NEURAL VARIATIONAL INFERENCE FOR EMBEDDING KNOWLEDGE GRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in Neural Variational Inference allowed for a renaissance in latent variable models in a variety of domains involving high-dimensional data. In this paper, we introduce two generic Variational Inference frameworks for generative models of Knowledge Graphs; Latent Fact Model and Latent Information Model. While traditional variational methods derive an analytical approximation for the intractable distribution over the latent variables, here we construct an inference network conditioned on the symbolic representation of entities and relation types in the Knowledge Graph, to provide the variational distributions. The new framework can create models able to discover underlying probabilistic semantics for the symbolic representation by utilising parameterisable distributions which permit training by back-propagation in the context of neural variational inference, resulting in a highly-scalable method. Under a Bernoulli sampling framework, we provide an alternative justification for commonly used techniques in large-scale stochastic variational inference, which drastically reduces training time at a cost of an additional approximation to the variational lower bound. The generative frameworks are flexible enough to allow training under any prior distribution that permits a reparametrisation trick, as well as under any scoring function that permits maximum likelihood estimation of the parameters. Experiment results display the potential and efficiency of this framework by improving upon multiple benchmarks with Gaussian prior representations. Code publicly available on Github additionally allows learning Hyperspherical representations under a von-Mises Fisher prior distribution.

1 INTRODUCTION

In many fields, including physics and biology, being able of representing *uncertainty* is of crucial importance (Ghahramani, 2015). For instance, link prediction in Knowledge Graphs is used for driving expensive pharmaceutical experiments (Bean et al., 2017). It would be beneficial to know what is the confidence of a model in its predictions. However, a significant shortcoming of current neural link prediction models – and for the vast majority of neural representation learning approaches – is their inability to express a notion of uncertainty.

In particular, neural link prediction models usually only return point estimates of parameters and predictions (Nickel et al., 2016), and are trained *discriminatively* rather than *generatively*: they aim at predicting one variable of interest conditioned on all the others, rather than accurately representing the relationships between different variables (Ng & Jordan, 2001). This is an important issue when applying representation learning models to Knowledge Graphs: such graphs often suffer from incompleteness and sparsity (Dong et al., 2014), and it is fundamental to know the uncertainty or variance associated with a prediction.

Furthermore, Knowledge Graphs can be very large and Web-scale (Dong et al., 2014). In a probabilistic model, we can leverage the variance in model parameters and predictions for finding which facts to sample during training, in an Active Learning setting (Kapoor et al., 2007; Gal et al., 2017).

2 BACKGROUND

In this work, we focus on models for *predicting missing links* in large, multi-relational networks such as FREEBASE. In the literature, this problem is referred to as *link prediction*. We specifically focus on *knowledge graphs*, i.e., graph-structured knowledge bases where factual information is stored in the form of relationships between entities. Link prediction in knowledge graphs is also known as *knowledge base population*. We refer to Nickel et al. (2016) for a recent survey on approaches to this problem.

A knowledge graph $\mathcal{G} \triangleq \{(r, a_1, a_2)\} \subseteq \mathcal{R} \times \mathcal{E} \times \mathcal{E}$ can be formalised as a set of triples (facts) consisting of a relation type $r \in \mathcal{R}$ and two entities $a_1, a_2 \in \mathcal{E}$, respectively referred to as the *subject* and the *object* of the triple. Each triple (r, a_1, a_2) encodes a relationship of type r between a_1 and a_2, represented by the fact $r(a_1, a_2)$.

Link prediction in knowledge graphs is often simplified to a *learning to rank* problem, where the objective is to find a score or ranking function $\phi_r^{\Theta} : \mathcal{E} \times \mathcal{E} \mapsto \mathbb{R}$ for a relation r that can be used for ranking triples according to the likelihood that the corresponding facts hold true.

2.1 NEURAL LINK PREDICTION

Recently, a specific class of link predictors received a growing interest (Nickel et al., 2016). These predictors can be understood as multi-layer neural networks. Given a triple $\mathbf{x} = (s, r, o)$, the associated score $\phi_r^{\Theta}(s, o)$ is given by a neural network architecture encompassing an *encoding layer* and a *scoring layer*.

In the encoding layer, the subject and object entities s and o are mapped to low-dimensional vector representations (embeddings) $\mathbf{h}_s \triangleq \mathbf{h}(s) \in \mathbb{R}^k$ and $\mathbf{h}_o \triangleq \mathbf{h}(o) \in \mathbb{R}^k$, produced by an encoder $\mathbf{h}^{\Gamma} : \mathcal{E} \rightarrow \mathbb{R}^k$ with parameters Γ. This layer can be pre-trained (Vylomova et al., 2016) or, more commonly, learnt from data by back-propagating the link prediction error to the encoding layer (Bordes et al., 2013; Nickel et al., 2016; Trouillon et al., 2016a).

In the scoring layer, the entity representations $\mathbf{h}_s$ and $\mathbf{h}_o$ are scored by a function $\phi_r^{\Theta}(\mathbf{h}_s, \mathbf{h}_o)$, parametrised by Θ.

Summarising, the high-level architecture is defined as:

$$\begin{aligned} \mathbf{h}_s, \mathbf{h}_o &\triangleq \mathbf{h}^{\Gamma}(s), \mathbf{h}^{\Gamma}(o) \\ \phi_r(s, o) &\triangleq \phi_r^{\Theta}(\mathbf{h}_s, \mathbf{h}_o) \end{aligned}$$

Ideally, more likely triples should be associated with higher scores, while less likely triples should be associated with lower scores.

While the literature has produced a multitude of encoding and scoring strategies, for brevity we overview only a small subset of these. However, we point out that our method makes no further assumptions about the network architecture other than the existence of an argument encoding layer.

2.2 ENCODING LAYER

Given an entity $e \in \mathcal{E}$, the entity encoder $\mathbf{h}^{\Gamma}$ is usually implemented as a simple embedding layer $\mathbf{h}^{\Gamma}(e) \triangleq [\Gamma]_e$, where Γ is an embedding matrix (Nickel et al., 2016). For pre-trained embeddings, the embedding matrix is fixed. Note that other encoding mechanisms are conceivable, such as; recurrent, graph convolution (Kipf & Welling, 2016a;b) or convolutional neural networks (Dettmers et al., 2017).

2.3 DECODING LAYER: SCORING FUNCTIONS

DistMult DISTMULT (Yang et al., 2015) represents each relation r using a parameter vector $\Theta_r \in \mathbb{R}^k$, and scores a link of type r between $(\mathbf{h}_s, \mathbf{h}_o)$ using the following scoring function:

$$\phi_r^{\Theta}(\mathbf{h}_s, \mathbf{h}_o) \triangleq \langle \Theta_r, \mathbf{h}_s, \mathbf{h}_o \rangle \triangleq \sum_{i=1}^{k} \Theta_{r,i} \mathbf{h}_{s,i} \mathbf{h}_{o,i},$$

where $\langle \cdot, \cdot, \cdot \rangle$ denotes the tri-linear dot product.

ComplEx COMPLEX (Trouillon et al., 2016a) is an extension of DISTMULT using complex-valued embeddings while retaining the mathematical definition of the dot product. In this model, the scoring function is defined as follows:

$$\phi_r^{\Theta}(\mathbf{h}_s, \mathbf{h}_o) \triangleq \text{Re}\left(\langle \Theta_r, \mathbf{h}_s, \overline{\mathbf{h}_o} \rangle\right),$$

where $\Theta_r, \mathbf{h}_s, \mathbf{h}_o \in \mathbb{C}^k$ are complex vectors, $\overline{\mathbf{x}}$ denotes the complex conjugate of $\mathbf{x}$, and $\text{Re}\,(\mathbf{x}) \in \mathbb{R}^k$ denotes the real part of $\mathbf{x}$.

3 RELATED WORK

Variational Deep Learning has seen great success in areas such as parametric/non-parametric document modelling Miao et al. (2017); Miao et al. (2016) and image generation (Kingma & Welling (2013)). Stochastic variational inference has been used to learn probability distributions over model weights (Blundell et al., 2015), which the authors named "Bayes By Backprop", as well as proven powerful enough to train deep belief networks (Vilnis & McCallum, 2014), by improving upon the stochastic variational bayes estimator (Kingma & Welling, 2013), using general variance reduction techniques.

Previous work has been done to re-frame word embeddings in a Bayesian framework (Zhang et al., 2014; Vilnis & McCallum, 2014), as well as re-frame graph embeddings in a Bayesian framework (He et al., 2015). However, these methods are expensive to train due to the evaluation of complex tensor inversions. Recent work by the authors of (Barkan, 2016; Bražinskas et al., 2017) show that it is possible to train word embeddings through a VB (Bishop, 2006) framework.

KG2E (He et al., 2015) proposed a probabilistic embedding method for modelling the uncertainties in KGs. However, this was not a generative model. The authors of (Xiao et al., 2016) argue they created the first generative model for knowledge graph embeddings. Firstly, this work is empirically worse than a few of the generative models built under our proposed framework. Secondly, their method is restricted to a Gaussian distribution prior, whereas we can use this, as well as any other prior that permits a re-parameterisation trick — such as the von-Mises distribution.

Later, the authors of (Kipf & Welling, 2016b) propose a generative model for graph embeddings. However, their method lacks scalability as it requires the use of the full adjacency tensor of the graph as input. Secondly, our work differs from (Kipf & Welling, 2016b) as they work with uni-relational data, whereas we create a framework for many variational generative models over multi-relational data.

Recent work by the authors of (Chen et al., 2018) led to successfully constructing a variational path ranking algorithm, a graph feature model. This work differs from ours for two reasons. Firstly, it does not produce a generative model for knowledge graph embeddings. Secondly, their work is a graph feature model, with the constraint of at most one relation per entity pair, whereas our model is a latent feature model with a theoretical unconstrained limit on the number of existing relationships between a given pair of entities.

4 GENERATIVE MODELS

Let $\mathcal{D} \triangleq \{(\tau_1, y_1), \ldots, (\tau_n, y_n)\}$ denote a set of labelled triples, where $\tau_i \triangleq \langle \mathrm{s}_i, \mathrm{p}_i, \mathrm{o}_i \rangle$, and $y_i \in \{0, 1\}$ denotes the corresponding label, denoting that the fact encoded by the triple is either *true* or *false*. We can assume $\mathcal{D}$ is generated by a corresponding *generative model*. In the following, we propose two alternative generative models.

4.1 LATENT FACT MODEL

In this model, we assume that the Knowledge Graph was generated according to the following generative model. Let $\mathcal{V} \triangleq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ the space of possible triples. where $\tau \triangleq \langle \mathrm{s}, \mathrm{p}, \mathrm{o} \rangle$, and $\mathbf{h}_\tau \triangleq [\mathbf{h}_\mathrm{s}, \mathbf{h}_\mathrm{p}, \mathbf{h}_\mathrm{o}]$ denotes the sampled embedding representations of $\mathrm{s}, \mathrm{o} \in \mathcal{E}$ and $\mathrm{p} \in \mathcal{R}$.

Note that, in this model, the embeddings are sampled for each triple. As a consequence, the set of latent variables in this model is $\mathcal{H} \triangleq \{\mathbf{h}_\tau \mid \tau \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}\}$.

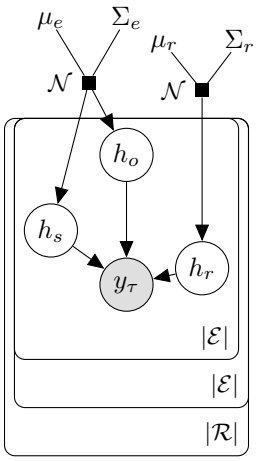

Figure 1: Latent Fact Model (LFM)

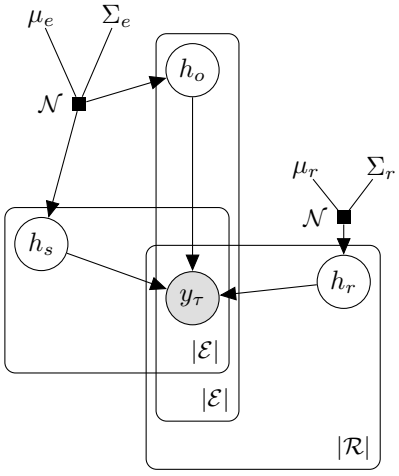

Figure 2: Latent Information Model (LIM)

The joint probability of the variables $p(\mathcal{H}, \mathcal{D})$ is defined as follows:

$$p(\mathcal{H}, \mathcal{D}) \triangleq \prod_{(\tau, y_\tau) \in \mathcal{D}} p(\mathbf{h}_\tau) p(y_\tau \mid \mathbf{h}_\tau) \tag{1}$$

The marginal distribution over $\mathcal{D}$ is then defined as follows:

$$p(\mathcal{D}) \geq \mathbb{E}_q \left[\log p(y_\tau \mid \mathbf{h}_\tau) \right] - D_{\mathrm{KL}} q(\mathbf{h}_\tau) p(\mathbf{h}_\tau) \tag{2}$$

As a consequence, the log-marginal likelihood of the data is bounded by:

$$\log p(\mathcal{D}) \leq \sum_{(\tau, y_\tau) \in \mathcal{D}} \mathrm{ELBO}_\tau \triangleq \mathrm{ELBO} \tag{3}$$

4.1.1 Optimising the ELBO

Note that this is an enormous sum over $|\mathcal{D}|$ elements. However, this can be approximated via Importance Sampling, or Bernoulli Sampling (Botev et al., 2017).

$$\begin{aligned} \mathrm{ELBO} &= \sum_{(\tau, y_\tau) \in \mathcal{D}} \mathbb{E}_q \left[\log p(y_\tau \mid \mathbf{h}_\tau) \right] - D_{\mathrm{KL}} q(\mathbf{h}_\tau) p(\mathbf{h}_\tau) \\ &= \sum_{(\tau, y_\tau) \in \mathcal{D}^+} \mathbb{E}_q \left[\log p(y_\tau \mid \mathbf{h}_\tau) \right] - D_{\mathrm{KL}} q(\mathbf{h}_\tau) p(\mathbf{h}_\tau) \\ &+ \sum_{(\tau, y_\tau) \in \mathcal{D}^-} \mathbb{E}_q \left[\log p(y_\tau \mid \mathbf{h}_\tau) \right] - D_{\mathrm{KL}} q(\mathbf{h}_\tau) p(\mathbf{h}_\tau) \end{aligned} \tag{4}$$

By using Bernoulli Sampling, ELBO can be approximated by:

$$\mathrm{ELBO} \approx \sum_{c : s_c = 1} \frac{\mathrm{ELBO}_{\tau_c}}{b_c} \tag{5}$$

where $p(s_c = 1) = b_c$ can be defined for each element c. We can define a probability distribution of sampling from $\mathcal{D}^+$ and $\mathcal{D}^-$ – similarly to Bayesian Personalised Ranking (Rendle et al., 2009), we sample one negative triple for each positive one — we use a constant probability for each element depending on whether it is in the positive or negative set. We end up with the following estimate:

$$\mathrm{ELBO} \approx \sum_{i=1}^{n} \frac{\mathrm{ELBO}_{\tau_c^+}}{b_c^+} + \frac{\mathrm{ELBO}_{\tau_c^-}}{b_c^-} \tag{6}$$

where $b_c^+ = |\mathcal{D}^+|/|\mathcal{D}|$ and $b_c^- = |\mathcal{D}^-|/|\mathcal{D}|$.

4.2 LATENT INFORMATION MODEL

In this model, we assume that the Knowledge Graph was generated according to the following generative model. Let $\mathcal{V} \triangleq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ the space of possible triples. We have that:

where $\tau \triangleq \langle \mathrm{s}, \mathrm{p}, \mathrm{o} \rangle$, and $\mathbf{h}_\tau \triangleq [\mathbf{h}_\mathrm{s}, \mathbf{h}_\mathrm{p}, \mathbf{h}_\mathrm{o}]$ denotes the sampled embedding representations of $\mathrm{s}, \mathrm{o} \in \mathcal{E}$ and $\mathrm{p} \in \mathcal{R}$. The set of latent variables in this model is $\mathcal{H} \triangleq \{\mathbf{h}_e \mid e \in \mathcal{E}\} \cup \{\mathbf{h}_\mathrm{p} \mid \mathrm{p} \in \mathcal{R}\}$. The joint probability of the variables $p(\mathcal{H}, \mathcal{D})$ is defined as follows:

$$p(\mathcal{H}, \mathcal{D}) \triangleq \prod_{e \in \mathcal{E}} p(\mathbf{h}_e) \prod_{\mathrm{p} \in \mathcal{E}} p(\mathbf{h}_\mathrm{p}) \prod_{(\tau, y_\tau) \in \mathcal{D}} p(y_\tau \mid \mathbf{h}_\tau) \tag{7}$$

The marginal distribution over $\mathcal{D}$ is then defined as follows:

$$p(\mathcal{D}) \triangleq \int \prod_{e \in \mathcal{E}} p(\mathbf{h}_e) \prod_{\mathrm{p} \in \mathcal{E}} p(\mathbf{h}_\mathrm{p}) \prod_{(\tau, y_\tau) \in \mathcal{D}} p(y_\tau \mid \mathbf{h}_\tau) d\mathcal{H} \tag{8}$$

The log-marginal likelihood of the data is the following:

$$\log p(\mathcal{D}) \geq \mathbb{E}_q \left[\log p(\mathcal{D} \mid \mathcal{H})\right] - D_{\mathrm{KL}} q(\mathcal{H}) p(\mathcal{H}) \tag{9}$$

4.3 LINK PREDICTION

Dataset	Scoring Function	MR		Hits @		
		Filter	Raw	1	3	10
WN18	V DistMult (LIM)	786	798	0.671	0.931	0.947
	DistMult	813	827	0.754	0.911	0.939
	V ComplEx (LIM)	**753**	**765**	0.934	**0.945**	**0.952**
	ComplEx*	-	-	**0.939**	0.944	0.947
WN18 RR	V DistMult (LIM)	6095	**6109**	0.357	0.423	0.440
	DistMult	8595	8595	0.367	0.390	0.412
	V ComplEx (LFM)	6500	6514	0.385	0.446	0.489
	ComplEx**	**5261**	-	**0.41**	**0.46**	**0.51**
FB15K -257	V DistMult (LIM)	679	813	0.171	0.271	0.397
	DistMult	**355**	**501**	**0.187**	**0.282**	0.400
	V ComplEx (LIM)	1221	1347	0.168	0.260	0.369
	ComplEx**	339	-	0.159	0.258	**0.417**

Table 1: Filtered and Mean Rank (MR) for the models tested on the WN18, WN18RR and FB15K datasets. Hits@m metrics are filtered. Variational written with a "V". *Results reported from (Trouillon et al., 2016b) and **Results reported from (Dettmers et al., 2017) for ComplEx model

.

Table 1 shows definite improvements on WN18 for Variational ComplEx compared with the initially published ComplEX. We believe this due to the well-balanced model regularisation induced by the zero mean unit variance Gaussian prior.

We now compare our model to the previous state-of-the-art multi-relational generative model TransG (Xiao et al., 2016), as well as to a previously published probabilistic embedding method KG2E (similarly represents each embedding with a multivariate Gaussian distribution) (He et al., 2015) on the WN18 dataset.

Table 2 makes clear the improvements in the performance of the previous state-of-the-art generative multi-relational knowledge graph model. LFM has marginally worse performance than the state-of-the-art model on raw Hits@10. We conjecture two reasons may cause this discrepancy. Firstly, the fact the authors of TransG use negative samples provided only (True negative examples), whereas we generated our negative samples using the LCWA. Secondly, we only use one negative sample per

Dataset	Scoring Function	MR		Raw Hits@	Filtered Hits @		
		Raw	Filter	10	1	3	10
WN18	KG2E He et al. (2015)	**362**	**345**	0.805	-	-	0.932
	TransG (Generative) Xiao et al. (2016)	345	357	**0.845**	-	-	0.949
	Variational ComplEx (LFM)	753	765	0.836	**0.934**	**0.945**	**0.952**

Table 2: Variational Framework vs. Generative Modles

positive to estimate the Evidence Lower Bound using Bernoulli sampling, whereas it is likely they used significantly more negative samples. This conjecture was proved true in a follow-up experiment on Nations; increasing performance on raw Hits@10 when using 20 negative samples, with no change in filtered Hits@10.

5 Link Prediction Analysis

Section 5.1 and Section 5.2 explores the predictions made by Latent Information Model with ComplEx scoring function, trained with Bernoulli sampling to estimate the ELBO on the WN18RR dataset, then Section 5.3 will analyse the values of embeddings learnt for this task. Lastly, Section 5.3.1 will perform an extrinsic evaluation on learnt embedding representations for the more accessible to interpret Nations dataset.

We split the analysis into the predictions of subject ($(?, r, o)$) or object ($(s, r, ?)$) for each test fact. Note all results are filtered predictions, i.e., ignoring the predictions made on negative examples generated under LCWA.

5.1 Subject Prediction

	Proportion	Hits@1	Hits@3	Hits@10
_hypernym	0.399170	0.091926	0.123102	0.162270
_derivationally_related_form	0.342693	0.947858	0.956238	0.959032
_member_meronym	0.080728	0.007905	0.019763	0.035573
_has_part	0.054882	0.011628	0.058140	0.122093
_instance_hypernym	0.038928	0.393443	0.508197	0.713115
_synset_domain_topic_of	0.036375	0.219298	0.315789	0.464912
_also_see	0.017869	0.589286	0.625000	0.625000
_verb_group	0.012444	0.743590	0.974359	0.974359
_member_of_domain_region	0.008296	0.000000	0.038462	0.115385
_member_of_domain_usage	0.007658	0.000000	0.000000	0.000000
_similar_to	0.000957	1.000000	1.000000	1.000000

Table 3: Latent Information Model with ComplEx: Subject Prediction on WN18RR

Table 3 shows that the relation "_derivationally_related_form", comprising 34% of test subject predictions, was the most accurate relation to predict for Hits@1 when removing the subject from the tested fact. Contrarily, "_member_of_domain_region" with zero Hits@1 subject prediction, making up less than 1% of subject test predictions. However, "_member_meronym " was the least accurate and prominent (8% of the test subject predictions) for subject Hits@1.

5.2 Object Prediction

Table 4 displays similar results to Table 3, as before the relation "_derivationally_related_form" was the most accurate relation to predict Hits@1. Table 4 differs from Table 3 as it highlights Model A's its inability to achieve a high Hits@1 performance predicting objects for the "_hypernym" relation, which is significantly hindering model performance as it is the most seen relation in the test set— its involvement in 40% of object test predictions.

	Proportion	Hits@1	Hits@3	Hits@10
_hypernym	0.399170	0.000000	0.014388	0.046363
_derivationally_related_form	0.342693	0.945996	0.957169	0.959032
_member_meronym	0.080728	0.031621	0.047431	0.086957
_has_part	0.054882	0.034884	0.081395	0.139535
_instance_hypernym	0.038928	0.024590	0.081967	0.131148
_synset_domain_topic_of	0.036375	0.035088	0.043860	0.078947
_also_see	0.017869	0.607143	0.625000	0.625000
_verb_group	0.012444	0.897436	0.974359	0.974359
_member_of_domain_region	0.008296	0.038462	0.076923	0.076923
_member_of_domain_usage	0.007658	0.000000	0.000000	0.000000
_similar_to	0.000957	1.000000	1.000000	1.000000

Table 4: Latent Information Model with ComplEx: Object Prediction on WN18RR

5.3 Embedding Analysis

These results hint at the possibility that the slightly stronger results of WN18 are due to covariances in our variational framework able to capture information about symbol frequencies. We verify this by plotting the mean value of co-variance matrices, as a function of the entity or predicate frequencies (Figure 3). The plots confirm our hypothesis: covariances for the variational Latent Information Model grows with the frequency, and hence the LIM would put a preference on predicting relationships between less frequent symbols in the knowledge graph. This also suggests that covariances from the generative framework can capture genuine information about the generality of symbolic representations.

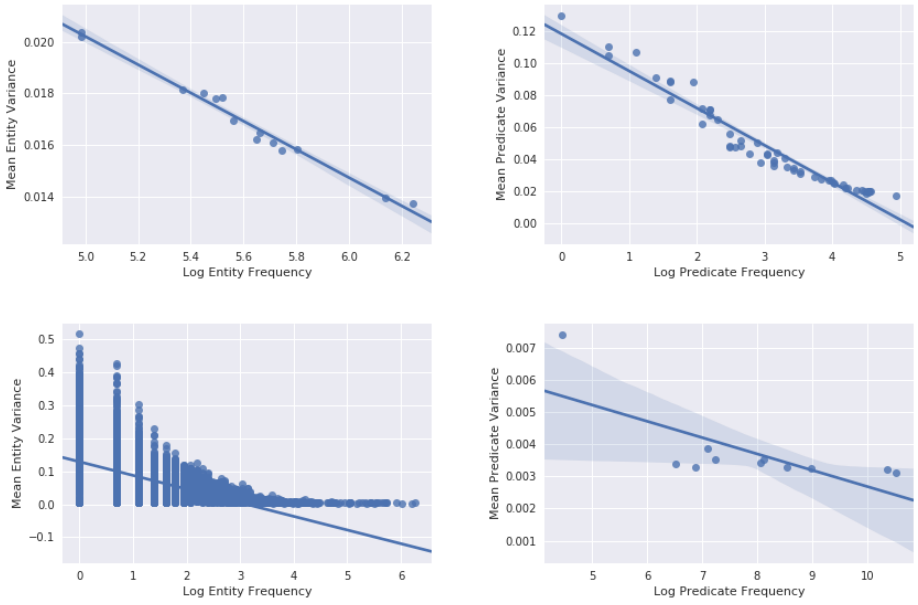

Figure 3: Mean Variance vs. log frequency. From left to right: Nations Entity Analysis, Nations Predicate Analysis, WN18RR Entity Analysis and WN18RR Predicate Analysis.

5.3.1 Extrinsic Evaluation: Visual Embedding Analysis

We project the high dimensional mean embedding vectors to two dimensions using Probabilistic Principal Component Analysis (PPCA) (Tipping & Bishop, 1999) to project the variance embedding vectors down to two dimensions using Non-negative Matrix Factorisation (NNMF) (Févotte & Idier,

2011). Once we have the parameters for a bivariate normal distribution, we then sample from the bivariate normal distribution 1,000 times and then plot a bi-variate kernel density estimate of these samples. By visualising these two-dimensional samples, we can conceive the space in which the entity or relation occupies. We complete this process for the subject, object, relation, and a randomly sampled corrupted entity (under LCWA) to produce a visualisation of a fact, as shown in Figure 4.

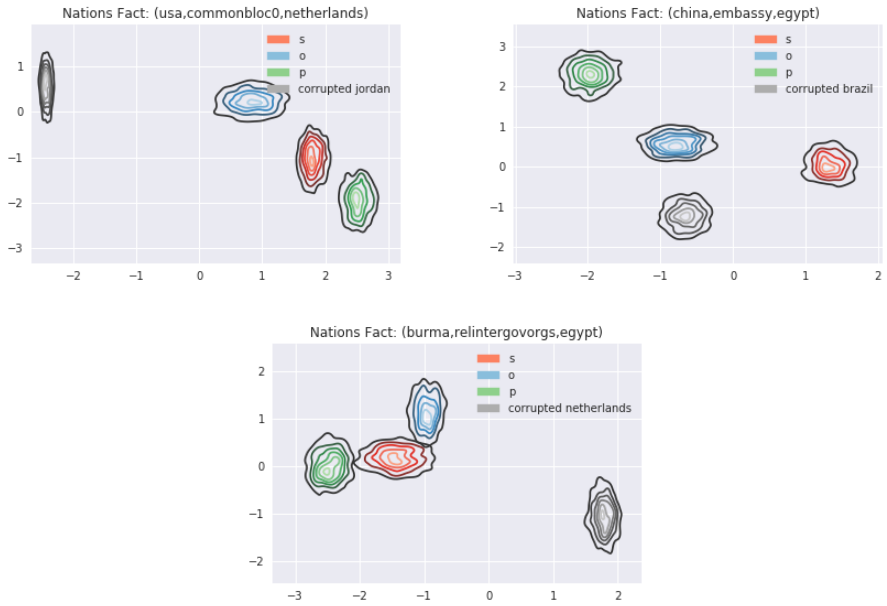

Figure 4: True Positives

Figure 4 displays three true positives from test time predictions. The plots show that the variational framework can learn high dimensional representations which when projected onto lower (more interpretable) dimensions.

Figure 4 displays a clustering of the subject, object and predicate that create a positive (true) fact. We also observe a separation between the items which generate a fact and a randomly sampled (corrupted) entity which is likely to create a negative (false) fact. The first test fact "(USA, Commonbloc0, Netherlands)" shows clear irrationality similarity between all objects in the tested fact, i.e. the vectors are pointing towards a south-east direction. We can also see that the corrupted entity Jordan is quite a distance away from the items in the tested fact, which is good as Jordan does not share a common bloc either USA or Netherlands.

5.4 Conclusion

We have successfully created a framework allowing a model to learn embeddings of any prior distribution that permits a re-parametrisation trick via any score function that permits maximum likelihood estimation of the scoring parameters. We have shown, from preliminary experiments, that these display competitive results with current models. Overall, we believe this work will enable knowledge graph researchers to work towards the goal of creating models better able to express their predictive uncertainty.

6 Further Work

The score we acquire at test time even through forward sampling does not seem to differ much compared with the mean embeddings, thus using the learnt uncertainty to impact the results positively is a fruitful path. We would also like to see additional exploration into various encoding functions, as we used only the most basic for these experiments.

ACKNOWLEDGMENTS

We would like to thank all members of the Machine Reading lab for useful discussions.

REFERENCES

Oren Barkan. Bayesian neural word embedding. *CoRR*, abs/1603.06571, 2016. URL http://arxiv.org/abs/1603.06571.

Daniel Bean, Honghan Wu, Olubanke Dzahini, Matthew Broadbent, Robert James Stewart, and Richard James Butler Dobson. Knowledge graph prediction of unknown adverse drug reactions and validation in electronic health records. *Scientific Reports*, 7(1), 11 2017. ISSN 2045-2322.

Christopher M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight Uncertainty in Neural Networks. *ArXiv e-prints*, May 2015.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013*, pp. 2787–2795, 2013.

Aleksandar Botev, Bowen Zheng, and David Barber. Complementary sum sampling for likelihood approximation in large scale classification. In Aarti Singh et al. (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1030–1038. PMLR, 2017.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pp. 10–21, 2016. URL http://aclweb.org/anthology/K/K16/K16-1002.pdf.

A. Bražinskas, S. Havrylov, and I. Titov. Embedding Words as Distributions with a Bayesian Skip-gram Model. *ArXiv e-prints*, November 2017.

Wenhu Chen, Wenhan Xiong, Xifeng Yan, and William Yang Wang. Variational knowledge graph reasoning. In *NAACL-HLT*, 2018.

T. R. Davidson, L. Falorsi, N. De Cao, T. Kipf, and J. M. Tomczak. Hyperspherical Variational Auto-Encoders. *ArXiv e-prints*, April 2018.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. *arXiv preprint arXiv:1707.01476*, 2017.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In Sofus A. Macskassy et al. (eds.), *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14*, pp. 601–610. ACM, 2014. ISBN 978-1-4503-2956-9.

Cédric Févotte and Jérôme Idier. Algorithms for nonnegative matrix factorization with the β-divergence. *Neural Computation*, 23(9):2421–2456, 2011. doi: 10.1162/NECO_a_00168. URL https://doi.org/10.1162/NECO_a_00168.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In Doina Precup et al. (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1183–1192. PMLR, 2017.

Zoubin Ghahramani. Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553): 452–459, 2015.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL http://proceedings.mlr.press/v9/glorot10a.html.

Shizhu He, Kang Liu, Guoliang Ji, and Jun Zhao. Learning to represent knowledge graphs with gaussian embedding. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pp. 623–632, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3794-6. doi: 10.1145/2806416.2806502. URL http://doi.acm.org/10.1145/2806416.2806502.

Ashish Kapoor, Kristen Grauman, Raquel Urtasun, and Trevor Darrell. Active learning with gaussian processes for object categorization. In *IEEE 11th International Conference on Computer Vision, ICCV 2007*, pp. 1–8. IEEE Computer Society, 2007. ISBN 978-1-4244-1630-1.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL http://arxiv.org/abs/1412.6980.

Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. *UvA*, pp. 1–14, 2013. URL http://arxiv.org/abs/1312.6114.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016a.

Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016b.

Y. Miao, E. Grefenstette, and P. Blunsom. Discovering Discrete Latent Topics with Neural Variational Inference. *ArXiv e-prints*, June 2017.

Yishu Miao, Lei Yu, and Phil Blunsom. Neural variational inference for text processing. *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

Andrew Y. Ng and Michael I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In Thomas G. Dietterich et al. (eds.), *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001]*, pp. 841–848. MIT Press, 2001.

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, 2016.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: bayesian personalized ranking from implicit feedback. In Jeff A. Bilmes et al. (eds.), *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pp. 452–461. AUAI Press, 2009.

Michael E. Tipping and Chris M. Bishop. Probabilistic principal component analysis. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 61(3):611–622, 1999.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In Maria-Florina Balcan et al. (eds.), *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 2071–2080. JMLR.org, 2016a.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. *CoRR*, abs/1606.06357, 2016b. URL http://arxiv.org/abs/1606.06357.

Luke Vilnis and Andrew McCallum. Word representations via gaussian embedding. *CoRR*, abs/1412.6623, 2014. URL http://arxiv.org/abs/1412.6623.

Ekaterina Vylomova, Laura Rimell, Trevor Cohn, and Timothy Baldwin. Take and Took, Gaggle and Goose, Book and Read: Evaluating the Utility of Vector Differences for Lexical Relation Learning. In *ACL*, 2016.

Han Xiao, Minlie Huang, and Xiaoyan Zhu. Transg : A generative model for knowledge graph embedding. In *ACL*, 2016.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *ICLR*, 2015.

Jingwei Zhang, Jeremy Salwen, Michael Glass, and Alfio Gliozzo. Word semantic representations using bayesian probabilistic tensor factorization. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014. doi: 10.3115/v1/d14-1161.

APPENDIX

EXPERIMENTAL SETUP

We run each experiment over 500 epochs and validate every 50 epochs. Each KB dataset is separated into 80 % training facts, 10% development facts, and 10% test facts. During the evaluation, for each fact, we include every possible corrupted version of the fact under the local closed world assumption, such that the corrupted facts do not exist in the KB. Subsequently, we make a ranking prediction of every fact and its corruptions, summarised by mean rank and filtered hits@m.

During training Bernoulli sampling to estimate the ELBO was used, with linear warm-up (Bowman et al., 2016; Davidson et al., 2018), compression cost (Blundell et al., 2015), ADAM (Kingma & Ba, 2014) Glorot's initialiser for mean vectors (Glorot & Bengio, 2010) and variance values initialised uniformly to embedding size^{-1}. We experimented both with a $\mathcal{N}(0, 1)$ and a $\mathcal{N}(0, \text{embedding size}^{-1})$ prior on the latent variables.

PROOF: LFM

The marginal distribution over $\mathcal{D}$ is then defined as follows:

$$p(\mathcal{D}) \triangleq \int \prod_{(\tau, y_\tau) \in \mathcal{D}} p(\mathbf{h}_\tau) p(y_\tau \mid \mathbf{h}_\tau) d\mathcal{H} \tag{10}$$

The log-marginal likelihood of the data is the following:

$$\begin{aligned}
\log p(\mathcal{D}) &= \log \int \prod_{(\tau, y_\tau) \in \mathcal{D}} p(\mathbf{h}_\tau) p(y_\tau \mid \mathbf{h}_\tau) d\mathcal{H} \\
&\geq \int \log \prod_{(\tau, y_\tau) \in \mathcal{D}} p(\mathbf{h}_\tau) p(y_\tau \mid \mathbf{h}_\tau) d\mathcal{H} \\
&= \int \sum_{(\tau, y_\tau) \in \mathcal{D}} \log p(\mathbf{h}_\tau) + \log p(y_\tau \mid \mathbf{h}_\tau) d\mathcal{H} \\
&= \sum_{(\tau, y_\tau) \in \mathcal{D}} \int \log p(\mathbf{h}_\tau) + \log p(y_\tau \mid \mathbf{h}_\tau) d\mathbf{h}_\tau \\
&= \sum_{(\tau, y_\tau) \in \mathcal{D}} \text{ELBO}_\tau
\end{aligned} \tag{11}$$

Given a triple τ, the term $\text{ELBO}(\tau)$ can be rewritten as follows:

$$
\begin{aligned}
\mathrm{ELBO}_\tau &= \int \log p(y_\tau \mid \mathbf{h}_\tau) p(\mathbf{h}_\tau) d\mathbf{h}_\tau \\
&= \int \log \frac{p(y_\tau \mid \mathbf{h}_\tau) p(\mathbf{h}_\tau)}{q(\mathbf{h}_\tau)} q(\mathbf{h}_\tau) d\mathbf{h}_\tau \\
&= \int \left[\log p(y_\tau \mid \mathbf{h}_\tau) + \log p(\mathbf{h}_\tau) - \log q(\mathbf{h}_\tau)\right] q(\mathbf{h}_\tau) d\mathbf{h}_\tau \\
&= \int \log p(y_\tau \mid \mathbf{h}_\tau) q(\mathbf{h}_\tau) d\mathbf{h}_\tau + \int q(\mathbf{h}_\tau) \log \frac{p(\mathbf{h}_\tau)}{q(\mathbf{h}_\tau)} d\mathbf{h}_\tau \\
&= \mathbb{E}_q \left[\log p(y_\tau \mid \mathbf{h}_\tau)\right] - D_{\mathrm{KL}} q(\mathbf{h}_\tau) p(\mathbf{h}_\tau)
\end{aligned}
\tag{12}
$$

PROOF: LIM

The marginal distribution over $\mathcal{D}$ is then defined as follows:

$$
p(\mathcal{D}) \triangleq \int \prod_{e \in \mathcal{E}} p(\mathbf{h}_e) \prod_{\mathrm{p} \in \mathcal{E}} p(\mathbf{h}_\mathrm{p}) \prod_{(\tau, y_\tau) \in \mathcal{D}} p(y_\tau \mid \mathbf{h}_\tau) d\mathcal{H} \tag{13}
$$

The log-marginal likelihood of the data is the following:

$$
\begin{aligned}
\log p(\mathcal{D}) &= \log \int \prod_{x \in \mathcal{E} \cup \mathcal{R}} p(\mathbf{h}_x) \prod_{(\tau, y_\tau) \in \mathcal{D}} p(y_\tau \mid \mathbf{h}_\tau) d\mathcal{H} \\
&= \log \int p(\mathcal{H}) p(\mathcal{D} \mid \mathcal{H}) d\mathcal{H} \\
&= \log \int \frac{q(\mathcal{H})}{q(\mathcal{H})} \left[p(\mathcal{H}) p(\mathcal{D} \mid \mathcal{H})\right] d\mathcal{H} \\
&\geq \int q(\mathcal{H}) \log \frac{p(\mathcal{H}) p(\mathcal{D} \mid \mathcal{H})}{q(\mathcal{H})} d\mathcal{H} \\
&= \int q(\mathcal{H}) \left[\log p(\mathcal{D} \mid \mathcal{H}) + \log \frac{p(\mathcal{H})}{q(\mathcal{H})}\right] d\mathcal{H} \\
&= \mathbb{E}_q \left[\log p(\mathcal{D} \mid \mathcal{H})\right] - D_{\mathrm{KL}} q(\mathcal{H}) p(\mathcal{H})
\end{aligned}
\tag{14}
$$