EM (PMTLM)	33	64	419
	EM (PMTLM-DC)	36	64	402
	EM (RTM)	992	597	2,194
	KL (PMTLM)	375	618	13,723
	KL (PMTLM-DC)	421	565	13,014

Table 1: The statistics of the three data sets, and the mean running time, for the EM algorithms in our model PMTLM, its degree-corrected variant PMTLM-DC, and PLSA, PHITS-PLSA, and RTM. Each corpus has K topics, N documents, M links, a vocabulary of size W, and a total size R. Running times for our algorithm, PLSA, and PHITS-PLSA are given for one run of 5000 EM iterations. Running times for RTM consist of up to 500 EM iterations, or until the convergence criteria are reached. Our EM algorithm is highly scalable, with a running time that grows linearly with the size of the corpus. In particular, it is much faster than the variational algorithm for RTM. Improving discrete labels with the Kernighan-Lin heuristic (KL) increases our algorithm's running time, but improves its accuracy for document classification in Cora and Citeseer.

For PMTLM and PMTLM-DC, we obtain discrete topic labels by running our EM algorithm and rounding the topic mixtures as described in Section 3.4. We also tested improving these labels with local search, using the Kernighan-Lin heuristic to change the label of one document at a time until we reach a local optimum of the likelihood. More precisely, of those 500 runs, we took the T best fixed points of the EM algorithm (i.e., with the highest likelihood) and attempted to improve them further with the KL heuristic. We used $T = 50$ for Cora and Citeseer and $T = 5$ for PubMed.

For RTM, in each E step, we initialize the variational parameters randomly, and in each M step we initialize the hyperparameters randomly. We execute 500 independent runs for each setting of the tunable parameter ρ.

4.3.2 Metrics

For each algorithm, we used several measures of the accuracy of the inferred labels as compared to the human-curated ones. The Normalized Mutual Information (NMI) between two labelings C_1 and C_2 is defined as

$$\mathrm{NMI}(C_1, C_2) = \frac{\mathrm{MI}(C_1, C_2)}{\max(\mathrm{H}(C_1), \mathrm{H}(C_2))}\,. \tag{16}$$

Here $\mathrm{MI}(C_1, C_2)$ is the mutual information between C_1 and C_2, and $\mathrm{H}(C_1)$ and $\mathrm{H}(C_2)$ are the entropies of C_1 and C_2 respectively. Thus the NMI is a measure of how much information the inferred labels give us about the true ones. We also used the Pairwise F-measure (PWF) [3] and the Variation of Information (VI) [25] (which we wish to minimize).

4.3.3 Results

The best NMI, VI, and PWF we observed for each algorithm are given in Table 2, where for LINK-LDA and C-PLDC we quote results from [32]. The metrics of NMI and PWF used in [32] are identical to ours. For algorithms with tunable parameters, including ours, PHITS-PLSA and RTM, we tuned them based on the entire data set in order to measure its best possible performance. Of course, in practice one would tune these parameters based on partial knowledge, such as the topics of a validation set of documents, and then use those parameter values to generalize to the test set.

We see that even without the additional step of local search, our algorithm does very well, outperforming all other methods we tried on Citeseer and PubMed and all but C-PLDC on Cora. (Note that we did not test LINK-LDA or C-PLDC on PubMed.) Degree correction (PMTLM-DC) improves accuracy significantly for PubMed.

Refining our labeling with the KL heuristic improved the performance of our algorithm significantly for Cora and Citeseer, giving us a higher accuracy than all the other methods we tested. For PubMed, local search did not increase accuracy in a statistically significant way. In fact, on some runs it decreased the accuracy slightly compared to the initial labeling $\hat{z}$ obtained from our EM algorithm; this is counterintuitive, but it shows that increasing the likelihood of a labeling in the model can decrease its accuracy.

In Figure 3, we show how the performance of PMTLM, PMTLM-DC, and PHITS-PLSA varies as a function of α, the relative weight of content vs. links. Recall that at $\alpha = 0$ these algorithms label documents solely on the basis of their links, while at $\alpha = 1$ they only pay attention to the content. Each point consists of the top 20 runs with that value of α.

Figure 3 also shows that the optimal α and its sensitivity to performance differs between data sets. For Cora and Citeseer, there is an intermediate value of α at which PMTLM and PMTLM-DC have the best accuracy. However, this peak is fairly broad, showing that we do not have to tune α very carefully. For PubMed, where we also normalized the content information by document length, PMTLM-DC performs best at a particular value of α.

	Cora			Citeseer			PubMed		
Algorithm	NMI	VI	PWF	NMI	VI	PWF	NMI	VI	PWF
PHITS-PLSA	0.382 (.4)	2.285 (.4)	0.447 (.3)	0.366 (.5)	2.226 (.5)	0.480 (.5)	0.233 (1.0)	1.633 (1.0)	0.486 (1.0)
LINK-LDA	0.359†	—	0.397†	0.192†	—	0.305†	—	—	—
C-PLDC	0.489†	—	0.464†	0.276†	—	0.361†	—	—	—
RTM	0.349	2.306	0.422	0.369	2.209	0.480	0.228	1.646	0.482
PMTLM	0.467 (.4)	1.957 (.4)	0.509 (.3)	0.399 (.4)	2.106 (.4)	0.509 (.3)	0.232 (.9)	1.639 (1.0)	0.486 (.9)
PMTLM (KL)	**0.514** (.4)	**1.778** (.4)	**0.525** (.4)	**0.414** (.6)	**2.057** (.6)	0.518 (.5)	0.233 (.9)	1.642 (.9)	0.488 (.9)
PMTLM-DC	0.474 (.3)	1.930 (.3)	0.498 (.3)	0.402 (.3)	2.096 (.3)	0.518 (.3)	**0.270** (.8)	**1.556** (.8)	**0.496** (.8)
PMTLM-DC (KL)	0.491 (.3)	1.865 (.3)	0.511 (.3)	0.406 (.3)	2.084 (.3)	**0.520** (.3)	0.260 (.8)	1.577 (.8)	0.492 (.8)

Table 2: The best normalized mutual information (NMI), variational of information (VI) and pairwise F-measure (PWF) achieved by each algorithm. Values marked by † are quoted from [32]; other values are based on our implementation. The best values are shown in bold; note that we seek to maximize NMI and PWF, and minimize VI. For PHITS-PLSA, PMTLM, and PMTLM-DC, the number in parentheses is the best value of the relative weight α of content vs. links. Refining the labeling returned by the EM algorithm with the Kernighan-Lin heuristic is indicated by (KL).

We compare the running time of these algorithms, including PMTLM and PMTLM-DC with and without the KL heuristic, in Table 1. For algorithms with tunable parameters, we show the running time for a single value of that parameter. For our algorithms and PHITS-PLSA, we show the running time for $\alpha = 0.5$, giving the content and the links equal weight. We see that our EM algorithm is much faster than the variational EM algorithm for RTM, and is scalable in that it grows linearly with the size of the corpus.

4.4 Link Prediction

Link prediction (e.g. [8, 22, 34]) is a natural generalization task in networks, and another way to measure the quality of our model and our EM algorithm. Based on a training set consisting of a subset of the links, our goal is to rank all pairs without an observed link according to the probability of a link between them. For our models, we rank pairs according to the expected number of links $A_{dd'}$ in the Poisson distribution, (2) and (3), which is monotonic in the probability that at least one link exists.

We can then predict links between those pairs where this probability exceeds some threshold. Since we are agnostic about this threshold and about the cost of Type I vs. Type II errors, we follow other work in this area by defining the accuracy of our model as the AUC, i.e. the probability that a random true positive link is ranked above a random true non-link. Equivalently, this is the area under the *receiver operating characteristic* curve (ROC). Our goal is to do better than the baseline AUC of 1/2, corresponding to a random ranking of the pairs.

We carried out 10-fold cross-validation, in which the links in the original graph are partitioned into 10 subsets with equal size. For each fold, we use one subset as the test links, and train the model using the links in the other 9 folds. We evaluated the AUC on the held-out links and the non-links. For Cora and Citeseer, all the non-links are used. For PubMed, we randomly chose 10% of the non-links for comparison. We trained the models with the same settings as those for document classification in Section 4.3; we executed 100 independent runs for each test. Note that unlike the document classification task, here we used the full topic mixtures to predict links, not just the discrete labels consisting of the most-likely topic for each document.

Note that PMTLM-DC assigns S_d to be zero if the degree of d is zero. This makes it impossible for d to have any test link with others if its observed degree is zero in the training data. One way to solve this is to assign a small positive value to S_d even if d's degree is zero. Our approach assigns S_d to be the smallest value among those $S_{d'}$ that are non-zero.

Figure 4(a) gives the AUC values for PMTLM and PMTLM-DC as a function of the relative weight α of content vs. links. The green horizontal line in each of those subplots represent the highest AUC value achieved by the RTM model for each data set, using the best value of ρ among those specified in Section 4.3. Note that the optimal value of the tunable parameters is task-dependent: the optimal value ρ in RTM, or α in our algorithms and PHITS-PLSA, is not necessarily the same for link prediction as it is for document classification. Interestingly, for Cora and Citeseer the optimal value of α is smaller than in Figure 3, showing that content is less important for link prediction than for document classification. Thus, according to our experiments on both document classification and link prediction, the best choice of α depends not only on the data set, but also on the task.

We also plot the *receiver operating characteristic* (ROC) curves and precision-recall curves that achieve the highest AUC values in Figure 4(b) and Figure 4(c) respectively. We see that, for all three data sets, our models outperform RTM, and that the degree-corrected model PMTLM-DC is significantly more accurate than the uncorrected one.

5. CONCLUSIONS

We have introduced a new generative model for document networks. It is a marriage between Probabilistic Latent Semantic Analysis [18] and the Ball-Karrer-Newman mixed membership block model [2]. Because of its mathematical simplicity, its parameters can be inferred with a particularly simple and scalable EM algorithm. Our experiments on document classification and link prediction show that it achieves high accuracy and efficiency for a variety of data sets, outperforming other methods. In future work, we plan to apply it to other tasks including semisupervised learning and content prediction, i.e., predicting the presence or absence of words in a document based on its links to other documents and/or a subset of its text.

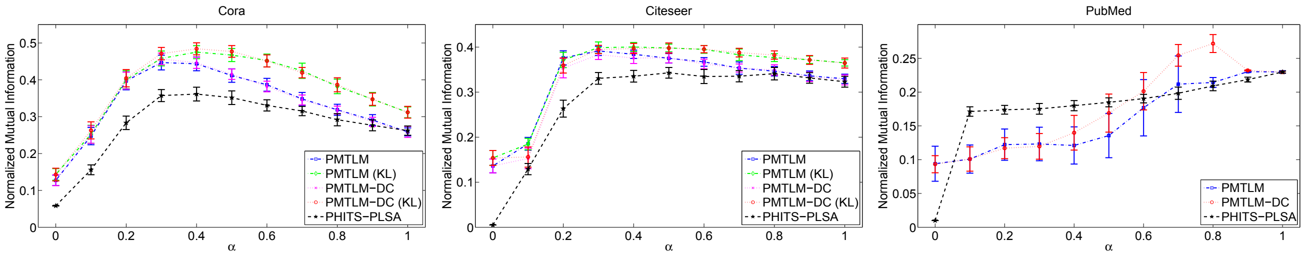
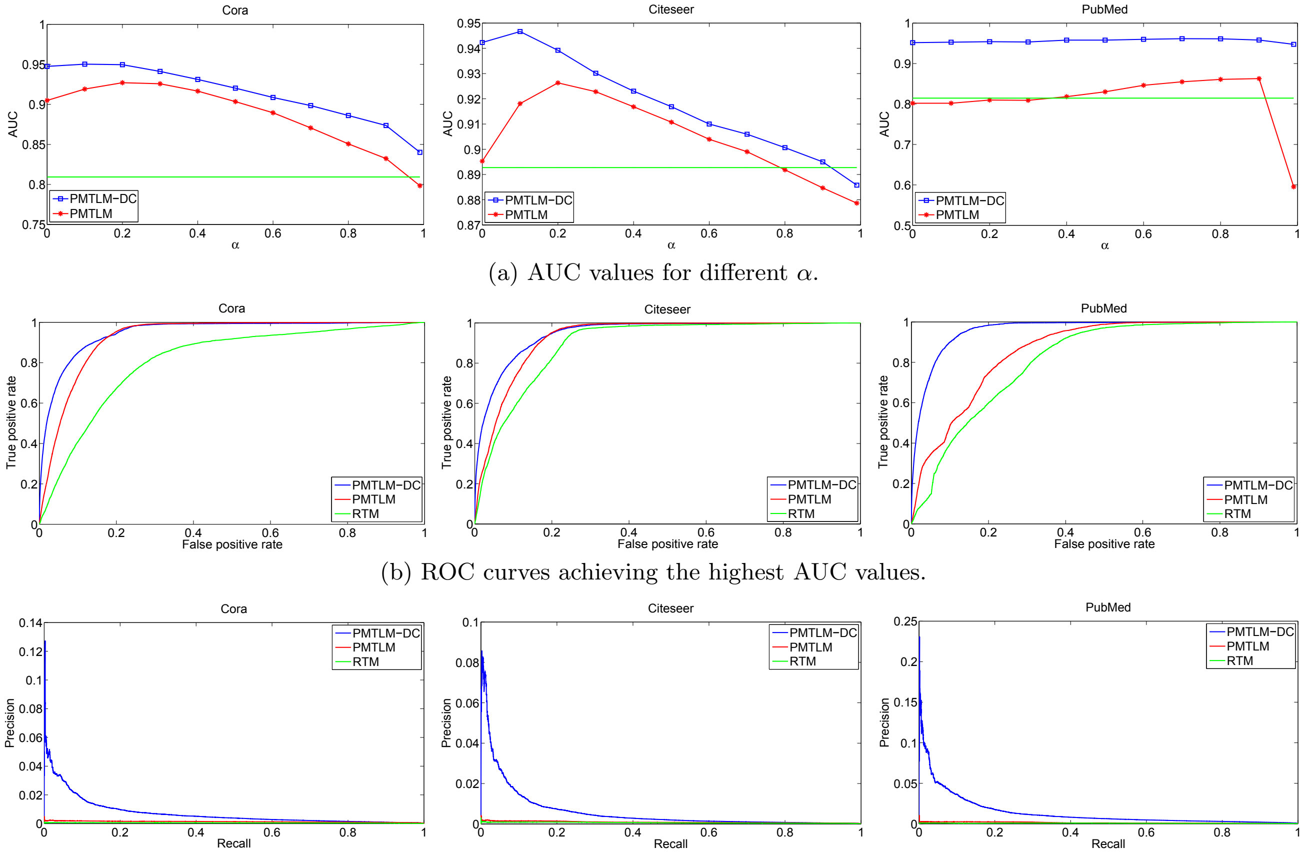


Figure 3: The accuracy of PMTLM, PMTLM-DC, and PHITS-PLSA on the document classification task, measured by the NMI, as a function of the relative weight α of the content vs. the links. At $\alpha = 0$ these algorithms label documents solely on the basis of their links, while at $\alpha = 1$ they pay attention only to the content. For Cora and Citeseer, there is a broad range of α that maximizes the accuracy. For PubMed, the degree-corrected model PMTLM-DC performs best at a particular value of α.



(a) AUC values for different α.

(b) ROC curves achieving the highest AUC values.

(c) Precision-recall curves achieving the highest AUC values.

Figure 4: Performance on the link prediction task. For all three data sets and all the α values, the PMTLM-DC model achieves higher accuracy than the PMTLM model. In contrast to Figure 3, for this task the optimal value of α is relatively small, showing that the content is less important, and the topology is more important, for link prediction than for document classification. The green line in Figure 4(a) indicates the highest AUC achieved by the RTM model, maximized over the tunable parameter ρ. Our models outperform RTM on all three data sets. In addition, the degree-corrected model (PMTLM-DC) does significantly better than the uncorrected version (PMTLM).

6. ACKNOWLEDGMENTS

We are grateful to Brian Ball, Brian Karrer, Mark Newman and David M. Blei for helpful conversations. Y.Z., X.Y., and C.M. are supported by AFOSR and DARPA under grant FA9550-12-1-0432.

7. REFERENCES

[1] E. Airoldi, D. Blei, S. Fienberg, and E. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014, 2008.

[2] B. Ball, B. Karrer, and M. E. J. Newman. Efficient and principled method for detecting communities in networks. *Phys. Rev. E*, 84(3):036103, 2011.

[3] S. Basu. *Semi-supervised Clustering: Probabilistic Models, Algorithms and Experiments.* PhD thesis, Department of Computer Sciences, University of Texas at Austin, 2005.

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

[5] J. Chang and D. M. Blei. Relational topic models for document networks. In *Proc. of Conf. on AI and Statistics (AISTATS)*, 2009.

[6] J. Chang and D. M. Blei. Hierarchical relational models for document networks. *Annals of Applied Statistics*, 4(1):124–150, 2010.

[7] A. Chen, A. A. Amini, P. J. Bickel, and E. Levina. Fitting community models to large sparse networks. *CoRR*, abs/1207.2340, 2012.

[8] A. Clauset, C. Moore, and M. E. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98–101, 2008.

[9] D. Cohn and H. Chang. Learning to probabilistically identify authoritative documents. In *International Conference on Machine Learning (ICML)*, 2000.

[10] D. Cohn and T. Hofmann. The missing link-a probabilistic model of document content and hypertext connectivity. In *Neural Information Processing Systems (NIPS)*, 2001.

[11] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Phys. Rev. E*, 84(6), 2011.

[12] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová. Inference and phase transitions in the detection of modules in sparse networks. *Phys. Rev. Lett.*, 107(6):065701, 2011.

[13] E. Erosheva, S. Fienberg, and J. Lafferty. Mixed-membership models of scientific publications. *Proc. National Academy of Sciences*, 101 Suppl:5220–7, 2004.

[14] S. E. Fienberg and S. S. Wasserman. Categorical data analysis of single sociometric relations. *Sociological methodology*, 12:156–192, 1981.

[15] L. Getoor, N. Friedman, D. Koller, and B. Taskar. Learning probabilistic models of relational structure. *Journal of Machine Learning Research*, 3:679–707, 2002.

[16] P. Gopalan, D. Mimno, S. Gerrish, M. Freedman, and D. Blei. Scalable inference of overlapping communities. In *Neural Information Processing Systems (NIPS)*, 2012.

[17] A. Gruber, M. Rosen-Zvi, and Y. Weiss. Latent topic models for hypertext. In *Proc. 24th Conf. on Uncertainty in Artificial Intelligence (UAI)*, 2008.

[18] T. Hofmann. Probabilistic latent semantic indexing. In *International ACM SIGIR conference on research and development in Information Retrieval (SIGIR)*, 1999.

[19] P. Holland, K. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983.

[20] B. Karrer and M. E. J. Newman. Stochastic blockmodels and community structure in networks. *Phys. Rev. E*, 83:016107, 2011.

[21] M. Kim and J. Leskovec. Latent multi-group membership graph model. *CoRR*, abs/1205.4546, 2012.

[22] L. Lü and T. Zhou. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, 390(6):1150–1170, 2011.

[23] Q. Lu and L. Getoor. Link-based classification. In *International Conference on Machine Learning (ICML)*, 2003.

[24] Q. Lu and L. Getoor. Link-based classification using labeled and unlabeled data. *ICML Workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, 2003.

[25] M. Meilă. Comparing clusterings by the variation of information. *Learning theory and kernel machines*, pages 173–187, 2003.

[26] C. Moore, X. Yan, Y. Zhu, J. Rouquier, and T. Lane. Active learning for node classification in assortative and disassortative networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2011.

[27] R. M. Nallapati, A. Ahmed, E. P. Xing, and W. W. Cohen. Joint latent topic models for text and citations. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2008.

[28] M. E. J. Newman and E. A. Leicht. Mixture models and exploratory analysis in networks. *Proc. National Academy of Sciences*, 104(23):9564–9, 2007.

[29] P. Sen, G. Namata, M. Bilgic, and L. Getoor. Collective classification in network data. *AI Magazine*, pages 1–24, 2008.

[30] T. Snijders and K. Nowicki. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of Classification*, 14(1):75–100, 1997.

[31] C. Sun, B. Gao, Z. Cao, and H. Li. HTM: A topic model for hypertexts. In *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, 2008.

[32] T. Yang, R. Jin, Y. Chi, and S. Zhu. A Bayesian framework for community detection integrating content and link. In *Proc. Conf. on Uncertainty in Artificial Intelligence (UAI)*, 2009.

[33] P. Yu, J. Han, and C. Faloutsos. *Link Mining: Models, Algorithms, and Applications.* Springer, 2010.

[34] Y. Zhao, E. Levina, and J. Zhu. Link prediction for partially observed networks. *arXiv preprint arXiv:1301.7047*, 2013.