Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

# Relational Deep Learning: A Deep Latent Variable Model for Link Prediction

**Hao Wang, Xingjian Shi, Dit-Yan Yeung**
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong
hwangaz@cse.ust.hk, xshiab@connect.ust.hk, dyyeung@cse.ust.hk

## Abstract

Link prediction is a fundamental task in such areas as social network analysis, information retrieval, and bioinformatics. Usually link prediction methods use the link structures or node attributes as the sources of information. Recently, the relational topic model (RTM) and its variants have been proposed as hybrid methods that jointly model both sources of information and achieve very promising accuracy. However, the representations (features) learned by them are still not effective enough to represent the nodes (items). To address this problem, we generalize recent advances in deep learning from solely modeling i.i.d. sequences of attributes to jointly modeling graphs and non-i.i.d. sequences of attributes. Specifically, we follow the Bayesian deep learning framework and devise a hierarchical Bayesian model, called relational deep learning (RDL), to jointly model high-dimensional node attributes and link structures with layers of latent variables. Due to the multiple nonlinear transformations in RDL, standard variational inference is not applicable. We propose to utilize the product of Gaussians (PoG) structure in RDL to relate the inferences on different variables and derive a *generalized variational inference* algorithm for learning the variables and predicting the links. Experiments on three real-world datasets show that RDL works surprisingly well and significantly outperforms the state of the art.

## Introduction

With the rapid growth of social network services (SNS) and other Internet applications, network data have become very pervasive (Wang et al. 2016). For example, there exist hyperlinks among web pages, social relationships among friends in online social networks like Facebook, and citations among scientific articles. Link prediction, as a fundamental task for such network data, can help to recommend relevant pages for newly created websites, new friends in online social networks, or citations for newly written articles. Roughly speaking, existing link prediction methods can be categorized into three classes (Goldenberg et al. 2010): link-based methods, attribute-based methods, and hybrid methods. Link-based methods seek to model the link structures of networks in a principled way (Taskar et al. 2003; Airoldi et al. 2008), e.g., using latent variable models or linear algebraic formulations. Attribute-based methods (Doppa et al. 2009) view the link prediction problem as a supervised classification task where each instance corresponds to a pair of nodes in the network. Hybrid methods (Chang and Blei 2010; Chen et al. 2014) try to jointly model the link structures and node attributes in an attempt to get the best of both worlds.

Link-based methods, though powerful, account only for the link structures of networks. They ignore the node attributes which are in fact also useful for link prediction (Al Hasan et al. 2006; Doppa et al. 2009; Hunter et al. 2011). For example, the text and abstracts (text-based attributes) of scientific articles play a crucial role in determining the links of citation networks, the similarity and relevance of content in web pages often affect whether they link each other, and the profile descriptions in online social networks may be the sole source of information deciding how friends are recommended to new users. On the other hand, attribute-based methods first extract attribute-based features for pairs of nodes and pose link prediction as a classification problem. Although attribute-based methods can take node attributes into account and are easy to implement, they often involve tedious feature crafting which is very labor intensive. There are models that directly use the node attributes for link prediction (Hoff, Raftery, and Handcock 2002), but they fail to make meaningful prediction with high-dimensional attributes like text data, as mentioned in (Chang and Blei 2010).

On the other hand, by jointly modeling the node attributes and link structures, hybrid methods can get the best of both worlds and deliver state-of-the-art performance. They can fully integrate the node attributes into a principled model without the need for feature crafting. What's more, they can even infer the node attributes according to the link structures. This is impossible for both link-based and attribute-based methods. Among the hybrid methods, the relational topic model (RTM) (Chang and Blei 2010) integrates both node attributes and link structures into a principled probabilistic model and gives very promising accuracy. Subsequently, discriminative RTM (Chen et al. 2014) extends RTM by modeling topic interaction and using regularized Bayesian inference (RegBayes), leading to significant performance boost. However, the representations (features) that the current hybrid methods learn from the link structures and node attributes are still not effective enough.

As a separate research direction, recent advances in deep learning show that models like stacked denoising autoen-

coders (SDAE) (Vincent et al. 2010) and convolutional neural networks (CNN) (Krizhevsky, Sutskever, and Hinton 2012) have great potential to learn effective and compact representations in such fields as computer vision (Karpathy, Joulin, and Li 2014) and natural language processing (Salakhutdinov and Hinton 2009; Irsoy and Cardie 2014). However, conventional deep learning models often assume i.i.d. input and hence are incapable of modeling relational data (network data) and performing link prediction. Besides, the non-probabilistic formulations of deep learning models do not allow them to integrate relational data in a principled manner and perform Bayesian inference like RTM variants.

To address the problems, we follow the Bayesian deep learning framework (Wang and Yeung 2016) and devise a hierarchical Bayesian model, called relational deep learning (RDL), to jointly and deeply model high-dimensional node attributes and link structures with layers of latent variables. Unfortunately, due to the extreme nonlinearity of RDL, standard variational inference is not applicable. We therefore propose to utilize the product of Gaussians (PoG) structure in RDL to relate the inferences on different variables and derive a *generalized variational inference* (GVI) algorithm for learning the variables and predicting the links. Note that the value of GVI goes beyond RDL since it can be adapted to seamlessly unify arbitrary types of neural networks and Bayesian networks (with Bayesian treatment). The main contributions of this paper are summarized as follows:

- We devise a hierarchical Bayesian model, RDL, to seamlessly integrate the node attributes and link structures of network data and perform relational deep learning.
- Besides the learning algorithm for maximum a posteriori (MAP) estimation, a generalized variational inference algorithm is derived to handle the multiple nonlinear transformations, model the uncertainty, and perform joint learning in RDL.
- Experiments on three real-world datasets show that our model works surprisingly well and significantly outperforms the state of the art.

## Related Work

As mentioned in the previous section, deep learning models have been used for various applications showing great potential. However, very few attempts have been made for the link prediction problem, especially for the joint modeling of node attributes and link structures on network data, which is crucial for link prediction. To the best of our knowledge, RDL is the first deep learning model that incorporates the node attributes and link structures into a hierarchical Bayesian model for link prediction. For completeness, we review some recent work relevant to RDL.

In (Zeng et al. 2014), a deep model is built to solve the relation classification problem in which the relationships between words in a given sentence are classified. The approach adopted is essentially a combination of feature engineering and CNN, which cannot be directly used to handle the link prediction problem in relational (network) data. (Li et al. 2014; Wang, Cui, and Zhu 2016) deal with the link prediction problem in dynamic/static networks. However, they only take account of the link structure information of the networks to predict the future relationship while ignoring the node attributes. Doing so inevitably harms the predictive performance (Chang and Blei 2010; Hunter et al. 2011). DeepWalk (Perozzi, Al-Rfou, and Skiena 2014) is another model that deals with relational data using deep learning models. It uses local information obtained from truncated random walks and uses hierarchical softmax to learn the latent representation for each node by treating the walks as the equivalent of sentences. (Wang, Shi, and Yeung 2015) uses relational information to construct priors for generating representations. Although DeepWalk (Perozzi, Al-Rfou, and Skiena 2014) and (Wang, Shi, and Yeung 2015) are relevant to both relational data and deep learning, they are used for learning the low-dimensional representation for each node in the network instead of performing link prediction.

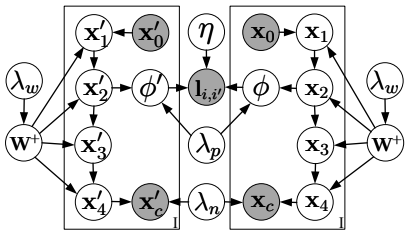


Figure 1: Graphical model of a 2-layer RDL ($L = 4$).

## Notation and Problem Formulation

The attributes of $I$ nodes are denoted by an $I$-by-$B$ matrix $\mathbf{X}_c$ where $B$ is the number of attributes (size of vocabulary) for each node. Each row $\mathbf{X}_{c,i*}$ is the bag-of-words representation for node $i$ if each node is a document (article). $\mathbf{W}_l$ and $\mathbf{b}_l$ are the weight matrix and bias vector, respectively, in layer $l$. $K_l$ is the number of hidden units in the $l$-th layer. $K = K_{\frac{L}{2}}$ is the dimensionality of item representations. $\mathbf{W}_{l,*n}$ denotes column $n$ of $\mathbf{W}_l$ and $L$ is the number of layers. For convenience, $\mathbf{W}^+$ is used to denote the collection of all layers of weight matrices and biases. Note that an $L/2$-layer SDAE corresponds to an $L$-layer network. $l_{i,i'}$ indicates the existence of links, where $l_{i,i'} = 1$ means there is a link between node $i$ and node $i'$. Similar to (Chang and Blei 2010), for both methodological and computational reasons, only observed links will be modeled in RDL (i.e., $l_{i,i'}$ is either 1 or unobserved). The task is to predict a new node's (for example, a document which is not in the training set) links to other nodes given the current link structures and node attributes. Note that the links from new nodes are not available in the training set. Hence link-based methods are not applicable in our problem setting.

## Model Formulation

In this section, we start with the introduction of RDL and then discuss two learning algorithms, MAP estimation and Bayesian treatment for this model.

### Relational Deep Learning

Using the probabilistic SDAE (pSDAE) in (Wang, Shi, and Yeung 2015; Wang, Wang, and Yeung 2015) as a building

block (Step 1 and 2 below), the generative process of RDL is defined as follows:

1. For each layer $l$ of the probabilistic SDAE network,
   (a) For each column $n$ of the weight matrix $\mathbf{W}_l$, draw
   $$\mathbf{W}_{l,*n} \sim \mathcal{N}(\mathbf{0}, \lambda_w^{-1}\mathbf{I}_{K_{l-1}}).$$
   (b) Draw the bias vector $\mathbf{b}_l \sim \mathcal{N}(\mathbf{0}, \lambda_w^{-1}\mathbf{I}_{K_l})$.
   (c) For each row $i$ of $\mathbf{X}_l$, draw
   $$\mathbf{X}_{l,i*} \sim \mathcal{N}(\sigma(\mathbf{X}_{l-1,i*}\mathbf{W}_l + \mathbf{b}_l), \lambda_s^{-1}\mathbf{I}_{K_l}).$$
2. For each item $i$, draw a clean input
   $$\mathbf{X}_{c,i*} \sim \mathcal{N}(\mathbf{X}_{L,i*}, \lambda_n^{-1}\mathbf{I}_B).$$
3. For each item $i$, generate features
   $$\boldsymbol{\phi}_i \sim \mathbf{h}(\boldsymbol{\phi}_i|\mathbf{X}^T_{\frac{L}{2},i*}, \lambda_p).$$
4. Draw the parameter $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \lambda_e^{-1}\mathbf{I}_K)$.
5. For each pair of items $(i, i')$ with an observed link, draw a binary link indicator
   $$l_{i,i'}|\boldsymbol{\phi}_i, \boldsymbol{\phi}_{i'} \sim \psi(\cdot|\boldsymbol{\phi}_i, \boldsymbol{\phi}_{i'}, \boldsymbol{\eta}).$$

Here $\lambda_w$, $\lambda_n$, $\lambda_p$, $\lambda_s$, and $\lambda_e$ are *hyperparameters*. $\mathbf{X}_{l,i*}$ and $\boldsymbol{\phi}_i$ are *latent variables* while $\boldsymbol{\eta}$ and $\mathbf{W}^+$ are *parameters*. For computational efficiency, we can also take $\lambda_s$ to infinity. $\mathbf{h}(\boldsymbol{\phi}_i|\mathbf{X}^T_{\frac{L}{2},i*}, \lambda_p)$ is a *feature generator distribution*. For example, it can be a Gaussian distribution $\mathcal{N}(\mathbf{X}^T_{\frac{L}{2},i*}, \lambda_p^{-1}\mathbf{I}_K)$ or a Dirichlet distribution $\mathrm{Dir}(\lambda_p\mathbf{X}^T_{\frac{L}{2},i*})$. The link probability function is defined as

$$\psi(l_{j,j'} = 1|\boldsymbol{\phi}_i, \boldsymbol{\phi}_{i'}, \boldsymbol{\eta}) = \sigma(\boldsymbol{\eta}^T(\boldsymbol{\phi}_i \circ \boldsymbol{\phi}_{i'})). \quad (1)$$

The graphical model of RDL is shown in Figure 1, where, for notational simplicity, we omit $\lambda_s$ and use $\boldsymbol{\phi}$, $\boldsymbol{\phi}'$, $\mathbf{x}_l$, and $\mathbf{x}_c$ in place of $\boldsymbol{\phi}_i$, $\boldsymbol{\phi}_{i'}$, $\mathbf{X}^T_{l,j*}$, and $\mathbf{X}^T_{c,j*}$, respectively.

## Learning Algorithms

We first derive an algorithm for the MAP estimation of the variables and then provide the GVI algorithm for the Bayesian treatment of RDL. Note that (Wang, Shi, and Yeung 2015; Wang, Wang, and Yeung 2015) provide *only MAP estimation* for pSDAE. Hence efficient *Bayesian treatment* and integration with network data are both *nontrivial* here.

**MAP Estimation** We derive below an EM-style algorithm for obtaining the MAP estimates when the feature generator distribution $\mathbf{h}(\boldsymbol{\phi}_i|\mathbf{X}^T_{\frac{L}{2},i*}, \lambda_p) = \mathcal{N}(\boldsymbol{\phi}_i|\mathbf{X}^T_{\frac{L}{2},i*}, \lambda_p^{-1}\mathbf{I}_K)$.

Maximizing the posterior probability is equivalent to maximizing the joint log-likelihood of $\{\mathbf{X}_l\}$, $\mathbf{X}_c$, $\{\mathbf{W}_l\}$, $\{\mathbf{b}_l\}$, $\{\boldsymbol{\phi}_i\}$, $\boldsymbol{\eta}$, and $\{l_{i,i'}\}$ given $\lambda_p$, $\lambda_e$, $\lambda_w$, $\lambda_s$, and $\lambda_n$:

$$\begin{aligned}\mathscr{L} = &-\frac{\lambda_w}{2}\sum_l(\|\mathbf{W}_l\|_F^2 + \|\mathbf{b}_l\|_2^2)\\ &-\frac{\lambda_p}{2}\sum_i\|\boldsymbol{\phi}_i - \mathbf{X}^T_{\frac{L}{2},i*}\|_2^2 - \frac{\lambda_n}{2}\sum_i\|\mathbf{X}_{L,i*} - \mathbf{X}_{c,i*}\|_2^2\\ &-\frac{\lambda_s}{2}\sum_l\sum_i\|\sigma(\mathbf{X}_{l-1,i*}\mathbf{W}_l + \mathbf{b}_l) - \mathbf{X}_{l,i*}\|_2^2\\ &-\frac{\lambda_e}{2}\|\boldsymbol{\eta}\|_2^2 + \sum_{l_{i,i'}=1}\log\sigma(\boldsymbol{\eta}^T(\boldsymbol{\phi}_i \circ \boldsymbol{\phi}_{i'})). \quad (2)\end{aligned}$$

Update rules can be derived based on gradients with respect to different variables.

Another choice of the distribution $\mathbf{h}(\boldsymbol{\phi}_i|\mathbf{X}^T_{\frac{L}{2},i*}, \lambda_p)$ is the Dirichlet distribution $\mathrm{Dir}(\boldsymbol{\phi}_i|\lambda_p\mathbf{X}^T_{\frac{L}{2},i*})$, which makes the joint log-likelihood more complex.

**Bayesian Treatment** The MAP estimation approach only computes a point estimate of the prediction result without modeling the variance (uncertainty), which is often important not only for robust prediction but also for applications like ensemble learning, reinforcement learning (bandits included), and active learning. To also take the variance into consideration we need a full Bayesian treatment of our model. Unfortunately, due to multiple nonlinear transformations in RDL, standard variational inference (Bishop 2006) cannot be used for the Bayesian inference of RDL. To solve the problem, we propose to utilize the *product of Gaussians (PoG) structure* in RDL to relate the inferences on $\mathbf{W}^+$, $\boldsymbol{\eta}$, and $\{\boldsymbol{\phi}_i\}$. A generalized variational inference algorithm for learning the variables (i.e., latent variables and parameters) and predicting the links is designed. Note that GVI goes beyond RDL and is general enough to unify other neural networks and Bayesian networks. Again, we assume the feature generator $\mathbf{h}(\boldsymbol{\phi}|\mathbf{X}^T_{\frac{L}{2},i*}, \lambda_p) = \mathcal{N}(\mathbf{X}^T_{\frac{L}{2},i*}, \lambda_p^{-1}\mathbf{I}_K)$ here, and the derivation is similar for other choices.

***GVI Framework***: We follow the procedure of variational inference to update the logarithm of variational distributions as the expectation of the joint log-likelihood in Equation (2). Specifically, we have the following general update rule:

$$\log q_j^*(\mathbf{Z}_j) = \mathrm{E}_{i\neq j}[\log p(\mathbf{X}_0, \mathbf{X}_c, \mathbf{Z})] + const,$$

where $\mathbf{Z}$ denotes the collection of all latent variables and parameters to learn, i.e., $\mathbf{W}^+$, $\{\boldsymbol{\phi}_i\}$, $\boldsymbol{\eta}$, and $\xi_{ii'}$ (note that $\xi_{ii'}$ is the *variational parameter* to approximate the sigmoid function $\sigma(\cdot)$ in Equation (1)). The $j$-th part of $\mathbf{Z}$ (e.g., $\boldsymbol{\eta}$) is denoted by $\mathbf{Z}_j$ with $q_j^*(\mathbf{Z}_j)$ as its corresponding variational distribution.

***Learning*** $\{\boldsymbol{\phi}_i\}$: To learn $\{\boldsymbol{\phi}_i\}$, we write down terms in $\mathscr{L}$ relevant to $\{\boldsymbol{\phi}_i\}$ as (for simplicity $\lambda_s$ is taken to infinity):

$$\begin{aligned}\mathscr{L}_{\{\boldsymbol{\phi}_i\}} = &-\frac{\lambda_p}{2}\sum_i\|\boldsymbol{\phi}_i - f_e(\mathbf{X}_{0,i*}, \mathbf{w})^T\|_2^2\\ &+\sum_{l_{i,i'}=1}\log\sigma(\boldsymbol{\eta}^T(\boldsymbol{\phi}_i \circ \boldsymbol{\phi}_{i'})) + const, \quad (3)\end{aligned}$$

where $f_e(\mathbf{X}_{0,i*}, \mathbf{w}) = \mathbf{X}_{\frac{L}{2},i*}$ in pSDAE. Since the first two terms can both be approximated by Gaussians, $\boldsymbol{\phi}_i$ can be approximated using the *product of Gaussians* (still a Gaussian distribution). We take one term at a time.

(a) ***First Gaussian***: If we *omit the second term* and use the vectorization $\mathbf{w} = vec(\mathbf{W}^+) = (\mathbf{w}_e, \mathbf{w}_d)^T$ ($\mathbf{w}_e$ and $\mathbf{w}_d$ are the weights of the encoder and decoder), the features

$$\boldsymbol{\phi}_i \sim \mathcal{N}(f_e(\mathbf{X}_{0,i*}, \mathbf{w})^T, \lambda_p^{-1}\mathbf{I}_K),$$

we can further approximate the distribution of $\boldsymbol{\phi}_i$:

$$q_1(\boldsymbol{\phi}_i^{(j)}|\mathbf{X}_{0,i*}) = \int p(\boldsymbol{\phi}_i^{(j)}|\mathbf{X}_{0,i*}, \mathbf{w}_e^{(j)})q(\mathbf{w}_e^{(j)})d\mathbf{w}_e^{(j)},$$

where $\boldsymbol{\phi}_i^{(j)}$ is the $j$-th element of $\boldsymbol{\phi}_i$ and $\mathbf{w}_e^{(j)}$ is a sub-vector of $\mathbf{w}_e$ which corresponds to the computation of $\boldsymbol{\phi}_i^{(j)}$.

Thus we have the *first Gaussian* $q_1(\boldsymbol{\phi}_i|\mathbf{X}_{0,i*}) = \mathcal{N}(\boldsymbol{\phi}_i|\mathbf{m}_i, \mathbf{S}_i)$ where

$$\mathbf{m}_i = f_e(\mathbf{X}_{0,i*}, \mathbf{w}),$$

and $\mathbf{S}_i$ is a diagonal matrix where

$$\mathbf{S}_{i,jj} = \lambda_p^{-1} + \mathbf{g}_{ij}^T (\mathbf{A}_e^{(j)})^{-1} \mathbf{g}_{ij},$$

where $\mathbf{g}_{ij}$ and $\mathbf{A}_e^{(j)}$ are the first-order and second-order information of the pSDAE.

*Remark*: The mean of $q_1(\boldsymbol{\phi}_i|\mathbf{X}_{0,i*})$ is the encoding of the input, and the covariance matrix depends on the second-order information of the pSDAE network.

(b) ***Second Gaussian***: For the second term of $\mathscr{L}_{\{\boldsymbol{\phi}_i\}}$, we can use the variational lower bound $\sigma(a) \geq \sigma(\xi) \exp\{(a - \xi)/2 - \lambda(\xi)(a^2 - \xi^2)\}$, where $\lambda(\xi) = \frac{1}{2\xi}(\sigma(\xi) - \frac{1}{2})$.

By replacing $\sigma(\cdot)$ in Equation (3) and completing the square for the second term, we can get the *second Gaussian*

$$\begin{aligned} q_2(\boldsymbol{\phi}_i|\mathbf{X}_{0,i*}) &= \mathcal{N}(\boldsymbol{\phi}_i|\mathbf{m}_i', \mathbf{S}_i') \\ \mathbf{m}_i' &= \frac{1}{2}\mathbf{S}_i' \sum_{l_{i,i'}=1} \mathrm{E}(\boldsymbol{\eta} \circ \boldsymbol{\phi}_{i'}) \\ \mathbf{S}_i'^{-1} &= 2 \sum_{l_{i,i'}=1} \lambda(\xi_{ii'}) \mathrm{E}((\boldsymbol{\eta} \circ \boldsymbol{\phi}_{i'})(\boldsymbol{\eta} \circ \boldsymbol{\phi}_{i'})^T), \end{aligned}$$

where the expectations are taken over the current $q(\boldsymbol{\eta})$ and $q(\boldsymbol{\phi}_{i'}|\mathbf{X}_{0,i'*})$.

*Remark*: The covariance matrix of $q_2(\boldsymbol{\phi}_i|\mathbf{X}_{0,i*})$ depends on a weighted sum of the covariance of $\boldsymbol{\eta} \circ \boldsymbol{\phi}_{i'}$, and the mean depends on the features of linked nodes transformed by $\mathbf{S}_i'$.

(c) ***Product of Gaussians***: Finally we can get the update rules for $q(\boldsymbol{\phi}_i|\mathbf{X}_{0,i*})$ according to $q_1(\boldsymbol{\phi}_i|\mathbf{X}_{0,i*})$ and $q_2(\boldsymbol{\phi}_i|\mathbf{X}_{0,i*})$:

$$\begin{aligned} q(\boldsymbol{\phi}_i|\mathbf{X}_{0,i*}) &\approx \mathcal{N}(\boldsymbol{\phi}_i|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \\ \boldsymbol{\mu}_i &= \boldsymbol{\Sigma}_i(\mathbf{S}_i^{-1}\mathbf{m}_i + \mathbf{S}_i'^{-1}\mathbf{m}_i') \\ \boldsymbol{\Sigma}_i^{-1} &= \mathbf{S}_i^{-1} + \mathbf{S}_i'^{-1}. \end{aligned}$$

*Remark*: The first Gaussian absorbs content information and the second absorbs link information. The final update rule as the product of these two Gaussians then summarizes both information sources and yields more powerful features.

***Learning $\mathbf{W}^+$ and $\boldsymbol{\eta}$***: Note that besides the use of PoG structure in our model, another difference from conventional variational inference, where $\log q_j^*(\mathbf{Z}_j)$ is reformulated as the closed form of the logarithm of the variational distribution, is that $q(\mathbf{W}^+)$ is updated based on *Laplace approximation*. The motivation behind is that variational inference tends to underestimate uncertainty (variance) while Laplace approximation tends to overestimate it. Hence incorporating Laplace approximation inside the variational inference would not only successfully handle the multiple nonlinear transformations but also to some degree counteract these two effects (underestimate/overestimate of uncertainty) and yield better uncertainty estimation. That is why we call the algorithm *generalized variational inference*.

***Prediction***: The link probabilities can be computed as $\psi(l_{l,l'} = 1|\boldsymbol{\phi}_i, \boldsymbol{\phi}_{i'}, \boldsymbol{\eta}) = \sigma(\kappa(\sigma_s^2)\mu_s)$, where $\kappa(\sigma_s^2) = (1 + \pi\sigma_s^2/8)^{-1/2}$, $\sigma_s^2$ is the variance involving the distribution of $\boldsymbol{\eta}$ and $\{\boldsymbol{\phi}_i\}$, and $\mu_s$ is the mean $\boldsymbol{\eta}^T(\boldsymbol{\phi}_i \circ \boldsymbol{\phi}_{i'})$ of the prediction. Note that since the final prediction takes both the mean and variance into account, the estimation is expected to be more robust.

# Experiments

Here we present both quantitative and qualitative experiment results on three datasets from different domains to demonstrate the effectiveness of RDL for link prediction.

## Datasets and Evaluation Metrics

We use three datasets, two from CiteULike[1] and one from arXiv[2], in our experiments. The first two datasets are from (Wang, Chen, and Li 2013). They were collected in different ways, specifically, with different scales and different degrees of sparsity to mimic different practical situations. The first dataset, *citeulike-a*, is mostly from (Wang and Blei 2011) and the second dataset, *citeulike-t*, was collected independently of the first one (Wang, Chen, and Li 2013). They manually selected 273 seed tags and collected all the articles with at least one of those tags. For *citeulike-a*, there are 16,980 nodes (documents) and 44,709 links (citations) among them. For *citeulike-t* the numbers are 25,975 and 32,565. The last dataset, *arXiv*, is from the SNAP datasets (Leskovec and Krevl 2014). The number of nodes is 27,770 and the number of observed links is 352,807. We use the bags of words from the documents as node attributes. The vocabulary size, which is denoted as $B$, for the three datasets is 8,000, 20,000, and 8,000 respectively.

As in (Chang and Blei 2010; Hunter et al. 2011; Chen et al. 2014) we use *link rank* and *AUC* (area under the ROC curve) as evaluation metrics. Link rank is defined as the average rank of the test nodes (documents) to the training nodes (Chen et al. 2014). AUC is computed for every test node and the average values are reported. Therefore lower link rank and higher AUC indicate better predictive performance.

## Baselines and Experiment Setup

Note that as mentioned in (Al Hasan et al. 2006; Doppa et al. 2009; Hunter et al. 2011), hybrid methods clearly outperform link-based and attribute-based methods. Besides, as mentioned before, links from the new nodes are not available in the training set, making link-based methods inapplicable in this experiment setting. Due to space constraints, we focus only on comparison among hybrid methods in most experiments. The hybrid models used for comparison are listed below:

- **CMF**: Collective Matrix Factorization (Singh and Gordon 2008) simultaneously factorizes multiple matrices (i.e., the adjacency matrix consisting of $l_{i,i'}$ and $\mathbf{X}_c$ in this paper).

[1] CiteULike allows users to create their own collections of articles. More details about the CiteULike data can be found at http://www.citeulike.org.

[2] http://www.arxiv.org

Table 1: Performance of RDL with different number of layers (MAP)

| | Link Rank | | | AUC | | |
|---|---|---|---|---|---|---|
| | RDL-1 | RDL-2 | RDL-3 | RDL-1 | RDL-2 | RDL-3 |
| *citeulike-a* | 825.74 | 495.97 | **488.41** | 0.939 | **0.964** | 0.963 |
| *citeulike-t* | 2060.17 | 951.31 | **912.43** | 0.894 | 0.954 | **0.955** |
| *arXiv* | 5241.97 | **2080.72** | 2730.08 | 0.755 | **0.905** | 0.855 |

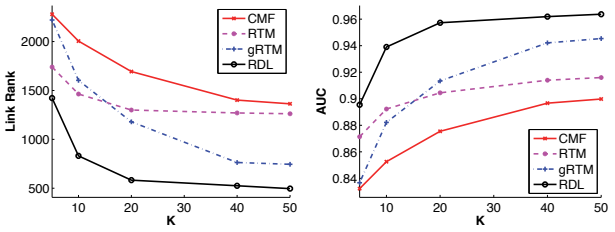


Figure 2: Link rank and AUC of compared models for *citeulike-a*. A 2-layer RDL is used.

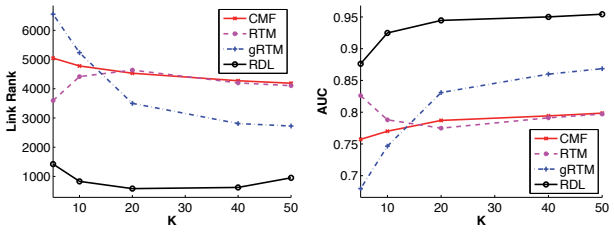


Figure 3: Link rank and AUC of compared models for *citeulike-t*. A 2-layer RDL is used.

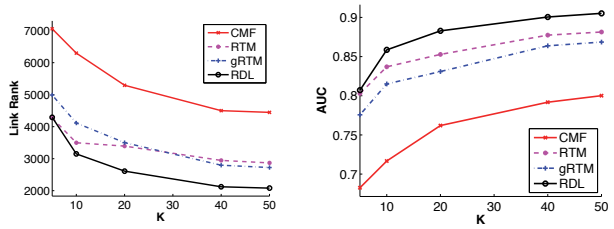


Figure 4: Link rank and AUC of compared models for *arXiv*. A 2-layer RDL is used.

- **RTM**: Relational Topic Model (Chang and Blei 2010) jointly models the node attributes (text of documents) and link structures.
- **gRTM**: generalized Relational Topic Model (also called discriminative RTM) (Chen et al. 2014) extends RTM by modeling topic interaction and using regularized Bayesian inference (RegBayes), which leads to significant performance boost.
- **RDL**: Relational Deep Learning is our proposed model. It deeply and jointly models the node attributes and link structures using a hierarchical Bayesian model with layers of latent variables. It can provide different levels of model complexity by varying the depth $L$.

In the experiments, we first use a validation set to find the optimal hyperparameters for CMF, RTM, gRTM, and RDL. For CMF, we set the regularization hyperparameters for the latent factors of different contexts to 10. After the grid search, we find that CMF performs best when the weights for the adjacency matrix and content matrix (BOW) are 8 and 2 for all three datasets. We find that RTM and gRTM achieve the best performance when $c = 12$, $\alpha = 1$, and the sampling ratio for unobserved links is set to $0.1\%$. For RDL we use the Gaussian feature generator distribution and network structures of $B$-$K$, $B$-100-$K$, and $B$-100-100-$K$. For all models we vary the representation dimensionality $K$ from 5 to 50. We randomly select $80\%$ of the nodes as the training set and use the rest as the test set. The experiments are repeated 5 times and the average performance is reported.

## Performance Evaluation

The left of Figure 2, 3, and 4 shows the link rank when $K$ is set to 5, 10, 20, 40, and 50 for the three datasets *citeulike-a*, *citeulike-t*, and *arXiv*. As we can see, RTM is able to achieve a lower link rank and outperform gRTM when $K$ is small, but gRTM can outperform RTM by a large margin when $K$ is large enough. CMF achieves the poorest performance in *citeulike-a* and *arXiv*. In *citeulike-t* it is able to achieve similar performance as RTM. As for RDL, it outperforms all the other models significantly. For example, when $K = 50$, the link rank for gRTM and RDL is 744 and 495 respectively in *citeulike-a*. For *citeulike-t* and *arXiv*, the margins are even larger (2724 versus 951 and 2724 versus 2080).

Similar phenomena can be observed for AUC on the right of Figure 2, 3, and 4. For the RTM variants, when $K$ is small RTM is better, and when $K$ is large gRTM prevails. The difference is that in *arXiv*, gRTM is not able to outperform RTM even when $K = 50$. We can also see that in terms of AUC, RDL can still significantly outperform the baselines. In the case of $K = 50$, the AUC for gRTM and RDL is $94.53\%$ and $96.37\%$ respectively for *citeulike-a*. Similarly, the margins are even larger for *citeulike-t* and *arXiv* ($86.85\%$ versus $96.37\%$ and $86.78\%$ versus $90.52\%$).

Table 1 shows the link rank and AUC of RDL when $K = 50$ and $L$ is set to 2, 4, and 6 (corresponding to 1-layer, 2-layer, and 3-layer RDL) when MAP estimation is used. As we can see, for *citeulike-a*, 3-layer RDL is able to achieve the lowest link rank while 2-layer RDL performs the best in terms of AUC. For *citeulike-t*, 3-layer RDL is able to achieve both the lowest link rank and the highest AUC. For *arXiv*, 2-layer RDL has the best predictive performance in terms of both link rank and AUC. The performance slightly degrades

Table 2: Performance of RDL with different number of layers (Bayesian treatment)

| | Link Rank | | | AUC | | |
|---|---|---|---|---|---|---|
| | RDL-1 | RDL-2 | RDL-3 | RDL-1 | RDL-2 | RDL-3 |
| *citeulike-a* | 789.85 | 473.59 | **471.47** | 0.946 | **0.971** | 0.970 |
| *citeulike-t* | 1904.83 | 911.31 | **867.78** | 0.906 | 0.956 | **0.960** |
| *arXiv* | 4965.01 | **1982.84** | 2612.12 | 0.801 | **0.914** | 0.866 |

Table 3: Link rank of baselines (the first 3 columns) and RDL variants (the last 4 columns) on three datasets ($L = 4$)

| Method | VAE+BLR | VFAE+BLR | SDAE+BLR | MAPRDL | BSDAE1+BLR | BSDAE2+BLR | BayesRDL |
|---|---|---|---|---|---|---|---|
| *citeulike-a* | 980.81 | 960.15 | 992.48 | 495.97 | 849.02 | 761.57 | **473.59** |
| *citeulike-t* | 1599.62 | 1531.16 | 1356.85 | 951.31 | 1341.15 | 1310.12 | **911.31** |
| *arXiv* | 3367.25 | 3316.29 | 2916.18 | 2028.72 | 2947.79 | 2708.17 | **1982.84** |

when $L$ further increases to 6 possibly due to overfitting.

Similarly, Table 2 shows the link rank and AUC of RDL when $K = 50$ and $L$ is set to 2, 4, and 6 when Bayesian treatment (GVI) is used. The results are consistent with those of MAP estimation. With the Bayesian treatment, prediction is more robust when both the mean and variance are taken into account, yielding a relative boost of about $5\%$ over RDL with MAP estimation.

Table 3 shows the link rank for different AE variants (with the same network structures) and RDL variants when $L = 4$ and $K = 50$. As we can see, the variational autoencoder (Kingma and Welling 2013) combined with Bayesian logistic regression (VAE+BLR), the variational fair autoencoder (Louizos et al. 2016) combined with Bayesian logistic regression (VFAE+BLR), and the stacked denoising autoencoder combined with Bayesian logistic regression (SDAE+BLR) achieve similar link rank. The Bayesian SDAE (pSDAE with our proposed Bayesian treatment) with Bayesian logistic regression (BSDAE+BLR) can outperform them three (these AE variants are not hybrid models since node attributes and link structures are not jointly modeled). Here BSDAE1+BLR uses only the mean produced by Bayesian SDAE as features in BLR, and BSDAE2+BLR uses both the mean and variance. The performance gap between BSDAE1+BLR and BSDAE2+BLR *verifies the effectiveness of BSDAE's estimated variance*. As the strongest models, RDL with MAP (MAPRDL) significantly outperforms the variants above and RDL with Bayesian treatment (BayesRDL) is able to further boost the performance. Note that the performance gap between BSDAE2+BLR and BayesRDL *verifies the importance of BayesRDL's joint training*. In this experiment, we use a variant of VFAE without nuisance variables (Louizos et al. 2016) in the semi-supervised setting (with the number of links connected to training nodes as targets) to learn the representations.

## Case Study

To gain a better insight into the difference between RDL and RTM, we select a test node (article) $t$ with the title 'From DNA sequence to transcriptional behaviour: a quantitative approach' as an example to visualize the latent factors (features $\phi_i$) learned by RDL and RTM using t-SNE (Van der Maaten and Hinton 2008). As shown in Figure 5, the red stars are the latent factors of articles with links to the test node $t$. The blue circles correspond to the latent factors of randomly sampled nodes without links to node $t$. As we can see, the nodes with links to node $t$ are scattered all over the plot for RTM. However, they are well separated from the ones without links to node $t$ in RDL. Moreover, interestingly in the RDL plot, the blue circles roughly form two clusters. Looking into the data, we find that the small cluster on the left consists of articles written in German, which are rare in the datasets. The large one in the middle corresponds to some bestselling books like 'The 4-Hour Work Week: Escape 9-5, Live Anywhere, and Join the New Rich' and 'Mary Bell's Complete Dehydrator Cookbook'.

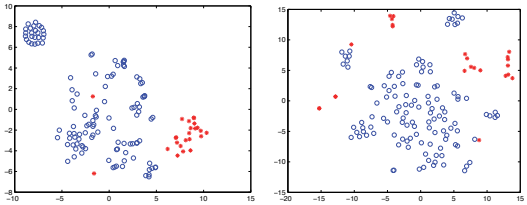

Figure 5: t-SNE visualization of latent factors learned by RDL (left) and RTM (right).

## Conclusion

In this paper we propose a hierarchical Bayesian model, RDL, to jointly and deeply model the node attributes and link structures of network data. Besides learning the model using MAP estimation, to cope with the multiple nonlinear transformations in RDL, we propose to utilize the PoG structure in RDL to relate the inferences on different variables and derive the GVI algorithm (that can be adapted for arbitrary neural networks and Bayesian networks) for learning the variables and prediction. Experiments on three real-world datasets show that RDL can significantly advance the state of the art.

The nature of Bayesian formulation makes it convenient to extend RDL to incorporate other auxiliary information for link prediction. Besides, RDL can also be extended naturally to handle multi-relational data (multiple networks). A multi-relational extension of RDL can not only jointly model multiple networks and boost the predictive performance, but it can also discover the relationships between different net-

works. Another interesting direction would be to adapt GVI to unify other neural networks (e.g., CNN) and other Bayesian networks (e.g., probabilistic topic models and probabilistic matrix factorization) for other tasks (e.g., text modeling and recommendation). We can also replace pSDAE with the recently proposed natural-parameter networks (Wang, Shi, and Yeung 2016) to improve efficiency and accuracy. Additionally, with the uncertainty modeled, Bayesian RDL is expected to perform much better for link prediction in settings like active learning and bandits. The possible extensions above will be pursued in our future work.

## Acknowledgments

This research has been supported by General Research Fund 16207316 from the Research Grants Council of Hong Kong.

## References

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

Al Hasan, M.; Chaoji, V.; Salem, S.; and Zaki, M. 2006. Link prediction using supervised learning. In *SDM: Workshop on Link Analysis, Counter-terrorism and Security*.

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Chang, J., and Blei, D. M. 2010. Hierarchical relational models for document networks. *The Annals of Applied Statistics* 124–150.

Chen, N.; Zhu, J.; Xia, F.; and Zhang, B. 2014. Discriminative relational topic models. *PAMI* 37:973–986.

Doppa, J. R.; Yu, J.; Tadepalli, P.; and Getoor, L. 2009. Chance-constrained programs for link prediction. In *NIPS Workshop on Analyzing Networks and Learning with Graphs*.

Goldenberg, A.; Zheng, A. X.; Fienberg, S. E.; and Airoldi, E. M. 2010. A survey of statistical network models. *Foundations and Trends in Machine Learning* 2(2):129–233.

Hoff, P. D.; Raftery, A. E.; and Handcock, M. S. 2002. Latent space approaches to social network analysis. *JASA* 97(460):1090–1098.

Hunter, D.; Smyth, P.; Vu, D. Q.; and Asuncion, A. U. 2011. Dynamic egocentric models for citation networks. In *ICML*, 857–864.

Irsoy, O., and Cardie, C. 2014. Deep recursive neural networks for compositionality in language. In *NIPS*, 2096–2104.

Karpathy, A.; Joulin, A.; and Li, F. 2014. Deep fragment embeddings for bidirectional image sentence mapping. In *NIPS*, 1889–1897.

Kingma, D. P., and Welling, M. 2013. Auto-encoding variational Bayes. *CoRR* abs/1312.6114.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet classification with deep convolutional neural networks. In *NIPS*, 1106–1114.

Leskovec, J., and Krevl, A. 2014. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data.

Li, X.; Du, N.; Li, H.; Li, K.; Gao, J.; and Zhang, A. 2014. A deep learning approach to link prediction in dynamic networks. In *SDM*, 289–297.

Louizos, C.; Swersky, K.; Li, Y.; Welling, M.; and Zemel, R. 2016. The variational fair auto encoder. In *ICLR*.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. DeepWalk: online learning of social representations. In *KDD*, 701–710.

Salakhutdinov, R., and Hinton, G. E. 2009. Semantic hashing. *Int. J. Approx. Reasoning* 50(7):969–978.

Singh, A. P., and Gordon, G. J. 2008. Relational learning via collective matrix factorization. In *KDD*, 650–658.

Taskar, B.; Wong, M. F.; Abbeel, P.; and Koller, D. 2003. Link prediction in relational data. In *NIPS*, 659–666.

Van der Maaten, L., and Hinton, G. 2008. Visualizing data using t-SNE. *JMLR* 9:2579–2605.

Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; and Manzagol, P.-A. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *JMLR* 11:3371–3408.

Wang, C., and Blei, D. M. 2011. Collaborative topic modeling for recommending scientific articles. In *KDD*, 448–456.

Wang, H., and Yeung, D. 2016. Towards Bayesian deep learning: A framework and some existing methods. *TKDE* 27(5):1343–1355.

Wang, S.; Chen, Z.; Liu, B.; and Emery, S. 2016. Identifying search keywords for finding relevant social media posts. In *AAAI*, 3052–3058.

Wang, H.; Chen, B.; and Li, W.-J. 2013. Collaborative topic regression with social regularization for tag recommendation. In *IJCAI*, 2719–2725.

Wang, D.; Cui, P.; and Zhu, W. 2016. Structural deep network embedding. In *KDD*, 1225–1234.

Wang, H.; Shi, X.; and Yeung, D. 2015. Relational stacked denoising autoencoder for tag recommendation. In *AAAI*, 3052–3058.

Wang, H.; Shi, X.; and Yeung, D.-Y. 2016. Natural-parameter networks: A class of probabilistic neural networks. In *NIPS*, 118–126.

Wang, H.; Wang, N.; and Yeung, D. 2015. Collaborative deep learning for recommender systems. In *KDD*, 1235–1244.

Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; and Zhao, J. 2014. Relation classification via convolutional deep neural network. In *COLING*, 2335–2344.