Supplementary Materials for Relational Deep Learning: A Deep Latent Variable Model for Link Prediction

1 MAP Estimation for RDL

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 λ_p, λ_e, λ_w, λ_s, and λ_n:

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

If λ_s goes to infinity, the likelihood becomes:

$$\begin{aligned}\mathscr{L} =& -\frac{\lambda_e}{2}\|\boldsymbol{\eta}\|_2^2 - \frac{\lambda_w}{2}\sum_l(\|\mathbf{W}_l\|_F^2 + \|\mathbf{b}_l\|_2^2) - \frac{\lambda_p}{2}\sum_i \|\boldsymbol{\phi}_i - f_e(\mathbf{X}_{0,i*}, \mathbf{W}^+)^T\|_2^2 \\ & - \frac{\lambda_n}{2}\sum_i \|f_r(\mathbf{X}_{0,i*}, \mathbf{W}^+) - \mathbf{X}_{c,i*}\|_2^2 + \sum_{l_{i,i'}=1} \log\sigma(\boldsymbol{\eta}^T(\boldsymbol{\phi}_i \circ \boldsymbol{\phi}_{i'})),\end{aligned} \tag{1}$$

where the encoder function $f_e(\cdot, \mathbf{W}^+)$ takes the corrupted content vector $\mathbf{X}_{0,i*}$ of item i as input and computes the encoding of the item, and the function $f_r(\cdot, \mathbf{W}^+)$ also takes $\mathbf{X}_{0,i*}$ as input, computes the encoding and then the reconstructed content vector of item i. For example, if the number of layers $L = 6$, $f_e(\mathbf{X}_{0,i*}, \mathbf{W}^+)$ is the output of the third layer while $f_r(\mathbf{X}_{0,i*}, \mathbf{W}^+)$ is the output of the sixth layer.

For $\boldsymbol{\phi}_i$ and $\boldsymbol{\eta}$, since we cannot directly take the gradients of $\mathscr{L}$ with respect to $\boldsymbol{\phi}_i$ or $\boldsymbol{\eta}$ and set them to zero, gradient descent is used given the current $\mathbf{W}^+$. The gradient of $\mathscr{L}$ with respect to $\boldsymbol{\eta}$ is:

$$\begin{aligned}\nabla_{\boldsymbol{\phi}_i}\mathscr{L} &= \sum_{l_{i,i'}=1}(1 - \sigma(\boldsymbol{\eta}^T(\boldsymbol{\phi}_i \circ \boldsymbol{\phi}_{i'})))(\boldsymbol{\eta} \circ \boldsymbol{\phi}_{i'}) - \lambda_p(\boldsymbol{\phi}_i - f_e(\mathbf{X}_{0,i*}, \mathbf{W}^+)^T), \\ \nabla_{\boldsymbol{\eta}}\mathscr{L} &= \sum_{l_{i,i'}=1}(1 - \sigma(\boldsymbol{\eta}^T(\boldsymbol{\phi}_i \circ \boldsymbol{\phi}_{i'})))(\boldsymbol{\phi}_i \circ \boldsymbol{\phi}_{i'}) - \lambda_e\boldsymbol{\eta}.\end{aligned}$$

Given $\boldsymbol{\phi}_i$ and $\boldsymbol{\eta}$, we can learn the weights $\mathbf{W}_l$ and biases $\mathbf{b}_l$ for each layer using the back-propagation learning algorithm.

Another choice of the distribution $\mathbf{h}(\boldsymbol{\phi}_i|\mathbf{X}^T_{\frac{L}{2},i*}, \lambda_p)$ is the Dirichlet distribution $\mathrm{Dir}(\boldsymbol{\phi}_i|\lambda_p\mathbf{X}^T_{\frac{L}{2},i*})$. In this more complex case, the joint log-likelihood in Equation (1) would become:

$$\begin{aligned}\mathscr{L} =& -\frac{\lambda_e}{2}\|\boldsymbol{\eta}\|_2^2 - \frac{\lambda_w}{2}\sum_l(\|\mathbf{W}_l\|_F^2 + \|\mathbf{b}_l\|_2^2) + \sum_i\sum_k(\lambda_p f_e^{(k)}(\mathbf{X}_{0,i*}, \mathbf{W}^+) - 1)\log\boldsymbol{\phi}_i^{(k)} \\ & + \sum_i \Gamma(\sum_k f_e^{(k)}(\mathbf{X}_{0,i*}, \mathbf{W}^+)) - \sum_i\sum_k \Gamma(f_e^{(k)}(\mathbf{X}_{0,i*}, \mathbf{W}^+)) \\ & - \frac{\lambda_n}{2}\sum_i \|f_r(\mathbf{X}_{0,i*}, \mathbf{W}^+) - \mathbf{X}_{c,i*}\|_2^2 + \sum_{l_{i,i'}=1} \log\sigma(\boldsymbol{\eta}^T(\boldsymbol{\phi}_i \circ \boldsymbol{\phi}_{i'})),\end{aligned} \tag{2}$$

where $f_e^{(k)}(\mathbf{X}_{0,i*}, \mathbf{W}^+)$ is the k-th element of $f_e(\mathbf{X}_{0,i*}, \mathbf{W}^+)$ and $\boldsymbol{\phi}_i^{(k)}$ is the k-th element of $\boldsymbol{\phi}_i$. $\Gamma(\cdot)$ is the gamma function. The gradient of $\mathscr{L}$ with respect to $\boldsymbol{\phi}_i$ becomes:

$$\nabla_{\boldsymbol{\phi}_i}\mathscr{L} = \sum_{l_{i,i'}=1}(1 - \sigma(\boldsymbol{\eta}^T(\boldsymbol{\phi}_i \circ \boldsymbol{\phi}_{i'})))(\boldsymbol{\eta} \circ \boldsymbol{\phi}_{i'}) + (\lambda_p f_e(\mathbf{X}_{0,i*}, \mathbf{W}^+)^T - \mathbf{1}) \circ \boldsymbol{\phi}_i^{-1},$$

where $\mathbf{1}$ is a vector of all 1's, $\circ$ is the Hadamard product (element-wise product) of two vectors, and $\boldsymbol{\phi}_i^{-1}$ denotes a vector where the k-th element is $1/\boldsymbol{\phi}_i^{(k)}$. We normalize $\boldsymbol{\phi}_i$ to ensure that the sum of all elements is 1 and all elements are positive after each update. Also, when using back-propagation to update $\mathbf{W}^+$, the gradient $\mathbf{g}_{e,\frac{L}{2}}$ with respect to $f_e(\mathbf{X}_{0,i*}, \mathbf{W}^+)$ becomes

$$\mathbf{g}_{e,\frac{L}{2}} = \lambda_p \log \boldsymbol{\phi}_i + \sum_i \psi(\sum_k f_e^{(k)}(\mathbf{X}_{0,i*}, \mathbf{W}^+)) - \sum_i \psi(f_e(\mathbf{X}_{0,i*}, \mathbf{W}^+)) + \mathbf{g}_{r,\frac{L}{2}},$$

where $\log \boldsymbol{\phi}_i$ is a vector where the k-th element is $\log \boldsymbol{\phi}_i^{(k)}$. $\psi(a) = \frac{d}{da} \ln \Gamma(a)$ is the digamma function, and $\mathbf{g}_{r,\frac{L}{2}}$ is the gradient from the sixth term of Equation (2).

2 Bayesian Treatment for RDL

In this section we detail the Bayesian treatment of RDL. The learning process is summarized in Algorithm 1.

As mentioned in the paper, we follow the procedure of variational inference to update the logarithm of variational distributions as the expectation of the joint log-likelihood. 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 $\sigma(\cdot)$). 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 $\mathbf{W}^+$: We denote the vectorization of $\mathbf{W}^+$, $\mathrm{vec}(\mathbf{W}^+)$, as $\mathbf{w} = (\mathbf{w}_e, \mathbf{w}_d)^T$ where $\mathbf{w}_e$ is the collection of weights and biases of the encoder part of the RDL while $\mathbf{w}_d$ is for the decoder part.

For $\mathbf{w}$, we can first write down the terms in $\mathscr{L}$ associated with $\mathbf{w}$:

$$\begin{aligned}\mathscr{L}_{\mathbf{w}} = &- \frac{\lambda_w}{2} \mathbf{w}^T \mathbf{w} - \frac{\lambda_p}{2} \sum_i \|\boldsymbol{\phi}_i - f_e(\mathbf{X}_{0,i*}, \mathbf{w})^T\|_2^2 \\ &- \frac{\lambda_n}{2} \sum_i \|f_r(\mathbf{X}_{0,i*}, \mathbf{w}) - \mathbf{X}_{c,i*}\|_2^2 + const.\end{aligned}$$

Given the hyperparameters, we can find a local maximum of the posterior $\mathbf{w}_{MAP}$ using the back-propagation algorithm. Having found the mode $\mathbf{w}_{MAP}$, we can make a local Gaussian approximation by evaluating the Hessian matrix $\mathbf{A}$ of $-\mathscr{L}_{\mathbf{w}}$: $\mathbf{A} = -\nabla\nabla\mathscr{L}_{\mathbf{w}} = \lambda_w \mathbf{I} + \mathbf{H}$, where $\mathbf{H}$ is the Hessian matrix corresponding to the negation of the last two terms (except the constant term) in $\mathscr{L}_{\mathbf{w}}$. Note that to dramatically speed up training we can approximate the Hessian matrix using diagonal approximation [1] or outer product approximation (Levenberg-Marquardt approximation) [2]. The approximation of the posterior is given by $q(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{w}_{MAP}, \mathbf{A}^{-1})$.

Algorithm 1 Bayesian RDL

Input: corrupted attributes $\mathbf{X}_0$, clean attributes $\mathbf{X}_c$, observed links $\{l_{i,i'}\}_{(i,i')=(1,1)}^{(I,I)}$, number of iterations T, learning rate $\{\rho_t\}_{t=1}^T$, hyperparameters λ_w, λ_p, λ_e, and λ_n
for $t = 1 : T$ **do**
  // For distribution $q(\mathbf{w})$
  Update $\mathbf{w}_{MAP} := \mathbf{w}_{MAP} - \rho_t \nabla_{\mathbf{w}} \mathscr{L}_{\mathbf{w}}$
  Compute the Hessian matrix $\mathbf{H}$
  // For distribution $q(\boldsymbol{\phi}_i)$
  Update $\boldsymbol{\Sigma}_i^{-1} := \mathbf{S}_i^{-1} + \mathbf{S}_i'^{-1}$
  Update $\boldsymbol{\mu}_i := \boldsymbol{\Sigma}_i(\mathbf{S}_i^{-1}\mathbf{m}_i + \mathbf{S}_i'^{-1}\mathbf{m}_i')$
  // For distribution $q(\boldsymbol{\eta})$
  Update $\mathbf{S}_e^{-1} = \lambda_e \mathbf{I}_K + 2\sum_{l_{i,i'}=1} \lambda(\xi_{ii'}) \mathrm{E}((\boldsymbol{\phi}_i \circ \boldsymbol{\phi}_{i'})(\boldsymbol{\phi}_i \circ \boldsymbol{\phi}_{i'})^T)$
  Update $\mathbf{m}_e := \frac{1}{2}\mathbf{S}_e \sum_{l_{i,i'}=1} \mathrm{E}(\boldsymbol{\phi}_i \circ \boldsymbol{\phi}_{i'})$
  // For variational parameters $\xi_{ii'}$
  Update $\xi_{ii'} := \sqrt{(\mathbf{m}_e^T(\boldsymbol{\mu}_i \circ \boldsymbol{\mu}_{i'}))^2 + \sigma_s^2}$

Learning $\{\boldsymbol{\phi}_i\}$: We can write down the terms in $\mathscr{L}$ associated with $\{\boldsymbol{\phi}_i\}$ as:

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

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.

First Gaussian: If we *omit the second term*, given $\mathbf{w}$, 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 $\boldsymbol{\phi}_i$:

$$q_1(\boldsymbol{\phi}_i^{(j)}|\mathbf{X}_{0,i*}) = \int p(\boldsymbol{\phi}_i^{(j)}|\mathbf{X}_{0,i*}, \mathbf{w}_e^{(j)})q(\mathbf{w}_e^{(j)})d\mathbf{w}_e^{(j)}, \tag{4}$$

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 $\boldsymbol{\phi}_i^{(j)}$. Unfortunately the integration is still analytically intractable due to the nonlinearity of $f_e(\mathbf{X}_{0,i*}, \mathbf{w})$ with respect to $\mathbf{w}$. If we assume that $q(\mathbf{w})$ has small variance, a Taylor series expansion of $f_e^{(j)}(\mathbf{X}_{0,i*}, \mathbf{w}_e^{(j)})$ can be made around $\mathbf{w}_{e,MAP}^{(j)}$ where $f_e^{(j)}(\cdot)$ is the j-th element of $f_e(\cdot)$ and $\mathbf{w}_{e,MAP}^{(j)}$ is a sub-vector of $\mathbf{w}_{e,MAP}$ which corresponds to the computation of $\boldsymbol{\phi}_i^{(j)}$:

$$\begin{aligned} f_e^{(j)}(\mathbf{X}_{0,i*}, \mathbf{w}_e^{(j)}) &\approx f_e^{(j)}(\mathbf{X}_{0,i*}, \mathbf{w}_{e,MAP}^{(j)}) + \mathbf{g}_{ij}^T(\mathbf{w}_e^{(j)} - \mathbf{w}_{e,MAP}^{(j)}) \\ \mathbf{g}_{ij} &= \nabla_{w_e^{(j)}} f_e^{(j)}(\mathbf{X}_{0,i*}, \mathbf{w}_e^{(j)})|_{\mathbf{w}_e^{(j)}=\mathbf{w}_{e,MAP}^{(j)}}. \end{aligned}$$

We then have

$$p(\boldsymbol{\phi}_i^{(j)}|\mathbf{X}_{0,i*}, \mathbf{w}_e^{(j)}) \approx \mathcal{N}(\boldsymbol{\phi}_i^{(j)}|f_e^{(j)}(\mathbf{X}_{0,i*}, \mathbf{w}_{e,MAP}^{(j)}) + \mathbf{g}_{ij}^T(\mathbf{w}_e^{(j)} - \mathbf{w}_{e,MAP}^{(j)}), \lambda_p^{-1}).$$

Taking the integration in Equation (4),

$$q_1(\boldsymbol{\phi}_i^{(j)}|\mathbf{X}_{0,i*}) \approx \mathcal{N}(\boldsymbol{\phi}_i^{(j)}|f_e^{(j)}(\mathbf{X}_{0,i*}, \mathbf{w}_{e,MAP}^{(j)}), \lambda_p^{(-1)} + \mathbf{g}_{ij}^T(\mathbf{A}_e^{(j)})^{-1}\mathbf{g}_{ij}),$$

where $\mathbf{A}_e^{(j)}$ is a sub-matrix of $\mathbf{A}$ corresponding 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

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

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

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

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 network.

Second Gaussian: If we *omit the first term* and use the variational lower bound $\sigma(a) \geq \sigma(\xi)\exp\{(a - \xi)/2 - \lambda(\xi)(a^2 - \xi^2)\}$, where $\lambda(\xi) = \frac{1}{2\xi}(\sigma(\xi) - \frac{1}{2})$, we can write $\mathscr{L}_{\{\boldsymbol{\phi}_i\}}$ in Equation (3) as

$$\begin{aligned} \mathscr{L}_{\{\boldsymbol{\phi}_i\}} = &- \frac{\lambda_p}{2}\sum_i \|\boldsymbol{\phi}_i - f_e(\mathbf{X}_{0,i*}, \mathbf{w})^T\|_2^2 + \sum_{l_{i,i'}=1} \{\log\sigma(\xi_{ii'}) + (\boldsymbol{\eta}^T(\boldsymbol{\phi}_i \circ \boldsymbol{\phi}_{i'}) - \xi_{ii'})/2 \\ &- \lambda(\xi_{ii'})((\boldsymbol{\eta}^T(\boldsymbol{\phi}_i \circ \boldsymbol{\phi}_{i'}))^2 - \xi_{ii'}^2)\} + const. \end{aligned} \tag{5}$$

Thus by 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 $q(\boldsymbol{\phi}_{i'}|\mathbf{X}_{0,i'*})$. Thus we have

$$\begin{aligned} \mathbf{m}_i' =& \frac{1}{2}\mathbf{S}_i' \sum_{l_{i,i'}=1} (\mathbf{m}_e \circ \boldsymbol{\mu}_{i'}) \\ \mathbf{S}_i'^{-1} =& 2\sum_{l_{i,i'}=1} \lambda(\xi_{ii'})(\mathbf{S}_e \circ \boldsymbol{\Sigma}_{i'} + (\mathbf{m}_e\mathbf{m}_e^T) \circ \boldsymbol{\Sigma}_{i'} + (\boldsymbol{\mu}_{i'}\boldsymbol{\mu}_{i'}^T) \circ \mathbf{S}_e + (\mathbf{m}_e \circ \boldsymbol{\mu}_{i'})(\mathbf{m}_e \circ \boldsymbol{\mu}_{i'})^T). \end{aligned}$$

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 $\mathbf{S}'_i$.

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 information from the content and the second is relevant to the link information. The final update rule as the product of these two Gaussians then summarizes both information sources and yields more powerful features.

Learning $\boldsymbol{\eta}$: Similar to the second part of learning $\{\boldsymbol{\phi}_i\}$, we can get the update rules for $\boldsymbol{\eta}$:

$$\begin{aligned} q(\boldsymbol{\eta}) &= \mathcal{N}(\boldsymbol{\eta}|\mathbf{m}_e,\mathbf{S}_e) \\ \mathbf{m}_e &= \frac{1}{2}\mathbf{S}_e \sum_{l_{i,i'}=1} \mathrm{E}(\boldsymbol{\phi}_i\circ\boldsymbol{\phi}_{i'}) \\ \mathbf{S}_e^{-1} &= \lambda_e\mathbf{I}_K + 2\sum_{l_{i,i'}=1}\lambda(\xi_{ii'})\mathrm{E}((\boldsymbol{\phi}_i\circ\boldsymbol{\phi}_{i'})(\boldsymbol{\phi}_i\circ\boldsymbol{\phi}_{i'})^T), \end{aligned}$$

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

$$\begin{aligned} \mathbf{m}_e =& \frac{1}{2}\mathbf{S}_e \sum_{l_{i,i'}=1}(\boldsymbol{\mu}_i\circ\boldsymbol{\mu}_{i'}) \\ \mathbf{S}_e^{-1} =& \lambda_e\mathbf{I}_K + 2\sum_{l_{i,i'}=1}\lambda(\xi_{ii'})(\boldsymbol{\Sigma}_i\circ\boldsymbol{\Sigma}_{i'} + (\boldsymbol{\mu}_i\boldsymbol{\mu}_i^T)\circ\boldsymbol{\Sigma}_{i'} + (\boldsymbol{\mu}_{i'}\boldsymbol{\mu}_{i'}^T)\circ\boldsymbol{\Sigma}_i + (\boldsymbol{\mu}_i\circ\boldsymbol{\mu}_{i'})(\boldsymbol{\mu}_i\circ\boldsymbol{\mu}_{i'})^T). \end{aligned}$$

Learning $\xi_{ii'}$: To update $\xi_{ii'}$, we can set the derivative of $\mathrm{E}(\mathscr{L})$ with respect to $\xi_{ii'}$ to zero and get

$$0 = \lambda'(\xi_{ii'})(\mathrm{E}((\boldsymbol{\eta}^T(\boldsymbol{\phi}_i\circ\boldsymbol{\phi}_{i'}))^2) - \xi_{ii'}^2).$$

Since $\lambda'(\xi)$ is a monotonic function of ξ when $\xi\geq 0$ and we set $\xi\geq 0$ without loss of generality due to symmetry of the bound around $\xi=0$, $\lambda'(\mathbf{x})\neq 0$. Hence the square $\xi_{ii'}^2 = \mathrm{E}((\boldsymbol{\eta}^T(\boldsymbol{\phi}_i\circ\boldsymbol{\phi}_{i'}))^2)$, where the expectation is taken over the current $q(\boldsymbol{\phi}_i|\mathbf{X}_{0,i*})$, $q(\boldsymbol{\phi}_{i'}|\mathbf{X}_{0,i'*})$, and $q(\boldsymbol{\eta})$. Thus we have

$$\begin{aligned} \xi_{ii'}^2 &= (\mathbf{m}_e^T(\boldsymbol{\mu}_i\circ\boldsymbol{\mu}_{i'}))^2 + \sigma_s^2 \\ \sigma_s^2 &= tr(\mathbf{S}_e\mathbf{S}_h) + \mathbf{m}_e^T\mathbf{S}_h\mathbf{m}_e + \mathbf{m}_h^T\mathbf{S}_e\mathbf{m}_h \\ \mathbf{m}_h &= \boldsymbol{\mu}_i\circ\boldsymbol{\mu}_{i'} \\ \mathbf{S}_h &= \boldsymbol{\Sigma}_i\circ\boldsymbol{\Sigma}_{i'} + (\boldsymbol{\mu}_i\boldsymbol{\mu}_i^T)\circ\boldsymbol{\Sigma}_{i'} + (\boldsymbol{\mu}_{i'}\boldsymbol{\mu}_{i'}^T)\circ\boldsymbol{\Sigma}_i. \end{aligned}$$

Predicting $\psi(l_{i,i'}=1|\boldsymbol{\phi}_i,\boldsymbol{\phi}_{i'},\boldsymbol{\eta}) = \sigma(\boldsymbol{\eta}^T(\boldsymbol{\phi}_i\circ\boldsymbol{\phi}_{i'}))$: To calculate the probability of the link between item i and item i', $\sigma(\boldsymbol{\eta}^T(\boldsymbol{\phi}_i\circ\boldsymbol{\phi}_{i'}))$, we first approximate $a=\boldsymbol{\eta}^T(\boldsymbol{\phi}_i\circ\boldsymbol{\phi}_{i'})$ using the Gaussian distribution $\mathcal{N}(a|\mu_s,\sigma_s^2)$, where

$$\begin{aligned} \mu_s &= \mathbf{m}_e^T(\boldsymbol{\mu}_i\circ\boldsymbol{\mu}_{i'}) \\ \sigma_s^2 &= tr(\mathbf{S}_e\mathbf{S}_h) + \mathbf{m}_e^T\mathbf{S}_h\mathbf{m}_e + \mathbf{m}_h^T\mathbf{S}_e\mathbf{m}_h \\ \mathbf{m}_h &= \boldsymbol{\mu}_i\circ\boldsymbol{\mu}_{i'} \\ \mathbf{S}_h &= \boldsymbol{\Sigma}_i\circ\boldsymbol{\Sigma}_{i'} + (\boldsymbol{\mu}_i\boldsymbol{\mu}_i^T)\circ\boldsymbol{\Sigma}_{i'} + (\boldsymbol{\mu}_{i'}\boldsymbol{\mu}_{i'}^T)\circ\boldsymbol{\Sigma}_i. \end{aligned}$$

The probability

$$\psi(l_{i,i'}=1|\boldsymbol{\phi}_i,\boldsymbol{\phi}_{i'},\boldsymbol{\eta}) = \int \sigma(a)\mathcal{N}(a|\mu_s,\sigma_s^2)da.$$

Since it cannot be evaluated analytically, we approximate $\sigma(a)$ by the probit function $\Phi(\lambda a) = \int_{-\infty}^{\lambda a}\mathcal{N}(\theta|0,1)d\theta$ and $\lambda^2=\pi/8$. Finally, we can get $\psi(l_{i,i'}=1|\boldsymbol{\phi}_i,\boldsymbol{\phi}_{i'},\boldsymbol{\eta}) = \sigma(\kappa(\sigma_s^2)\mu_s)$ where $\kappa(\sigma_s^2) = (1+\pi\sigma_s^2/8)^{-1/2}$. Since the final prediction takes both the mean and variance into account, the estimation is expected to be more robust.

Table 1: Top 10 link predictions made by gRTM and RDL for two articles from *citeulike-a*.

	Query: Object class recognition by unsupervised scale-invariant learning
gRTM	Layered depth images Using spin images for efficient object recognition in cluttered 3D scenes Snakes: active contour models Visual learning and recognition of 3-D objects from appearance Contextual priming for object detection **Visual categorization with bags of keypoints** Non-parametric model for background subtraction Alignment by maximization of mutual information **Rapid object detection using a boosted cascade of simple features** W4: real-time surveillance of people and their activities
RDL	**Distinctive image features from scale-invariant keypoints** visual learning and recognition of 3-D objects from appearance **Object recognition with features inspired by visual cortex** **Unsupervised learning of models for recognition** **Robust object recognition with cortex-like mechanisms** **Generative versus discriminative methods for object recognition** Using spin images for efficient object recognition in cluttered 3D scenes **Learning generative visual models from few training examples** 3D object modeling and recognition using affine-invariant patches A Bayesian approach to unsupervised one-shot learning of object categories

	Query: SCOP database in 2004: refinements integrate structure and sequence family data
gRTM	Pfam: multiple sequence alignments and HMM-profiles of protein domains **Structure, function and evolution of multidomain proteins** Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB Nature of the protein universe The CATH domain structure database and related resources Phylogenetic classification of short environmental DNA fragments The catalytic site atlas: a resource of catalytic sites and residues identified in enzymes LGA: a method for finding 3D similarities in protein structures Amino acid substitution matrices from protein blocks Multiple protein sequence alignment
RDL	The universal protein resource (UniProt) **E-MSD: an integrated data resource for bioinformatics** Gene3D: comprehensive structural and functional annotation of genomes The universal protein resource (UniProt) in 2010 **Gene3D: modelling protein structure, function and evolution** The universal protein resource (UniProt): an expanding universe of protein information **Pfam: clans, web tools and services** **The Pfam protein families database** The protein data bank **SCOP: a structural classification of proteins database**

3 Case Study

To gain a better insight into the difference between RDL and gRTM, we first look at two example articles from the test set and the top 10 predicted links (citations) for them returned by RDL and gRTM. In Table 1 (articles with titles in bold mean correct predictions), the first example (query) is a computer vision paper with the title 'Object class recognition by unsupervised scale-invariant learning'. As we can see, while gRTM is able to capture the problem 'object class recognition' and suggest links to articles on 'visual categorization' and 'object detection' (which is a problem closely related to 'object class recognition'), it fails to identify the key notions on 'unsupervised learning' and 'scale-invariant learning' of the target article. On the other hand, these notions are successfully captured by RDL to predict links to articles like 'Distinctive image features from scale-invariant keypoints', 'Unsupervised learning of models for recognition', and 'Learning generative visual models from few training examples', aside from 'object class recognition' papers like 'Object recognition with features inspired by visual cortex'. Consequently, gRTM attains a precision of only 20% while RDL is able to significantly boost the performance to achieve a precision of 60%. Another example is a biology and bioinformatics paper with the title 'SCOP database in 2004: refinements integrate structure and sequence family data'. Similarly, gRTM can only recognize that the target paper is on the **structure of proteins** but miss the fact that this paper is mainly on a **bioinformatics database**. Again, RDL is able to recognize that this article is from the community researching on bioinformatics databases and predict the links to several relevant articles on this topic. As a result, the precision for gRTM is only 10% but RDL is able to achieve a much higher precision of 50%. From these examples, we can see that by jointly and deeply modeling the node attributes and link structures, RDL is able to better partition the topic space and community structures of nodes and more accurately pinpoint the target node (article) in the semantic space. In addition, RDL is able to simultaneously capture multiple concepts of interest while gRTM cannot.

4 Hyperparameter Sensitivity

Figure 1 shows the hyperparameter sensitivity of λ_e for different K. Since the hyperparameters are highly correlated to K, we use λ_e/K in the x-axis instead of λ_e for clarity and consistence among different K

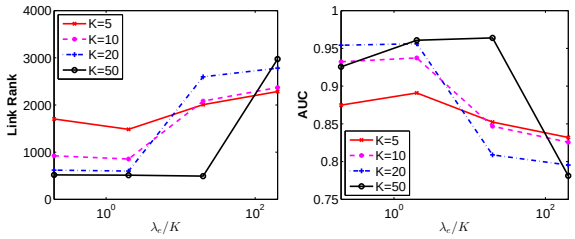

Figure 1: Hyperparameter sensitivity of λ_e for link rank and AUC.

values. As we can see, the performance is not very sensitive to λ_e. RDL can achieve both the lowest link rank and the highest AUC in a relatively wide range of values for λ_e/K for different K. The performance slightly decreases if λ_e/K is lower than the range and dramatically decreases if λ_e/K is higher than the range. Similar phenomena can be observed for other hyperparameters.

References

[1] S. Becker and Y. Le Cun. Improving the convergence of back-propagation learning with second order methods. In *Proceedings of the 1988 connectionist models summer school*, pages 29–37, 1988.

[2] F. D. Foresee and M. T. Hagan. Gauss-Newton approximation to Bayesian learning. In *IJCNN*, volume 3, pages 1930–1935, 1997.