Sparse Relational Topic Models for Document Networks

Aonan Zhang, Jun Zhu, and Bo Zhang

Department of Computer Science and Technology, Tsinghua University
{zan12,dcszj,dcszb}@mail.tsinghua.edu.cn

Abstract. Learning latent representations is playing a pivotal role in machine learning and many application areas. Previous work on relational topic models (RTM) has shown promise on learning latent topical representations for describing relational document networks and predicting pairwise links. However under a probabilistic formulation with normalization constraints, RTM could be ineffective in controlling the sparsity of the topical representations, and may often need to make strict mean-field assumptions for approximate inference. This paper presents sparse relational topic models (SRTM) under a non-probabilistic formulation that can effectively control the sparsity via a sparsity-inducing regularizer. Our model can also handle imbalance issues in real networks via introducing various cost parameters for positive and negative links. The deterministic optimization problem of SRTM admits efficient coordinate descent algorithms. We also present a generalization to consider all pairwise topic interactions. Our empirical results on several real network datasets demonstrate better performance on link prediction, sparser latent representations, and faster running time than the competitors under a probabilistic formulation.

1 Introduction

Given the fast growth of the Internet and data collection technologies, statistical network data analysis is playing an increasingly important role in both scientific and engineering areas, such as biology, social science, data mining, etc. A network is normally represented by a set of vertices (i.e., entities) and a set of edges (i.e., links) between these entities. Link prediction is a fundamental task in network analysis [1], and building link prediction models can provide solutions like suggesting friends for social network users or recommending products.

Many approaches have been developed for link prediction, including both parametric [2–4] and nonparametric [5, 6] Bayesian models as well as matrix factorization methods [7]. Most of these approaches focus on modeling the network structure. One work that accounts for both network structure and entity contents is the relational topic model (RTM) [8], an extension of latent Dirichlet allocation (LDA) [9] to model document networks. Because of its probabilistic formulation, RTM has some restrictions on modeling real networks, which can be highly complex and imbalanced. For example, real networks normally have very

H. Blockeel et al. (Eds.): ECML PKDD 2013, Part I, LNAI 8188, pp. 670–685, 2013.

few positive links while most are negative; but the standard maximum likelihood estimation (MLE) or Bayesian inference of RTM cannot handle this imbalance issue. Furthermore, sparsity is an important property in learning latent representations that are semantically meaningful and interpretable [10], especially in large-scale applications; but RTM cannot effectively control the sparsity of latent representations due to its probabilistic formulation with normalization constraints.

To deal with the above issues, we present an alternative formulation of relational topic models that discover nonnegative latent representations of words and documents and make predictions on unseen links. With a non-probabilistic formulation [11] and no normalization constraints, we can effectively control the sparsity of the latent representations by using a sparsity-inducing ℓ_1-norm regularizer; by using different regularization parameters on the positive link likelihood and negative link likelihood respectively, the sparse relational topic model (SRTM) can effectively deal with the imbalance issue of common real networks. Furthermore, SRTM can be generalized to capture all pairwise topic interactions in a link likelihood model and is applicable to both symmetric and asymmetric networks. Finally, SRTM admits efficient and simple coordinate descent algorithms. Empirical results on several real network datasets demonstrate better link prediction performance, sparser latent representations, as well as faster running time than the competitors under a probabilistic formulation.

The paper is structured as follows. Section 2 discusses related works. Section 3 introduces our sparse relational topic model as a cost-sensitive Maximum-a-Posteriori (MAP) estimate, as well as a coordinate descent optimization algorithm. In Section 4 we show empirical results and Section 5 concludes.

2 Related Work

Link prediction [1] has been considered as an important task in statistical network analysis. One promising branch for predicting links is to build latent variable models. Hoff et al. [3] proposed a Bayesian parametric latent variable model in which the relationship between two entities is measured by the distance between them in a latent "social space". Hoff [4] then extended the model by exploiting the low rank structure in the network link matrix. Airoldi et al. [2] built hierarchical Bayesian mixed membership block models where each entity pair has a local membership assignment and all the entity pairs are also governed by a global block matrix. To infer the dimension of the latent representations for entities from data, Miller et al. [5] developed non-parametric Bayesian models for link prediction and their max-margin variants under the regularized Bayesian framework were proposed by Zhu [6].

One drawback of the above models is that they do not account for contents of entities. This issue is even more important when we analyze document networks, where the semantic meaning of documents can be very useful for predicting links among them. Chang et al. [8] proposed probabilistic relational topic models (RTMs) built on latent Dirichlet allocation to consider both the network structure and the contents of each entity when predicting links, and their

performance exceeds several baseline methods that do not consider contents. Liu et al. [12] further considered the author communities behind the document networks in their models when predicting links among documents. Our SRTM model is a non-probabilistic variant of RTM.

SRTM is based on a non-probabilistic topic model named sparse topical coding (STC) [11], which is essentially a hierarchical non-negative matrix factorization method [10]. STC builds a two-level hierarchy by assigning codes for documents and each word in them. By relaxing normalization constraints and enforcing codes to be non-negative, STC can put an ℓ_1-norm regularizer on the word level and this makes STC a flexible model to control word code sparsity [11], which is a good property for learning topical representations especially in large-scale applications. The effectiveness of STC has been demonstrated on several domains including text [11], images and videos [13–15].

SRTM presents an extension of STC to address the challenging problem of link prediction, as we stated above. While sharing the merit of STC to learn sparse codes, SRTM can handle the imbalance issues among networks.

3 Sparse Relational Topic Models

In this section, we present the sparse relational topic model that solves a deterministic optimization problem. By relaxing the normalization constraints as in probabilistic models, SRTM can learn sparse word codes with an ℓ_1-norm regularizer and admits an efficient coordinate descent algorithm. In contrast, the probabilistic RTM often makes mean-field assumptions for approximate inference. Though SRTM can be defined from a regularized loss minimization perspective, for the ease of understanding we first introduce a probabilistic generative process and then cast SRTM as solving a MAP estimate with cost-sensitive regularization parameters to deal with imbalance issues of real networks.

3.1 A Generative Process for SRTM

Let $V = \{1, 2, \cdots, N\}$ be a vocabulary containing N terms and $\mathcal{D} = \{\mathbf{W}, \mathbf{Y}\}$ be a training dataset, where $\mathbf{W} = \{\mathbf{w}_d\}_{d=1}^D$ represents a corpus of D documents and $\mathbf{Y}$ denotes the set of pairwise links between documents. We will use $\mathcal{I}$ to denote the set of document pairs whose links are in the training set, i.e., $\mathcal{I} = \{(d, d') : y_{d,d'} \in \mathbf{Y}\}$. We adopt the conventional bag-of-words model, i.e., each document is represented as a set $\mathbf{w_d} = \{w_{dn}, n \in I_d\}$, where w_{dn} is the *word count* for the nth term in the dictionary and I_d is the set of terms in document d. Let $y_{d,d'}$ denote the label of the link between document d and d'. Though SRTM can be easily extended to do multi-type link prediction, for clarity we consider binary links, that is $y_{d,d'} = 1$ if there is a link between document[1] d and d', and $y_{d,d'} = -1$ otherwise.

[1] For asymmetric networks, $y_{d,d'}$ denotes the link from document d to document d'.

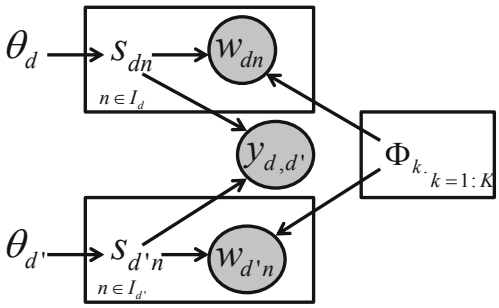

Fig. 1. Graphical Model for SRTM considering only one document pair as an illustration

As a relational topic model, SRTM models words $\mathbf{W}$ and links $\mathbf{Y}$ with two closely connected components. The first component is a hierarchical sparse topical coding (STC) [11] to describe words by using a topical dictionary $\mathbf{\Phi} \in \mathbb{R}^{K \times N}$ with K topical bases, that is, each row $\mathbf{\Phi}_{k\cdot}$ is a normalized distributional vector over the given vocabulary. We use $\mathbf{\Phi}_{\cdot n}$ to denote the nth column of $\mathbf{\Phi}$. Each document d has a topical representation $\boldsymbol{\theta}_d \in \mathbb{R}^K$ (i.e., *document code*) and each words in the document has an individual *word code* $\mathbf{s}_{dn} \in \mathbb{R}^K$ $(n \in I_d)$. Note that here we do not put normalization constraints on document codes or word codes. This relaxation enables us to build a more flexible topic model. In fact, we can achieve sparse word codes by imposing non-negative constraints and a sparsity-inducing regularizer [10, 11]. SRTM also assumes that word codes in one document are independent given the document code and the word count w_{dn} follows a distribution whose mean parameter is $\mathbf{s}_{dn}^\top \mathbf{\Phi}_{\cdot n}$ [11]. The second component of SRTM defines a likelihood model of the links between documents. Formally, the generative procedure of SRTM on document words and links can be described as:

1. for each document d
   (1) draw a document code $\boldsymbol{\theta}_d$ from $p(\boldsymbol{\theta}_d)$.
   (2) for each observed word $n \in I_d$
      (a) draw the word code $\mathbf{s}_{dn}$ from $p(\mathbf{s}_{dn}|\boldsymbol{\theta}_d)$
      (b) draw the observed word count w_{dn} from $p(w_{dn}|\mathbf{s}_{dn}^\top \mathbf{\Phi}_{\cdot n})$.
2. for each document pair (d, d'), draw a link from $p(y_{d,d'}|\bar{\mathbf{s}}_d, \bar{\mathbf{s}}_{d'})$.

where $\bar{\mathbf{s}}_d = \frac{1}{|I_d|} \sum_{n \in I_d} \mathbf{s}_{dn}$ is the average word code of document d, a representation of document d in the latent topic space. For the clarity of presentation, we show a graphical model of SRTM considering only one document pair in Fig. 1, and it can be easily extended to model a large network of documents. To fully specify the model, we need to define the word likelihood model $p(w_{dn}|\mathbf{s}_{dn}, \mathbf{\Phi})$ and the link likelihood model $p(y_{d,d'}|\bar{\mathbf{s}}_d, \bar{\mathbf{s}}_{d'})$. For word counts, since w_{dn} is a positive integer, we choose the commonly used Poisson distribution and set $\mathbf{s}_{dn}^\top \mathbf{\Phi}_{\cdot n}$ as the mean parameter:

$$p(w_{dn}|\mathbf{s}_{dn}, \mathbf{\Phi}) = Poisson(w_{dn}, \mathbf{s}_{dn}^\top \mathbf{\Phi}_{\cdot n}), \tag{1}$$

where $Poisson(x, \nu) = \frac{\nu^x e^{-\nu}}{x!}$. One benefit for setting the inner product $\mathbf{s}_{dn}^\top \mathbf{\Phi}_{\cdot n}$ as mean parameter is that we can easily constrain the word code space by enforcing $\mathbf{s}_{dn}$ to be non-negative and by using a sparsity-inducing ℓ_1-norm regularizer [10]. For the link likelihood, both the sigmoid function and exponential link function were used in [8]. But, the exponential function is itself unnormalized and some special treatment is needed to normalize it. Therefore, we choose the more common sigmoid function to model the probability of a link:

$$p(y_{d,d'}|\bar{\mathbf{s}}_d, \bar{\mathbf{s}}_{d'}) = \sigma\Big(y_{d,d'}(\boldsymbol{\eta}^\top(\bar{\mathbf{s}}_d \circ \bar{\mathbf{s}}_{d'}) + \nu)\Big), \tag{2}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$; $\boldsymbol{\eta} = (\eta_1, \eta_2, \cdots, \eta_K)^\top$ are the parameters describing how likely there is a link between two documents when they share a specific topic; and ν denotes the offset for the link probability. The symbol $\circ$ denotes the element-wise product.

3.2 Cost-Sensitive MAP Estimate

Let $\mathbf{\Theta} = \{\boldsymbol{\theta}_d\}$ and $\mathbf{S} = \{\mathbf{s}_d\}$ denote the latent representations of documents and words respectively. Then the joint distribution of SRTM can be written as:

$$p(\mathbf{W}, \mathbf{Y}, \mathbf{\Theta}, \mathbf{S}|\mathbf{\Phi}) = \prod_d \left(p(\boldsymbol{\theta}_d) \prod_{n \in I_d} p(\mathbf{s}_{dn}|\boldsymbol{\theta}_d) p(w_{dn}|\mathbf{s}_{dn}, \mathbf{\Phi}) \right) \prod_{(d,d') \in \mathcal{I}} p(y_{d,d'}|\bar{\mathbf{s}}_d, \bar{\mathbf{s}}_{d'}) \tag{3}$$

We naturally impose a normal prior on $\boldsymbol{\theta}_d$ so that $p(\boldsymbol{\theta}_d) \propto \exp(-\lambda\|\boldsymbol{\theta}_d\|_2^2)$. For the word code $\mathbf{s}_{dn}$ we use a Laplace prior to achieve sparsity [16]. Furthermore, we restrict the word codes not too far away from the document code by a normal regularizer. This results in a composite prior $p(\mathbf{s}_{dn}|\boldsymbol{\theta}_d) \propto \exp(-\gamma\|\boldsymbol{\theta}_d - \mathbf{s}_{dn}\|_2^2 - \rho\|\mathbf{s}_{dn}\|_1)$, which is super-Gaussian [17] and the ℓ_1-term drives our estimates to be sparse. The hyper-parameters (λ, γ, ρ) can be predefined or selected using cross-validation. We will provide sensitivity analysis to these parameters in experiments.

With the above joint distribution, a standard MAP estimate with dictionary learning can be formulated as solving the problem:

$$\begin{aligned} \min_{\mathbf{\Theta}, \mathbf{S}, \mathbf{\Phi}} \quad & \ell(\mathbf{S}, \mathbf{\Phi}; \mathbf{W}) + \ell(\mathbf{S}, \boldsymbol{\eta}; \mathbf{Y}) + \Omega(\mathbf{\Theta}, \mathbf{S}) \\ \text{s.t.:} \quad & \boldsymbol{\theta}_d \geq 0, \forall d; \mathbf{s}_{dn} \geq 0, \forall d, n \in I_d; \mathbf{\Phi}_{k\cdot} \in \mathcal{P}, \forall k, \end{aligned} \tag{4}$$

where $\ell(\mathbf{S}, \mathbf{\Phi}; \mathbf{W}) = \sum_{d,n \in I_d} \ell(\mathbf{s}_{dn}, \mathbf{\Phi}) = -\sum_{d,n \in I_d} \log Poisson(w_{dn}, \mathbf{s}_{dn}^\top \mathbf{\Phi}_{\cdot n})$ is the negative log-likelihood of word counts; $\ell(\mathbf{S}, \boldsymbol{\eta}; \mathbf{Y}) = \sum_{(d,d') \in \mathcal{I}} \ell(\mathbf{s}_d, \mathbf{s}_{d'}; y_{d,d'}) = -\sum_{(d,d') \in \mathcal{I}} \log p(y_{d,d'}|\bar{\mathbf{s}}_d, \bar{\mathbf{s}}_{d'})$ is the negative log-likelihood of links; $\Omega(\mathbf{\Theta}, \mathbf{S}) = \lambda \sum_d \|\boldsymbol{\theta}_d\|_2^2 + \sum_{d,n \in I_d} (\gamma\|\mathbf{s}_{dn} - \boldsymbol{\theta}_d\|_2^2 + \rho\|\mathbf{s}_{dn}\|_1)$ is the regularization term; $\mathcal{P}$ is the $(N-1)$-dimensional simplex. The negative log-likelihood is usually called a log-loss. We have imposed non-negative constraints on the latent representations in order to obtain good interpretability, as a non-negative code can be interpreted

as the importance of a topic. Moreover, non-negative constraints are good for our objective of a sparse estimate.

It is worth noting that there could be two imbalance issues with the standard MAP estimate. Firstly, for each pair of documents there is only one link variable while there could be hundreds of words. This difference would lead to an imbalanced combination of word likelihood and link likelihood in problem (4). Secondly, in common real networks only a few links are positive while most are negative, e.g., the widely used Cora citation network [8] has about 0.1% positive links. This difference would lead to an imbalanced combination of positive link likelihood and negative link likelihood. To address these imbalance issues, we can easily extend the regularized log-loss minimization problem to a cost-sensitive MAP estimate by introducing different regularization parameters for the positive and negative links respectively. Specifically, we replace the standard log-loss of links with the following cost-sensitive log-loss:

$$\ell(\mathbf{S}, \boldsymbol{\eta}; \mathbf{Y}) = C_+ \sum_{(d,d')\in\mathcal{I}_+} \ell(\mathbf{s}_d, \mathbf{s}_{d'}; y_{d,d'}) + C_- \sum_{(d,d')\in\mathcal{I}_-} \ell(\mathbf{s}_d, \mathbf{s}_{d'}; y_{d,d'}), \quad (5)$$

where $\mathcal{I}_+ = \{(d, d') \in \mathcal{I} : \ y_{d,d'} = 1\}$ and $\mathcal{I}_- = \mathcal{I} \backslash \mathcal{I}_+$. Then, by setting C_+ and C_- at a value larger than 1, we can improve the influence of links and overcome the imbalance issue between words and links; and by setting C_+ at a value larger than C_-, we can better balance the influence of positive links and negative links. We will provide more insights in the experiment section.

If we look back at the generative formulation, which is easy to understand, an intuitive understanding of the regularization parameters C_+ and C_- is that they are *pseudo-counts* of the links, and the likelihood of the links are correspondingly:

$$p(y_{d,d'} = 1|\bar{\mathbf{s}}_d, \bar{\mathbf{s}}_{d'}) = \sigma(\boldsymbol{\eta}^\top(\bar{\mathbf{s}}_d \circ \bar{\mathbf{s}}_{d'}) + \nu)^{C_+}$$
$$p(y_{d,d'} = -1|\bar{\mathbf{s}}_d, \bar{\mathbf{s}}_{d'}) = \sigma(-\boldsymbol{\eta}^\top(\bar{\mathbf{s}}_d \circ \bar{\mathbf{s}}_{d'}) - \nu)^{C_-}.$$

Note that these likelihood functions are unnormalized if the pseudo-counts are not 1. But the un-normalization does not affect our estimates in the cost-sensitive log-loss minimization framework.

3.3 Optimization Algorithms

We first present our learning algorithm for solving problem (4). Since the optimization problem is bi-convex, i.e. convex over $\boldsymbol{\Theta}$ and $\mathbf{S}$ given the dictionary $\boldsymbol{\Phi}$ and the networks parameters $\boldsymbol{\eta}$ and ν; and convex over $\boldsymbol{\Phi}$, $\boldsymbol{\eta}$, and ν given the document codes $\boldsymbol{\Theta}$ and the word codes $\mathbf{S}$, we use a coordinate descent algorithm to iteratively optimize the objective function. As outlined in Algorithm 1, the algorithm iteratively solves three subproblems:

1. *Hierarchical Sparse Coding*: learns document codes and sparse word codes for the documents;
2. *Dictionary Learning*: learns the topical dictionary with document codes and word codes given;

Algorithm 1. Sparse Relational Topic Models

```
Initialize Φ, Θ, S, η, ν
read corpus D
while not converge do
    (Θ, S) = HierarchicalSparseCoding(Φ, η, ν);
    Φ = DictionaryLearning(S);
    (η, ν) = LinkModelLearning(S);
end while
```

3. *Link Model Learning*: learns the link likelihood model with the codes and topical dictionary given.

Below, we discuss each step in detail. For notation simplicity, we will set $C_+ = C_- = C$.

Hierarchical Sparse Coding: This step involves solving for the word codes and document codes. Since the subproblem is convex, we can apply a generic algorithm to solve it. Here, we use the similar coordinate descent method as in [11]. For *document codes*, since the documents are independent, we can solve for each $\boldsymbol{\theta}_d$ separately and this results in a convex subproblem:

$$\min_{\boldsymbol{\theta}_d} \lambda \|\boldsymbol{\theta}_d\|_2^2 + \gamma \sum_{n \in I_d} \|\mathbf{s}_{dn} - \boldsymbol{\theta}_d\|_2^2, \text{ s.t.: } \boldsymbol{\theta}_d \geq 0. \tag{6}$$

It can be shown that the optimum solution is $\boldsymbol{\theta}_d = \frac{\gamma \sum_{n \in I_d} \mathbf{s}_{dn}}{\lambda + \gamma |I_d|}$, that is, the document code is the average (with some re-scaling) of word codes.

For *word codes*, again we can treat each document separately. Formally, the optimization problem for word codes of document d is:

$$\begin{aligned} &\min_{\mathbf{s}_d} \sum_{n \in I_d} \ell(\mathbf{s}_{dn}, \beta) + \sum_{n \in I_d} (\gamma \|\mathbf{s}_{dn} - \boldsymbol{\theta}_d\|_2^2 + \rho \|\mathbf{s}_{dn}\|_1) + C \sum_{d' \in \mathcal{N}_d} \ell(\mathbf{s}_d, \mathbf{s}_{d'}; y_{d,d'}) \\ &\text{s.t.: } \mathbf{s}_{dn} \geq 0, \forall n \in I_d, \end{aligned} \tag{7}$$

where $\mathcal{N}_d = \{d' : (d, d') \in \mathcal{I}\}$ is the neighborhood of document d in the training network. For the sigmoid link function, the log-loss of links is

$$\ell(\mathbf{s}_d, \mathbf{s}_{d'}; y_{d,d'}) = \log\Big(1 + \exp(-y_{d,d'}(\boldsymbol{\eta}^\top (\bar{\mathbf{s}}_d \circ \bar{\mathbf{s}}_{d'}) + \nu))\Big). \tag{8}$$

Since the objective function w.r.t. a single word code is convex given other word codes, we can iteratively optimize each word code $\mathbf{s}_{dn}$ by solving:

$$\begin{aligned} &\min_{\mathbf{s}_{dn}} \ell(\mathbf{s}_{dn}, \boldsymbol{\Phi}) + \gamma \|\mathbf{s}_{dn} - \boldsymbol{\theta}_d\|_2^2 + \rho \|\mathbf{s}_{dn}\|_1 + C \sum_{d' \in \mathcal{N}_d} \ell(\mathbf{s}_d, \mathbf{s}_{d'}; y_{d,d'}) \\ &\text{s.t.: } \mathbf{s}_{dn} \geq 0. \end{aligned} \tag{9}$$

This subproblem does not have a closed-form solution because of the nonlinearity of the sigmoid likelihood. Therefore, we resort to numerical methods using projected gradient descent [18] to take care of the constraints. Precisely, we take a

gradient descent step with a stepsize selected with line search, and then perform projection onto the convex feasible domain. Formally, the projected gradient descent is to update:

$$\mathbf{s}_{dn}^{new} = \mathit{\Pi}_P(\mathbf{s}_{dn}^{old} - t\nabla_{\mathbf{s}_{dn}}\mathcal{L})$$

where t is the step size; $\mathit{\Pi}_P$ is a projection operator; and $\mathit{\Pi}_P(x) = \arg\min_{x' \in P} d(x, x')$. Here P is the positive half space of $\mathbb{R}^K$ and $d(\cdot, \cdot)$ stands for the Euclidian distance. Let $\mathcal{L}$ be the objective function of the subproblem (9). We can verify that $s_{dnk}^{new} = 0$ if $s_{dnk}^{old} - t\nabla_{s_{dnk}}\mathcal{L} < 0$ and $s_{dnk}^{new} = s_{dnk}^{old} - t\nabla_{s_{dnk}}\mathcal{L}$ otherwise. To simplify notation, we first calculate the derivative of the sigmoid link function in Eq. (8) w.r.t. to $\mathbf{s}_{dn}$

$$\nabla_{\mathbf{s}_{dn}}\ell(\mathbf{s}_d, \mathbf{s}_{d'}; y_{d,d'}) = \frac{\partial \ell}{\partial z_{d,d'}} \cdot \frac{\partial z_{d,d'}}{\partial \mathbf{s}_{dn}} = \frac{-y_{d,d'}\exp(z_{d,d'})}{1+\exp(z_{d,d'})} \cdot \frac{\eta_k \bar{\mathbf{s}}_{d'}}{|I_d|}, \tag{10}$$

where $z_{d,d'} = -y_{d,d'}(\boldsymbol{\eta}^\top(\bar{\mathbf{s}}_d \circ \bar{\mathbf{s}}_{d'}) + \nu)$. Then, the gradient w.r.t. $\mathbf{s}_{dn}$ is

$$\nabla_{\mathbf{s}_{dn}}\mathcal{L} = (1 - \frac{w_{dn}}{\mathbf{s}_{dn}^\top \boldsymbol{\Phi}_{.n}})\boldsymbol{\Phi}_{.n} + 2\gamma(\mathbf{s}_{dn} - \boldsymbol{\theta}_d) + \rho + C\sum_{d' \in \mathcal{N}_d} \nabla_{\mathbf{s}_{dn}}\ell(\mathbf{s}_d, \mathbf{s}_{d'}; y_{d,d'}). \tag{11}$$

Dictionary Learning: This step involves solving for the topical dictionary $\boldsymbol{\Phi}$. Since $\boldsymbol{\Phi}$ is constrained on a probabilistic simplex, we can use projected gradient descent to update $\boldsymbol{\Phi}$ and then project each row onto an ℓ_1-simplex [11]. Efficient linear time projection methods are available to make this step fast [19].

Link Likelihood Learning: This step involves solving for the parameters $\boldsymbol{\eta}$ and ν of the link likelihood model. In this step we only need to account for the link part $\sum_{(d,d') \in \mathcal{I}} \ell(\mathbf{s}_d, \mathbf{s}_{d'}, y_{d,d'})$. The objective for each link is convex so the summation is also convex for $\boldsymbol{\eta}$ and ν. Simply taking gradient we get

$$\nabla_{\eta_k}\mathcal{L} = C\sum_{(d,d') \in \mathcal{I}} \frac{-y_{d,d'}\bar{s}_{dk}\bar{s}_{d'k}\exp(z_{d,d'})}{1+\exp(z_{d,d'})}$$

$$\nabla_{\nu}\mathcal{L} = C\sum_{(d,d') \in \mathcal{I}} \frac{-y_{d,d'}\exp(z_{d,d'})}{1+\exp(z_{d,d'})}$$

and we can use gradient descent with line search to solve the problem.

3.4 A Generalized Sparse Relational Topic Model

It is worth noticing that in SRTM we define the strength of a link between two documents by $\boldsymbol{\eta}^\top(\bar{\mathbf{s}}_d \circ \bar{\mathbf{s}}_{d'}) + \nu = \bar{\mathbf{s}}_d \top \mathrm{diag}(\boldsymbol{\eta})\bar{\mathbf{s}}_{d'} + \nu$, where $\mathrm{diag}(\boldsymbol{\eta})$ is a diagonal matrix with the diagonal elements being those of $\boldsymbol{\eta}$. Therefore, SRTM can only capture the *same-topic-interactions* (i.e., only when two documents have the same topic, there is a nonzero contribution to the link likelihood); and it could be unsuitable for modeling asymmetric networks because of the symmetric nature of diagonal matrices. To relax these constraints and capture *all-pairwise-topic-interactions*, one straightforward extension is to use a full weight matrix $H^{K \times K}$ and define the link likelihood model as:

$$p(y_{d,d'}|\bar{\mathbf{s}}_d, \bar{\mathbf{s}}_{d'}) = \sigma(y_{d,d'}(\bar{\mathbf{s}}_d^\top H \bar{\mathbf{s}}_{d'}^\top + \nu)). \tag{12}$$

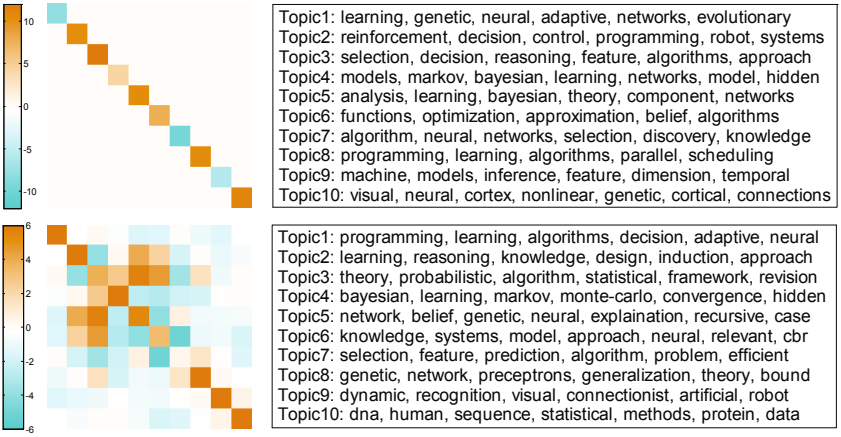

Fig. 2. Weight matrix and according representative words for each topic learned by SRTM (first row) and gSRTM (second row) on the Cora citation network data

where H_{ij} represents the strength of two documents being connected when they have topic i and topic j respectively. We denote this generalized SRTM by gSRTM. Formally, using the sigmoid likelihood function we have a similar optimization problem, and a similar coordinate descent algorithm can be applied with few changes for learning word codes and link likelihood models when taking the gradient descent steps.

Before presenting all the details of our experiments, we first illustrate the latent semantic structures learned by the sparse relational topic models and compare the diagonal SRTM and the generalized SRTM with a full weight matrix. Specifically, Fig. 2 shows the weight matrices learned by SRTM and gSRTM on the Cora citation network data (details are in the next section), as well as the top words of each of the 10 topics, respectively. For the diagonal SRTM, since the latent features $\bar{\mathbf{s}}_d$ in the link likelihood are nonnegative, the learned weight matrix must have some negative diagonal entries although most diagonal entries are positive in order to fit the training data with binary links. The negative diagonal entries somehow conflict our intuition that papers with the same topic should be more likely to have a citation link. In contrast, the full weight matrix learned by gSRTM has only positive diagonal entries, which are consistent with our intuition; and many off-diagonal entries are negative, again consistent with our intuition that papers with different topics are less likely to have a citation relation. We also note that some topics are generic, and papers with these topics are likely to get cited by or cite the papers with other closely related topics. For example, Topic3 in gSRTM is a generic topic about *theory*, *probabilistic*, *algorithm* and *statistical*; and the papers with Topic3 are likely to have a citation relationship with the papers with the related topics, such as Topic4 (*Bayesian*, *learning*, *Markov*, etc.), Topic5 (*network*, *belief*, *genetic*, etc.), and Topic6 (*knowledge*, *systems*, *model*, etc.).

Table 1. Statistics of the datasets used in our experiments

Dataset	# Entities	# Terms (N)	# Links	Link sparsity ratio
Cora [20]	2,708	1,433	5,429	0.07%
WebKB [21]	877	1,608	1,703	0.2%
CiteSeer	3,312	3,703	4,714	0.04%

4 Experiments

In this section, we present more experimental results and compare with several models on link prediction tasks. We further present a sensitivity analysis over some built-in hyper-parameters to verify that SRTM can handle the imbalance issues in real networks while effectively learning sparse word codes.

4.1 Datasets and Models

Our experiments are conducted on three publicly available datasets. All the datasets contain very sparse positive links, as detailed below:

- The *Cora* dataset [20] consists of 2,708 research papers with a vocabulary of 1,433 terms in total. Among the papers there are 5,429 positive links, each representing a citation from one paper to the other. So on average each paper has about 2 citations and the ratio of positive links is roughly 0.07%;

- The *WebKB* dataset [21] consists of 877 webpages collected from computer science departments of four universities, with 1,608 hyper-links among pages. In total, there are 1,703 terms in the dictionary. Again, this network is sparse and about 0.2% of the pairs have links;

- The *CiteSeer* dataset is another sparse document network consisting of 3,312 papers and 3,703 citations among those papers (i.e., the link sparsity ratio is about 0.04%). Its dictionary consists of 4,712 individual words.

Since RTM has been shown to outperform several baseline models on link prediction [8], our empirical studies are concentrated on analyzing the effectiveness of sparse learning in relational topic models. We use RTM as our competitive baseline method, and compare all the methods on the above three real network datasets. In summary, the methods we compare are the followings:

- **RTM** [8]: the probabilistic relational topic model built on LDA using variational methods with mean-field assumptions to approximately infer the posterior distribution. We consider the case where the logistic link function is used to model links with a diagonal weight matrix;

- **STC+Regression**: a two-step model in which we first train an unsupervised sparse topic coding (STC) [11] to discover the latent representations of

all documents and then learn a logistic regression model on training links to predict the links of testing document pairs. Note that the link information does not affect the latent representations in this method;

– **SRTM**: the proposed sparse relational topic model that uses a diagonal weight matrix in the logistic link likelihood function;

– **gSRTM**: the generalized SRTM with a full symmetric weight matrix in the logistic link likelihood model.

4.2 Results on Link Prediction

We follow the same approach as in [8] to predict links for unseen documents. Namely, for each testing document, we predict its links to the training documents. For SRTM models (i.e., SRTM and gSRTM), this can be done by first inferring the latent representation of the testing document through solving a hierarchical sparse coding step, and then applying the logistic link likelihood function to compute the probability of existing a link. Given a link's probability, we can make binary decision, that is, if the probability is larger than 0.5, there is a link exists; otherwise, no link exists. Here, we use *link rank*[2] as the performance measure, the same as in [8]. We also compare the training time to analyze the efficiency of various methods. Since all the methods are very efficient in testing, we omit the comparison on testing time.

To partly address the serious imbalance issues of the real networks and improve time efficiency, we randomly sample 0.2% of the negative links[3] and form the training data together with all the positive links to learn the sparse topic models, including SRTM, gSRTM and the de-coupled approach of STC+Regression. For the probabilistic RTM, since there is no effective mechanism on balancing positive and negative links, we found that using the same down-sampling strategy would produce worse results on both link prediction and time efficiency than the "regularization" strategy suggested in [8]. Thus, we choose to use only positive links and put a regularizer over $\boldsymbol{\eta}$ and ν to make sure that they will not diverge.

Fig. 3 shows the results on link prediction and training time. We tune hyper-parameters for all the models to their best settings for link prediction. For RTM, we tune the Dirichlet hyper-parameters α and for the SRTM models we fix $\lambda = \gamma$ and tune the ratio ρ/γ. Those hyper-parameters will affect link prediction results and the sparsity of word codes, and we will provide a sensitivity analysis on them in Section 4.3. But, in general, SRTM models have a wide range of these parameters to get good link prediction and sparsity of word codes. Overall, we

[2] For a document, its link rank is defined an average over the ranks of positive links in the list of all testing pairs. Then, the overall link rank is an average of the link rank over all testing documents.

[3] Other sampling ratios (e.g., 1%, 0.5%, 0.1%, etc.) do not affect the link prediction results of the SRTM models much, due to the effective balancing strategy by tuning regularization parameters.

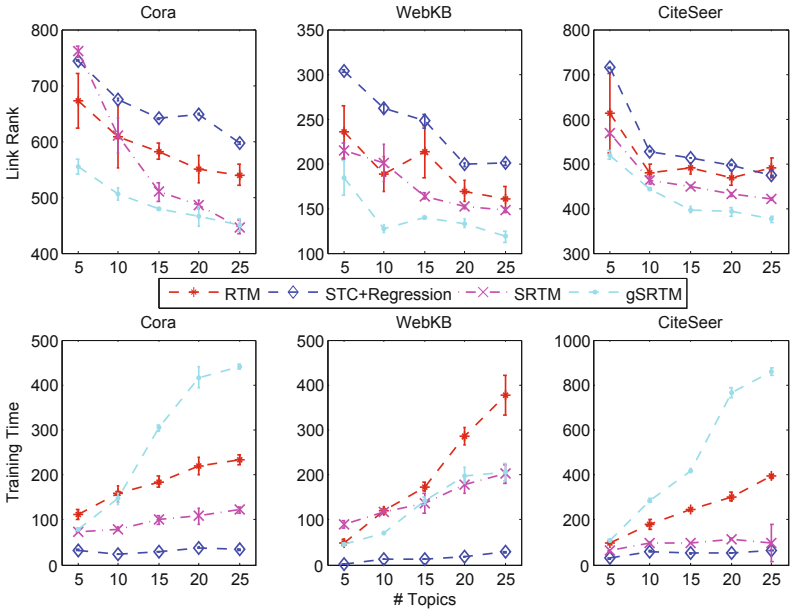

Fig. 3. First row: Link rank on three datasets using different models when changing the number of topics. Second row: Training time (in seconds) on three datasets using different models when changing the number of topics.

can see that the sparse relational topic models obtain significantly better results on all datasets. A closer examination can be done by comparing the following model pairs:

– *RTM vs. SRTM*: On all the datasets, SRTM makes more accurate link prediction (e.g., SRTM improves the average link rank by about 100 on the Cora dataset) and uses less (about 2 times when the number of topics is relatively large) time than RTM. These improvements are attributed to several factors. First, SRTM accounts for the imbalance issues in the network, which can affect the link prediction performance, while RTM cannot handle that within its Bayesian framework. Second, RTM makes mean-field assumptions, which can be too strict [22], while SRTM avoids making this assumption by solving a deterministic optimization problem. Finally, SRTM uses coordinate descent methods to optimize the objective function, where each step breaks down to very quick projected gradient methods. All these factors make SRTM perform better in link prediction while still faster than RTM, even though RTM does not use negative links;

– *STC+Regression vs. SRTM*: Since SRTM takes link information into account during the hierarchical sparse coding step, its latent representations could be more discriminative for link prediction and thus SRTM obtains a huge gain in link prediction as shown in Fig. 3. With moderate values of C_+ and C_-, SRTM accounts for both links and words to produce a much

powerful network model for link prediction. With no surprise, we require more time as a cost for considering links in SRTM. Notice that SRTM collapses to STC+Regression when $C_+ = C_- = 0$ and the behavior of SRTM approximates the matrix factorization approach for link prediction when C is significantly larger than other factors in Eq. (4). We will further analyze this phenomenon in Section 4.3;

– *SRTM vs. gSRTM*: Fig. 3 shows that the generalized gSRTM can make better prediction on all the datasets than SRTM, while spending more training time on the Cora and CiteSeer datasets. The reason is that by using a $K \times K$ full weight matrix and capturing all pairwise topic interactions in link likelihood model, gSRTM can capture valuable topic relationships and thus fit the network data better as we have illustrated in Fig. 2. Of course, using a full weight matrix with more (i.e., K^2) non-zero elements would increase the computational burden, obviously in the steps of link likelihood learning and less obviously in the step of learning word codes when computing gradients and objective functions. On the WebKB dataset the training time of both SRTM and gSRTM seems comparable. The reason is that gSRTM converges in fewer steps on this dataset and thus the total time cost is low.

4.3 Sensitivity Analysis

Word Code Sparsity. The strength of SRTM partly lies on its flexibility to learn sparse word codes by adjusting the hyper-parameters (λ, γ, ρ). Following [11] we fix $\lambda = \gamma$ and only tune the ratio ρ/γ. By checking problem (4) we can clearly see that when setting ρ/γ to a relative large value, SRTM is encouraged to learn sparse word codes. But this can cause a high divergence between word codes and the corresponding document code. From our experiments we verify that balancing the two factors can let the model generalize well to unseen data while effectively learning sparse word codes. For the RTM model, the Dirichlet hyper-parameters α control the sparsity level[4]. As it will be shown in the experiments, RTM cannot learn sparse word codes while maintaining good link prediction performance by tuning α.

In Fig. 4(a) and Fig. 4(b) we compare the sparsity ratio of word codes[5] between RTM and SRTM with different numbers of topics when tuning their hyper-parameters. For RTM, we tune the Dirichlet parameter α and for SRTM we fix γ to a constant and tune[6] ρ. This results in a change of ρ/γ. Fig. 4(a) shows a sharp drop of sparsity ratio when α grows to a certain level in RTM[7]. This is due to the property of the Dirichlet prior, where a little shift can cause the

[4] We use the common symmetric Dirichlet prior for the topic mixing proportions in RTMs.

[5] The sparsity ratio is defined as the average ratio of zero elements in word codes.

[6] Changing both γ and ρ will lead to even better link prediction results.

[7] In theory, RTM does not have sparse word codes if $\alpha > 0$. Here, we treat a small value ϵ (e.g., $\epsilon < 0.001$) as zero.

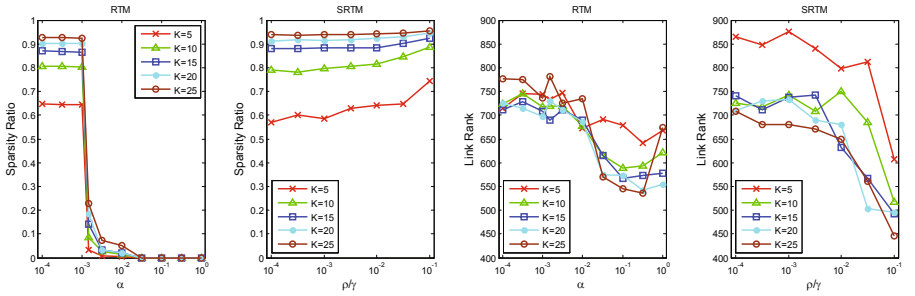

Fig. 4. Sparsity ratio (a) and link rank (c) for RTM with different number of topics when tuning hyper-parameter α on the Cora dataset; Sparsity ratio (b) and link rank (d) for SRTM with different number of topics when tuning the ratio of hyper-parameters ρ/γ on the Cora dataset.

"sharpness" of the prior changes significantly. For SRTM, Fig. 4(b) demonstrates that the sparsity ratio stays at a relative high level. When the number of topics is relatively small, changing ρ can gradually affect the sparsity ratio. There is a trend that SRTM does not learn a dense word code, which is probably due to a clear meaning of words in the dataset that each word only has a few topical meanings.

We also analyze how the hyper-parameters affect link prediction accuracy. Fig. 4(c) shows that the best link prediction results of RTM can be reached when α is around 0.1. At this point, the sparsity ratio is zero. So on the Cora dataset, RTM tends to perform better when learning dense codes. This is not a coincidence because a small α can produce a very "sharp" Dirichlet prior, which can dramatically bias the model and result in an inefficient control of sparsity ratio. In contrast, from Fig. 4(d) we can see that for SRTM there is a gradual change in link rank when ρ grows. Finally, the model reaches its best link rank result at a high sparsity ratio when ρ/γ is around 0.1. The reason is that SRTM relaxes the probability constraints of codes and thus effectively learn sparse codes by introducing ℓ_1-norm constraints at the word code level. SRTM achieves a built-in sparsity control mechanism by constructing a two-level hierarchical topic model.

The Hyper-Parameter C. As we have discussed, a relational topic model might have two imbalance issues, i.e., the imbalance between modeling words and links, and the imbalance between positive links and negative links. To address both issues, SRTM introduces the hyper-parameter C_+ for positive links and C_- for negative links. For the first issue, we can fix a reasonable value of C_+ and C_- to balance words and links. For the second issue, since negative links usually dominate positive links, we can either tune the C_+/C_- ratio or sub-sample the negative links. In our experiments, we use both strategies and find that sub-sampling a few negative links while tuning C_+/C_- can make very good prediction results.

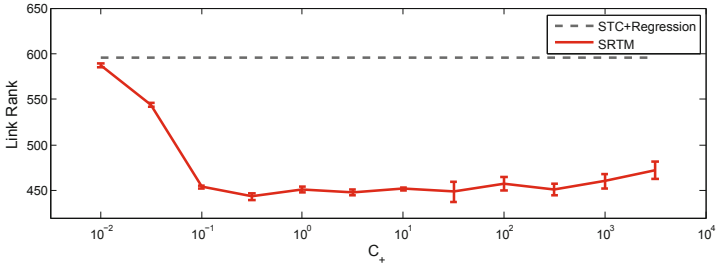

Fig. 5. Link rank of SRTM (red sold line) using different C_+ values and STC+Regression (black dash line). Both with 25 topics on the Cora Dataset. Note that C_- also changes with C_+.

To analyze the sensitivity, we fix a reasonable ratio $C_+ = 10C_-$ to balance the links[8] and tune C_+ for training SRTM with 25 topics on the Cora dataset. The link ranks for different C_+ values are shown in Fig. 5. We can see that SRTM performs best when C_+ is not too large nor too small, e.g., in the wide range between 0.1 and 100. When C_+ approaches zero SRTM collapses to a sparse topical coding followed by regression. On the other end, when C_+ grows large, the link part dominates the whole objective function. Thus, the behavior of SRTM approximates the matrix factorization approach for link prediction. SRTM does better link prediction, both utilizing words and links with a moderate C_+ than merely using any one of them. This verifies that SRTM successfully combines the knowledge of each part to get an overall better model.

5 Conclusions and Discussions

We present sparse relational topic models (SRTM), a non-probabilistic formulation of relational topic models to understand document networks and predict missing links. By relaxing the normalization constraints of probabilistic models and introducing appropriate regularization terms, SRTM can handle the common imbalance issues in real networks and efficiently learn sparse latent representations. SRTM admits a simple coordinate descent algorithm, and it can be naturally extended to capture all pairwise topic interactions for predicting links among document networks. Empirical results show that our models perform significantly better than probabilistic relational topic models in link prediction, training time, and discovering sparse representations.

The current batch algorithm to learn the topical dictionary and link likelihood model may cause limitations on applying SRTM to large-scale applications. Therefore, it is worth investigating stochastic gradient descent methods [23] in the future. Furthermore, though a restricted grid search works well as we have done in the experiments, in general it is hard to search for the optimal hyper-parameters for SRTM, and developing more efficient methods for hyper-parameter estimation is an interesting topic.

[8] As in the link prediction experiments, we sub-sample 0.2% of negative links as our training data.

References

1. Liben-Nowell, D., Kleinberg, J.: The link prediction problem for social networks. In: CIKM (2003)
2. Airoldi, E.M., Blei, D.M., Fienberg, S.E., Xing, E.P.: Mixed membership stochastic blockmodels. Journal of Machine Learning Research 9, 1981–2014 (2008)
3. Hoff, P., Raftery, A., Handcock, M.: Latent space approches to social network analysis. Journal of American Statistical Association 97, 1090–1098 (2002)
4. Hoff, P.: Modeling homophily and stochastic equivalence in symmetric relational data. In: NIPS (2007)
5. Miller, K., Griffiths, T., Jordan, M.: Nonparametric latent feature models for link prediction. In: NIPS (2009)
6. Zhu, J.: Max-margin nonparametric latent feature models for link prediction. In: ICML (2012)
7. Menon, A.K., Elkan, C.: Link prediction via matrix factorization. In: Gunopulos, D., Hofmann, T., Malerba, D., Vazirgiannis, M. (eds.) ECML PKDD 2011, Part II. LNCS, vol. 6912, pp. 437–452. Springer, Heidelberg (2011)
8. Chang, J., Blei, D.: Relational topic models for document networks. In: AISTATS (2009)
9. Blei, D., Ng, A., Jordan, M.: Latent Dirichlet allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
10. Lee, D., Seung, H.: Learning the parts of objects by non-negative matrix factorization. Nature 401, 788–791 (1999)
11. Zhu, J., Xing, E.: Sparse topical coding. In: UAI (2011)
12. Liu, Y., Niculescu-Mizil, A., Gryc, W.: Topic-link lda: Joint models of topic and author community. In: ICML (2009)
13. Fu, W., Wang, J., Li, Z., Lu, H., Ma, S.: Learning semantic motion patterns for dynamic scenes by improved sparse topical coding. In: ICME (2012)
14. Ji, R., Duan, L., Chen, J., Gao, W.: Towards compact topical descriptors. In: CVPR (2012)
15. Li, L.-J., Zhu, J., Su, H., Xing, E.P., Fei-Fei, L.: Multi-level structured image coding on high-dimensional image representation. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012, Part II. LNCS, vol. 7725, pp. 147–161. Springer, Heidelberg (2013)
16. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. Royal. Statist. Soc. B58, 267–288 (1996)
17. Hyvarinen, A.: Sparse code shrinkage: Denoising of nongaussian data by maximum likelihood estimation. Neural Computation 11, 1739–1768 (1999)
18. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press (2004)
19. Duchi, J., Shalev-Shwartz, S., Singer, Y., Chandra, T.: Efficient projections onto the ℓ_1-ball for learning in high dimensions. In: ICML (2008)
20. McCallum, A., Nigam, K., Rennie, J., Seymore, K.: Automating the construction of internet portals with machine learning. Information Retrieval (2000)
21. Craven, M., Dipasquo, D., Freitag, D., McCallum, A.: Learning to extract symbolic knowledge from the world wide web. In: AAAI (1998)
22. Jordan, M.I., Ghahramani, Z., Jaakkola, T., Saul, L.K.: An introduction to variational methods for graphical models. In: Jordan, M.I. (ed.) Learning in Graphical Models. MIT Press, Cambridge (1999)
23. Zhang, A., Zhu, J., Zhang, B.: Sparse online topic models. In: WWW (2013)