JMLR: Workshop and Conference Proceedings 1:1–9, 2018. Symposium on Advances in Approximate Bayesian Inference

Probabilistic Knowledge Graph Embeddings

Farnood Salehi FARNOOD.SALEHI@EPFL.CH
Robert Bamler ROBERT.BAMLER@GMAIL.COM
Stephan Mandt MANDT@UCI.EDU

Abstract

We develop a probabilistic extension of embedding models for link prediction in relational knowledge graphs. Knowledge graphs are datasets of relational facts among a set of entities, such as people, places, or objects. Even large knowledge graphs typically contain only few facts per entity, motivating a Bayesian treatment. We find that the main benefit of the Bayesian approach is that it allows for efficient and scalable optimization over hyperparameters, which leads to an improvement over the state of the art on several benchmarks.
Keywords: knowledge graphs, variational inference, representation learning

1. Introduction

A knowledge graph is a dataset of relational facts in the form of triplets (*head, relation, tail*), where *head* and *tail* represent entities in the world, e.g., ('Paris', 'is in', 'France'). Empirical knowledge graphs often contain only a small fraction of the enormous number of true relational facts between a large set of entities, and predicting unobserved relational facts ('link prediction') has attracted a lot of attention (Bordes et al., 2013; Nickel et al., 2016b; Trouillon et al., 2016; Xiao et al., 2017; Shi and Weninger, 2017; Lacroix et al., 2018).

State of the art link prediction methods (Trouillon et al., 2016; Lacroix et al., 2018) rely on models that embed each entity and relation into a latent vector space (Nickel et al., 2016a). The performance of such knowledge graph embedding models has been steadily improving over the past five years. However, it was pointed out recently (Kadlec et al., 2017) that the improvement may largely be due to increasingly elaborate hyperparamter tuning rather than due to better models. Using a large scale grid search over hyperparameters, the authors obtained competitive performance even with a very simple model.

In this contribution, we propose an efficient method to learn a macroscopic number of hyperparameters for knowledge graph embedding models. This eliminates the need to reduce the number of hyperparameters by heuristics like frequency dependent regularizers (Lacroix et al., 2018; Srebro and Salakhutdinov, 2010). Our method is based on a Bayesian reinterpretation of knowledge graph embedding models, motivated by the observation that the number of data points *per entity* is typically small, even in large knowledge graphs. The Bayesian formulationa allows us to efficiently optimize over a macroscopic number of hyperparameters using variational expectation maximization (EM) (Bernardo et al., 2003). Using this method, we improve over the state of the art performance on standard benchmarks.

2. Knowledge Graph Embeddings as Generative Models of Facts

In this section, we describe a generative model for relational facts, which is the basis of the hyperparameter optimization method that we propose in Section 3. The generative model

is designed such that a maximum a posteriori (MAP) estimation of its parameters (Bishop, 2006) recovers existing models.

A knowledge graph embedding model over N_e entities and N_r relations assigns embedding vectors E_e and R_r to each entity $e \in [N_\mathrm{e}]$ and each relation $r \in [N_\mathrm{r}]$, respectively. We denote the concatenation of all entity and relation embeddings by $\mathbf{E}$ and $\mathbf{R}$, respectively.

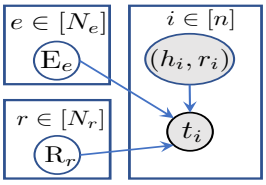


Figure 1: Generative process of triples.

Generative Process. We consider a dataset $\mathbb{S} = \{(h_i, r_i, t_i)\}_{i=1:n}$ of n relational facts with $h_i, t_i \in [N_\mathrm{e}]$ and $r_i \in [N_\mathrm{r}]$. Given some hyperparmeters $\boldsymbol{\lambda}$, we define a joint probability distribution $p(\mathbf{E}, \mathbf{R}, \mathbb{S}|\boldsymbol{\lambda})$ via the following generative process (see Figure 1):

- For each entity $e \in [N_\mathrm{e}]$ (relation $r \in [N_\mathrm{r}]$), draw an embedding vector E_e (similarly R_r) from a prior $p(\mathrm{E}_e|\lambda^\mathrm{E}_e)$ (similarly $p(\mathrm{R}_r|\lambda^\mathrm{R}_r)$), e.g., a normal distribution. Here, λ^E_e and λ^R_r are hyperparameters that depend, in general, on e and r, respectively.
- Repeat for each triplet index $i \in \{1, \ldots, n\}$:
  - Draw a head h_i and a relation r_i from a discrete joint distribution $P(h_i, r_i)$. The choice of probability distribution $P(h_i, r_i)$ has no influence on inference.
  - Draw $t_i \sim \mathrm{Multinomial}(\mathrm{softmax}_t(\mathsf{X}_{h_i,r_i,t}))$ with $\mathrm{softmax}_t(\mathsf{X}_{h_i,r_i,t}) = \frac{\exp \mathsf{X}_{h_i,r_i,t}}{\sum_{t'} \exp \mathsf{X}_{h_i,r_i,t'}}$ where the score $\mathsf{X}_{h,r,t}$ is a model dependent function of E_h, R_r, and E_t, see below.

When we fit the model to an observed dataset we preprocess the dataset as proposed in (Lacroix et al., 2018). For each observed fact (h, r, t) we add a new fact (t, r^{-1}, h) to the dataset, where r^{-1} is a new symbol. This allows us to use the fitted model for both head and tail prediction as it maps a head prediction task $(?, r, t)$ to a tail prediction task $(t, r^{-1}, ?)$.

DistMult and ComplEx model. In our experiments, we use two different models: DistMult (Yang et al., 2015) and ComplEx (Trouillon et al., 2016; Lacroix et al., 2018). DistMult uses a K-dimensional real embedding space $\mathbb{R}^K$ and scores $\mathsf{X}_{h,r,t} = \sum_{k=1}^K \mathrm{E}_{hk}\mathrm{R}_{rk}\mathrm{E}_{tk}$, where k indexes the embedding dimension. ComplEx uses a complex embedding space $\mathbb{C}^K$ and scores $\mathsf{X}_{h,r,t} = \mathrm{Re}\left[\sum_{k=1}^K \mathrm{E}_{hk}\mathrm{R}_{rk}\bar{\mathrm{E}}_{tk}\right]$, where $\mathrm{Re}[\,\cdot\,]$ denotes the real part and $\bar{\mathrm{E}}_{tk}$ is the complex conjugate. The original works learn embeddings $\mathbf{E}$ and $\mathbf{R}$ by minimizing a loss function $L(\mathbf{E}, \mathbf{R}; \boldsymbol{\lambda})$ for some fixed hyperparameters $\boldsymbol{\lambda}$. This is equivalent to MAP estimation of the above generative model since it turns out that, up to terms that depend only on $\boldsymbol{\lambda}$, the used loss function is $L(\mathbf{E}, \mathbf{R}; \boldsymbol{\lambda}) = -\log p(\mathbf{E}, \mathbf{R}, \mathbb{S}|\boldsymbol{\lambda})$.

3. Scalable Hyperparameter Tuning via Variational EM

In this section, we describe an efficient method to learn optimized hyperparameters λ^E_e and λ^R_r for each entity e and each relation r in the type of the models defined in Section 2. In contrast to grid search, our method scales to a macroscopic number of hyperparameters, eliminating the need for heuristics to reduce the number of hyperparameters (Srebro and Salakhutdinov, 2010) as done in existing works, e.g., (Lacroix et al., 2018).

Algorithm 1 summarizes our method. It consists of three stages. We first pre-train the model using fixed initial hyperparameters $\boldsymbol{\lambda}$ (line 2). As detailed below, we then optimize

Algorithm 1: Hyperparameter learning in knowledge graph embedding models.

Input: Number of training steps T and equilibration steps T_0; loss function L; learning rate $\eta > 0$.

1 Initialize $\boldsymbol{\lambda}$ and $\boldsymbol{\gamma} \equiv (\boldsymbol{\mu}^{\mathrm{E/R}}, \boldsymbol{\sigma}^{\mathrm{E/R}})$
2 Pre-train the model using SGD: $\boldsymbol{\mu}^{\mathrm{E}}, \boldsymbol{\mu}^{\mathrm{R}} \leftarrow \arg\min_{\mathbf{E},\mathbf{R}} L(\mathbf{E}, \mathbf{R}; \boldsymbol{\lambda})$
3 **for** $t \leftarrow 1$ **to** T **do**
4 draw mini-batch $\mathcal{B} \in \mathbb{S}$
5 draw uniform noise samples, $\boldsymbol{\epsilon}_e^{\mathrm{E}} \sim N(0, I)$ and $\boldsymbol{\epsilon}_r^{\mathrm{R}} \sim N(0, I)$
6 $\boldsymbol{\gamma} \leftarrow \boldsymbol{\gamma} - \eta \nabla_{\boldsymbol{\gamma}} \Big[L\big(\boldsymbol{\mu}^{\mathrm{E}} + \boldsymbol{\epsilon}^{\mathrm{E}} \boldsymbol{\sigma}^{\mathrm{E}}, \boldsymbol{\mu}^{\mathrm{R}} + \boldsymbol{\epsilon}^{\mathrm{R}} \boldsymbol{\sigma}^{\mathrm{R}}; \boldsymbol{\lambda}\big) - \sum_{e,k} \log \sigma_{ek}^{\mathrm{E}} - \sum_{r,k} \log \sigma_{rk}^{\mathrm{R}} \Big]$
7 **if** $t > T_0$ **then** $\boldsymbol{\lambda} \leftarrow \arg\min_{\boldsymbol{\lambda}} \mathbb{E}_{q_{\boldsymbol{\gamma}}}\big[p\,(\mathbf{E}, \mathbf{R} | \boldsymbol{\lambda})\big]$ ▷ See appendix for analytic solution.
8 **else** do not update hyperparameters $\boldsymbol{\lambda}$
9 **end**
10 Re-train model with learned hyperparameters: $\mathbf{E}^*, \mathbf{R}^* \leftarrow \arg\min_{\mathbf{E},\mathbf{R}} L(\mathbf{E}, \mathbf{R}; \boldsymbol{\lambda})$

over $\boldsymbol{\lambda}$ using variational expectation maximization (lines 3-9), initializing around the pre-trained model. Finally, we re-train the model using the learned hyperparameters (line 10).

Variational Expectation Maximization. The optimization over hyperaparameters $\boldsymbol{\lambda}$ in lines 3-9 of Algorithm 1 is based on variational expectation maximization (variational EM) (Bernardo et al., 2003). The EM algorithm (Dempster et al., 1977) maximizes the marginal likelihood $p(\mathbb{S}|\boldsymbol{\lambda}) = \int p(\mathbf{E}, \mathbf{R}, \mathbb{S}|\boldsymbol{\lambda})\, d\mathbf{E}\, d\mathbf{R}$ over $\boldsymbol{\lambda}$. This is difficult because of the intractable integral. Variational EM avoids the integration by maximizing instead a lower bound on $p(\mathbb{S}|\boldsymbol{\lambda})$ using variational inference (VI) (Jordan et al., 1999). In VI, one first chooses a family of variational distributions $q_{\boldsymbol{\gamma}}(\mathbf{E}, \mathbf{R})$, which is parameterized by so-called variational parameters $\boldsymbol{\gamma}$. Evoking Jensen's inequality, the marginal likelihood is then lower-bounded by the *evidence lower bound* (Blei et al., 2017; Zhang et al., 2017), or ELBO, as

$$\log p(\mathbb{S}|\boldsymbol{\lambda}) \geq \mathbb{E}_{\mathbf{E},\mathbf{R} \sim q_{\boldsymbol{\gamma}}} \left[\log p(\mathbf{E}, \mathbf{R}, \mathbb{S}|\boldsymbol{\lambda}) - \log q_{\boldsymbol{\gamma}}(\mathbf{E}, \mathbf{R}) \right] =: \mathrm{ELBO}(\boldsymbol{\gamma}, \boldsymbol{\lambda}). \tag{1}$$

We maximize the ELBO over both $\boldsymbol{\gamma}$ and $\boldsymbol{\lambda}$. We choose a fully factorized Gaussian variational distribution $q_{\boldsymbol{\gamma}}(\mathbf{E}, \mathbf{R})$. The variational parameters $\boldsymbol{\gamma}$ are the means μ_{ek}^{E} and μ_{rk}^{R} of E_{ek} and R_{rk}, respectively, and the standard deviations σ_{ek}^{E} and σ_{rk}^{R}. We initialize the means with pre-trained model parameters (line 2), and the standard deviations with the value 0.2.

The variational EM algorithm can easily be integrated into an existing implementation of a model. We update the variational parameters $\boldsymbol{\gamma}$ using Black Box Variational Inference (BBVI) with reparameterization gradients (Kingma and Welling, 2014; Rezende et al., 2014). In practice, this just means that we keep all $\boldsymbol{\lambda}$ dependent terms of the loss function $L(\mathbf{E}, \mathbf{R}; \boldsymbol{\lambda}) = -\log p(\mathbf{E}, \mathbf{R}, \mathbb{S}|\boldsymbol{\lambda})$, we inject random noise (line 5) into L, and we learn the optimal amount of noise by minimizing $L - \sum_{e/r,k} \log \sigma_{e/r,k}^{\mathrm{E/R}}$ (line 6).

We update the hyperparameters $\boldsymbol{\lambda}$ using an analytic solution of the minimization over $\boldsymbol{\lambda}$, see appendix. Since the variational distribution may initially be far from the true posterior, we begin updating $\boldsymbol{\lambda}$ only after an initial equilibration phase of T_0 steps (lines 7-8).

Table 1: Model performances (filtered); *(Trouillon et al., 2016); ‡(Kadlec et al., 2017); †our reimplementation of ComplEx and DistMult with reciprocal relations, weighted 3-norm regularizer, and hyperparameters for ComplEx from (Lacroix et al., 2018).

	dataset →	WN18RR	WN18	FB15K237	FB15K
	↓ model	MRR/MRR$_b$/H@10	"/"/"	"/"/"	"/"/"
ComplEx	Literature*	-/-/-	94.1/-/94.7	-/-/-	69.2/-/84.0
	Our MAP†	48.2/47.0/57.2	95.2/**95.8**/96.2	36.4/19.9/55.6	**85.8/82.0**/90.9
	Our EM	**48.6/47.3/57.9**	**95.3/95.8/96.4**	**36.5/20.3/56.0**	85.4/81.9/**91.5**
DistMult	Literature‡	-/-/-	79.0/-/95.0	-/-/-	83.7/-/90.4
	Our MAP†	44.7/43.4/53.3	89.8/90.0/95.8	35.5/18.9/54.6	**84.2**/80.0/90.8
	Our EM	**45.5/44.1/54.4**	**91.1/91.1/96.1**	**35.7/19.4/54.8**	84.1/**80.2/91.4**

4. Experimental Results

We compare the performance of the proposed variational EM algorithm to results from the literature for two models on four standard data sets. We consider two standard metrics: hits at 10 (H@10) and mean reciprocal rank, MRR $:= \frac{100}{|\mathbb{S}'|} \sum_{(h,r,t)\in\mathbb{S}'} 1/\text{rank}(t|h,r)$. Here, $\mathbb{S}'$ (with cardinality $|\mathbb{S}'|$) is the test set (with reciprocal relations) and $\text{rank}(t|h,r)$ is the (filtered, see (Bordes et al., 2013)) rank of the correct tail t in the tail prediction task $(h,r,?)$. We also introduce a more balanced metric that averages over targets t instead of test points, $\text{MRR}_b := \frac{100}{N_\text{e}} \sum_{t=1}^{N_\text{e}} \frac{1}{|\mathbb{S}'_t|} \left[\sum_{(h,r,t)\in\mathbb{S}'_t} 1/\text{rank}(t|h,r)\right]$, where $\mathbb{S}'_t$ is the set of test facts with tail t.

Table 1 summarizes our results. Higher values are better for all metrics. Variational EM (last row in each group) improves the performance of ComplEx on three out of the four datasets. The improvements are even more pronounced for DistMult. MRR_b is generally lower than its unbalanced counterpart MRR as the latter puts more weight on easy tasks.

5. Conclusions

We showed that the popular knowledge graph embedding models DistMult and ComplEx have an interpretation as probabilistic generative models. Drawing on this view, we presented a scalable method to optimize over hyperparameters using variational expectation maximization. Our method outperformed the state of the art in link prediction on several popular benchmark datasets. Due to the simplicity and generality of our method, we think that it lends itself to hyperparameter tuning in future knowledge graph embedding models.

Our approach amounted to training the model twice: once in a Bayesian fashion to optimize hyperparameters, and a second time by point-estimating model parameters using the optimized hyperparameters. One may wonder why we did not directly use the approximate posterior for link prediction. We found empirically that link prediction with the approximate posterior improved performance in low embedding dimensions, but underperformed in high dimensions (not shown). This may be due to a failure of the variational approximation in high dimensions, where the true posterior may not be locally Gaussian. For future work, we suggest to explore Bayesian link prediction using a structured variational distribution.

References

JM Bernardo, MJ Bayarri, JO Berger, AP Dawid, D Heckerman, AFM Smith, M West, et al. The variational Bayesian EM algorithm for incomplete data: with application to scoring graphical model structures. *Bayesian statistics*, 7:453–464, 2003.

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (methodological)*, pages 1–38, 1977.

Frank L Hitchcock. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6(1-4):164–189, 1927.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

Rudolf Kadlec, Ondrej Bajgar, and Jan Kleindienst. Knowledge base completion: Baselines strike back. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 69–74. Association for Computational Linguistics, 2017.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. Canonical tensor decomposition for knowledge base completion. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2016a.

Maximilian Nickel, Lorenzo Rosasco, Tomaso A Poggio, et al. Holographic embeddings of knowledge graphs. In *AAAI*, volume 2, pages 3–2, 2016b.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

Baoxu Shi and Tim Weninger. Proje: Embedding projection for knowledge graph completion. In *AAAI*, volume 17, pages 1236–1242, 2017.

Nathan Srebro and Ruslan R Salakhutdinov. Collaborative filtering in a non-uniform world: Learning with the weighted trace norm. In *Advances in Neural Information Processing Systems*, pages 2056–2064, 2010.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 2071–2080, 2016.

Han Xiao, Minlie Huang, Lian Meng, and Xiaoyan Zhu. Ssp: Semantic space projection for knowledge graph embedding with text descriptions. In *AAAI*, volume 17, pages 3104–3110, 2017.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *International Conference on Learning Representations (ICLR)*, 2015.

Cheng Zhang, Judith Butepage, Hedvig Kjellstrom, and Stephan Mandt. Advances in variational inference. *arXiv preprint arXiv:1711.05597*, 2017.

Appendix A. Computing the Terms in ELBO

In this section , we compute the expectations appear in the ELBO in Eq. 1. For simplicity, we choose DistMult as the embedding model, but the computations are similar for the other models such as ComplEx (Trouillon et al., 2016) or CP (Hitchcock, 1927). The term $-\mathbb{E}_{q_{\boldsymbol{\gamma}}}[\log q_{\boldsymbol{\gamma}}(\mathbf{E},\mathbf{R})]$ is the entropy of the Gaussian distribution, and it is

$$-\mathbb{E}_{q_{\boldsymbol{\gamma}}}[\log q_{\boldsymbol{\gamma}}(\mathbf{E},\mathbf{R})] = \sum_{e\in[N_e]}\sum_{k\in[K]} \log \sigma^{\mathrm{E}}_{ek} + \sum_{r\in[N_r]}\sum_{k\in[K]} \log \sigma^{\mathrm{R}}_{rk} + c, \tag{2}$$

where c is a constant and can be ignored in the optimization process. Now, let us focus on the evaluation of the expected prior. We derive the equations for $\mathbb{E}_{q_{\boldsymbol{\gamma}}}\left[\log p(\mathbf{E}|\boldsymbol{\lambda}^{\mathrm{E}})\right]$ and $\mathbb{E}_{q_{\boldsymbol{\gamma}}}\left[\log p(\mathbf{R}|\boldsymbol{\lambda}^{\mathrm{R}}))\right]$ can be computed in a similar way.

Gaussian prior. First, let us focus on the Gaussian prior

$$p(\mathrm{E}_e|\lambda^{\mathrm{E}}_e) = \sqrt{(\frac{\lambda^{\mathrm{E}}_e}{2\pi})^K}\exp(-\lambda^{\mathrm{E}}_e\|\mathrm{E}_e\|^2/2).$$

The log of Gaussian prior is

$$\log p(\mathrm{E}_e|\lambda^{\mathrm{E}}_e) = -\frac{\lambda^{\mathrm{E}}_e}{2}\|\mathrm{E}_e\|^2 + \frac{K}{2}\log\lambda^{\mathrm{E}}_e + c. \tag{3}$$

The expectation of Eq. 3 with respect to the variational distribution $q_{\boldsymbol{\gamma}}$ is

$$\mathbb{E}_{q_{\boldsymbol{\gamma}}}\left[\log p(\mathrm{E}_e|\lambda^{\mathrm{E}}_e)\right] = -\frac{\lambda^{\mathrm{E}}_e}{2}\left(\|\mu^{\mathrm{E}}_e\|^2 + \|\sigma^{\mathrm{E}}_e\|^2\right) + \frac{K}{2}\log\lambda^{\mathrm{E}}_e + c., \tag{4}$$

where $\|\sigma^{\mathrm{E}}_e\|^2 = \sum_{k\in[K]}(\sigma^{\mathrm{E}}_{ek})^2$, (details can be found in Appendix B of (Kingma and Welling, 2014)). Notice that λ^{E}_e only appears in $p(\mathrm{E}_e|\lambda^{\mathrm{E}}_e)$, so for a given μ^{E}_e and σ^{E}_e we can easily find its optimal value by maximizing $\mathbb{E}_{q_{\boldsymbol{\gamma}}}\left[\log p(\mathrm{E}_e|\lambda^{\mathrm{E}}_e)\right]$ over λ^{E}_e as follows,

$$\lambda^{\mathrm{E}}_e = \frac{K}{\|\mu^{\mathrm{E}}_e\|^2 + \|\sigma^{\mathrm{E}}_e\|^2}, \tag{5}$$

Nuclear 3-norm prior. Now, let us consider the prior

$$p(\mathrm{E}_e|\lambda^{\mathrm{E}}_e) = \frac{1}{Z}\exp(-\frac{\lambda^{\mathrm{E}}_e}{3}\|\mathrm{E}_e\|^3_3),$$

where Z is the normalization factor. The normalization factor Z is only a function of λ^{E}_e and not the variational parameters $\boldsymbol{\gamma}$, hence $\mathbb{E}_{q_{\boldsymbol{\gamma}}}[\log Z] = \log Z$. We start the analysis by computing the normalization factor Z,

$$Z = \int_{\mathrm{E}_e\in\mathbb{R}^K}\exp(-\frac{\lambda^{\mathrm{E}}_e}{3}\|\mathrm{E}_e\|^3_3)\ d\mathrm{E}_e, \tag{6}$$

we compute the integral with the change of the variables $\zeta = (\lambda^{\mathrm{E}}_e)^{1/3}\mathrm{E}_e$

$$Z = \frac{1}{(\lambda^{\mathrm{E}}_e)^{K/3}}\int_{\zeta\in\mathbb{R}^K}\exp(-\frac{1}{3}\|\zeta\|^3_3)\ d\zeta = \frac{c}{(\lambda^{\mathrm{E}}_e)^{K/3}}, \tag{7}$$

where $c = \int_{\zeta \in \mathbb{R}^K} \exp(-\frac{1}{3}\|\zeta\|_3^3)\ d\zeta$. Therefore,

$$\mathbb{E}_{q_{\boldsymbol{\gamma}}}[\log Z] = \log Z = -\frac{K}{3}\log \lambda_e^{\mathrm{E}} + c \tag{8}$$

Now, let us focus on computing

$$\mathbb{E}_{q_{\boldsymbol{\gamma}}}\left[\frac{\lambda_e^{\mathrm{E}}}{3}\|\mathrm{E}_e\|_3^3\right] = \mathbb{E}_{q_{\boldsymbol{\gamma}}}\left[\frac{\lambda_e^{\mathrm{E}}}{3}\sum_{k\in[K]}|\mathrm{E}_{ek}|^3\right].$$

Given the Gaussian variational distribution $N(\mu_{ek}^{\mathrm{E}}, (\sigma_{ek}^{\mathrm{E}})^2)$, the samples of $N(\mu_{ek}^{\mathrm{E}}, (\sigma_{ek}^{\mathrm{E}})^2)$ can be written as

$$\mathrm{E}_{ek} \sim \mu_{ek}^{\mathrm{E}} + \sigma_{ek}^{\mathrm{E}}\epsilon, \tag{9}$$

where $\epsilon \in \mathbb{R}$ is the noise sampled from the Gaussian distribution $N(0,1)$. Using this reparametrization, we get

$$\begin{aligned}
\mathbb{E}_{q_{\boldsymbol{\gamma}}}\left[|\mathrm{E}_{ek}|^3\right] &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}|\mu_{ek}^{\mathrm{E}} + \sigma_{ek}^{\mathrm{E}}x|^3 e^{-\frac{1}{2}x^2}\ dx \\
&\overset{\mu_0=\frac{\mu_{ek}^{\mathrm{E}}}{\sigma_{ek}^{\mathrm{E}}}}{=} \frac{(\sigma_{ek}^{\mathrm{E}})^3}{\sqrt{2\pi}}\int_{-\infty}^{\infty}|\mu_0 + x|^3 e^{-\frac{1}{2}x^2}\ dx \\
&= \frac{(\sigma_{ek}^{\mathrm{E}})^3}{\sqrt{2\pi}}\left[-\int_{\mu_0}^{\infty}(-\mu_0+x)^3 e^{-\frac{1}{2}x^2}\ dx + \int_{-\mu_0}^{\infty}(\mu_0+x)^3 e^{-\frac{1}{2}x^2}\ dx\right] \\
&= \frac{(\sigma_{ek}^{\mathrm{E}})^3}{\sqrt{2\pi}}\left[2\int_{\mu_0}^{\infty}(3\mu_0^2 x + x^3)e^{-\frac{1}{2}x^2}\ dx + \int_{-\mu_0}^{\mu_0}(3\mu_0 x^2 + \mu_0^3)e^{-\frac{1}{2}x^2}\ dx\right] \\
&= \frac{(\sigma_{ek}^{\mathrm{E}})^3}{\sqrt{2\pi}}\left[2(\mu_0^2+2)e^{-\frac{1}{2}\mu_0^2} + \sqrt{\frac{\pi}{2}}\left(6\mu_0 + 2\mu_0^3\right)\mathrm{erf}(\frac{\mu_0}{\sqrt{2}})\right] \\
&= \sqrt{\frac{2}{\pi}}\left((\mu_{ek}^{\mathrm{E}})^2\sigma_{ek}^{\mathrm{E}} + 2(\sigma_{ek}^{\mathrm{E}})^3\right)e^{-\frac{1}{2}\left(\frac{\mu_{ek}^{\mathrm{E}}}{\sigma_{ek}^{\mathrm{E}}}\right)^2} + \left(3|\mu_{ek}^{\mathrm{E}}|(\sigma_{ek}^{\mathrm{E}})^2 + |\mu_{ek}^{\mathrm{E}}|^3\right)\mathrm{erf}\left(\frac{|\mu_{ek}^{\mathrm{E}}|}{\sqrt{2}\sigma_{ek}^{\mathrm{E}}}\right),
\end{aligned} \tag{10}$$

where $\mathrm{erf}(\cdot)$ is the error function. Given $\mathbb{E}_{q_{\boldsymbol{\gamma}}}\left[\|\mathrm{E}_e\|_3^3\right]$ we can compute the optimal λ_e^{E} for a given $\boldsymbol{\gamma}$ by maximizing ELBO as

$$\lambda_e^{\mathrm{E}} = \frac{K}{\mathbb{E}_{q_{\boldsymbol{\gamma}}}\left[\|\mathrm{E}_e\|_3^3\right]}. \tag{11}$$

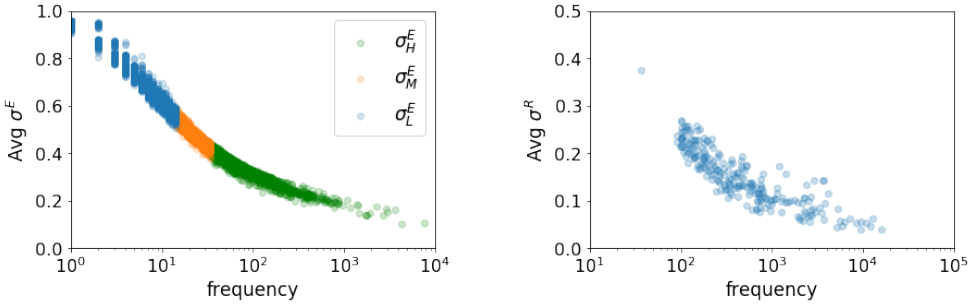
Arbitrary prior. For an arbitrary prior, we can always use the reparameterization trick to compute a stochastic estimate of log prior and its gradient (check Lines 4-6 in Algorithm 1).

Table 2: Dataset Statistics.

Dataset	#entities	#relations	#training$\times 10^3$	#test$\times 10^3$	# validation$\times 10^3$
FB15K237	15k	237	272k	20k	18k
FB15K	15K	1k	500k	60k	50k
WN18RR	41k	11	87k	3k	3k
WN18	41k	18	141k	5k	5k

Appendix B. Uncertainty Analysis

In this section, we show that, on average, the frequent entities or relations have a lower uncertainty and variational EM algorithm learns more confident embeddings for the frequent entities or relations. Figure 2 depicts the average standard deviation (i.e., $\boldsymbol{\sigma}^{\mathrm{E}}$ and $\boldsymbol{\sigma}^{\mathrm{R}}$) of the embeddings inferred by variational EM algorithm. We see that as the frequency of appearance of an entity or a relation increases, the variance of the variational distribution for those entities or relations decreases. Also, on average, the relations' embeddings have lower uncertainty compared to the embeddings' uncertainty. This is because there are fewer relations compared to entities, hence on average, there are more facts for a relation compared to an entity.



(a) Uncertainty of entities' embeddings vs frequencies.

(b) Uncertainty of relations' embeddings vs frequencies.

Figure 2: The average of standard deviations of Gaussian variational distribution for FB15K237 in variational EM algorithm. We see that on average the entities with higher frequencies reach a lower σ, that means a lower uncertainty.