Uncertain Knowledge Graph Embedding Using Probabilistic Logic Neural Networks

Zijie Huang Roshni G. Iyer Zhiping Xiao
605322634 105294233 604775684
{zijiehuang, roshnigiyer, patricia.xiao}@cs.ucla.edu

Abstract

Knowledge graph (KG) embedding embeds components of a KG into low-dimensional continuous vector spaces, while preserving the inherent structure of the KG. Existing embedding methods (e.g. TransE, DistMult) have two main drawbacks: (1) They fail to leverage the underlying domain knowledge, which can be captured by the symbolic rule-based approach with first-order logic; and (2) they are mostly designed for deterministic KGs and thereby fail to model the inherent uncertainty present in real-world KGs. In this paper, we propose the uncertain probabilistic logic neural network (PKGE) which captures domain knowledge and uncertainty information, preserves entity-relation semantic information, preserves graph structure, and can be trained effectively. To achieve this, PKGE employs the Markov Logic Network (MLN) to learn first-order logic and encodes uncertainty by leaning confidence scores using the novel Uncertain KG Embedding (UKGE) model. We conduct optimization using the variational EM algorithm.

1 Introduction

Knowledge Graph (KG) is a multi-relational graph, where entities (nodes) are interconnected with each other through various types of relationships. Such relational data can be characterized as a triple of the form *(h:head entity, r:relation, t:tail entity)*, also called a fact, indicating that two entities are connected by a specific relation. A large number of KGs [1, 2] have been successfully applied to many real-world applications [3], from semantic parsing [4],named entity disambiguation [5, 6], to information extraction [7]. As KG's coverage is limited, one fundamental problem is how to reason missing links based on existing triplets [8].

Existing KG reasoning models such as TransE [9], DistMult [10], and ComplEx [11] represent entities and relations in a continuous vector space, and defines a scoring function for each fact to measure its plausibility, based on these embeddings. However, these approaches have two main drawbacks.

Firstly, the underlying symbolic nature of KGs can serve as domain knowledge to enhance inference quality, which is not considered in the above models. For example, given a logic rule that $\forall x, y, Teacher(x, y) \implies Student(y, x)$, and a fact that A is the teacher of B, we can derive that B is a student of A. A principled way to utilize such logic rules is Markov Logic Network (MLN) [12], which combines first-order logic and probabilistic graphical models. However, inference solely using logic rules is insufficient, as many missing triples do not belong to any rules. A better approach is to combine MLN with the above KG embedding models.

Secondly, in many real-world scenarios, KGs contain knowledge uncertainty [13, 14]. Each fact is associated with a confidence score that represents the likelihood of the fact to be true. While how to embed deterministic KGs has been widely studied, embedding uncertain KGs is challenging.

Therefore, we propose PKGE, a novel KG embedding model which utilizes logical rules learned

from uncertain KGs and can be used to reason confident scores. We test PKGE on downstream inference tasks, against competitive existing baselines.

2 Related Work

Rule-based Methods. Prior methods have considered using MLN based approaches for KG reasoning [8]. MLNs can effectively learn weights of logical rules through a probabilistic graphical model and thereby handle knowledge uncertainty while leveraging domain knowledge. MLN models the joint distribution of all triples in UKGs as:

$$p_w(\mathbf{v}_O, \mathbf{v}_H) = \frac{1}{Z} exp(\sum_{l \in L} w_l \sum_{g \in G_l} p(\text{g is true})) \quad (1)$$

where $L = l_{i=1}^{L}$ is a set of logic rules, w_l is the weight of logic rule l, $p(\text{g is true})$ is the probability a grounding g is true in UKG. $\mathbf{v}_O$ and $\mathbf{v}_H$ are the confidence scores for observed triples and hidden triples respectively. Eqn (1) can be trained by maximizing ground truth confidence scores for observed triplets, i.e. $logp_w(\mathbf{v}_O)$. Since the optimization requires integration over $\mathbf{v}_H$, recent studies usually solve it using the variation EM algorithm. However the inference procedure of MLNs is inefficient and there exists missing triplets which logical rules cannot model.

Knowledge Graph Embedding. KGE models [9] learn important entity and relational embeddings that preserve node and edge semantic information in KGs. They measure the plausibility of a fact as the distance or semantic meaning between two entities in the vector space, by translating the relation as: $\mathbf{h}_r + r \approx \mathbf{t}_r$, where $\mathbf{h}_r$, $\mathbf{t}_r$ are the entity embeddings projected in a relation-specific space. These models can be scalably trained using gradient descent methods, however, they are not able to leverage the domain knowledge inherently present in the logical rules.

As such, we propose PKGE that combines bother methods to infer missing triplets utilizing both semantic and domain knowledge incorporated in the learned embeddings and logical rules, and also can be trained efficiently. To the best of our knowledge, we are the first to propose this approach. While several approaches that have sought to marry MLN and KG embedding based models to infer missing triplet information, such approaches do not utilize the uncertainty of the observed triplets to enhance learned embeddings and logical rule weights [8]. Additionally, other approaches have proposed to improve probabilistic logic reasoning by utilizing MLNs and Graph Neural Networks (GNNs) as opposed to KG embedding (KGE) models to construct the inference network. However, the reported findings of such methods are irreproducible, and appear to have lower performance compared to pLogicNet when examining reproduced results. As such, we omit using such approaches in our model and as a baseline for comparison.

3 PKGE Model

In this section, we will describe our model, PKGE, which is used for reasoning in uncertain knowledge graphs. PKGE utilizes the state-of-the-art logical rule based method MLN and embedding method of UKGE for effective learning and inference of KGs. Variational EM algorithm is applied to optimize parameter learning of the MLN and KGE model. As such, PKGE exploits domain knowledge, KG uncertainty, and effective training for the inference of missing triplets.

3.1 Problem Formulation

Consider a KG denoted as (E, R, O), where E is the set of entities, R is the set of relations, and O is the set of observed triplets (h, r, t). Each triplet (h, r, t) has an associated continuous indicator variable $\mathbf{v}_{(h,r,t)}$, also referred to as confidence score, which denotes how certain we are of a triple existing. We aim to leverage $\mathbf{v}_O = \{\mathbf{v}_{(h,r,t)} = ConfidenceScore\}_{(h,r,t) \in O}$, to predict $\mathbf{v}_H = \{\mathbf{v}_{(h,r,t)} = ?\}_{(h,r,t) \in H}$.

3.2 PKGE: Variational EM

Our approach uses a uncertain Markov Logic Network to model the joint distribution of both hidden and observed triplets as in Eqn (1). We use PSL to compute $p(g\ is\ true)$ in Eqn (1) to take into account uncertainty in KGs: For a rule $\gamma : \gamma_{body} \rightarrow \gamma_{head}$, we can rewrite it as $\neg\gamma_{body} \vee \gamma_{head}$. And its value can be

computed as:

$$p_{\gamma_{body}\to\gamma_{head}} = min\{1, 1 - y(\gamma_{body}) + y(\gamma_{head})\}$$

$$y(\gamma) = \begin{cases} max\{0, v_{(h_1,r_1,t_1)} + v_{(h_2,r_2,t_2)} - 1\} \\ \qquad \text{if } \gamma = (h_1,r_1,t_1) \wedge (h_2,r_2,t_2) \\ min\{1, v_{(h_1,r_1,t_1)} + v_{(h_2,r_2,t_2)}\} \\ \qquad \text{if } \gamma = (h_1,r_1,t_1) \vee (h_2,r_2,t_2) \\ 1 - v_{(h,r,t)} \\ \qquad \text{if } \gamma = \neg(h,r,t) \end{cases} \tag{2}$$

Directly maximizing $logp(V_O)$ is computational-expensive, since it requires integration over $\mathbf{v}_H$. Therefore we instead optimize the evidence lower bound (ELBO) of the log-likelihood function, which is given in Eqn (3).

$$\begin{aligned} &log\ p_w(V_O) \geq L_{ELBO}(q_\theta, p_w) := \\ &\quad E_{q_\theta(V_H)}[log\ p_w(V_O, V_H) - log\ q_\theta(V_H)] \end{aligned} \tag{3}$$

Such a lower bound can be efficiently trained with variational EM algorithm, which consist of E-step and M-step. In E-step, we fix p_w and update q_θ to minimize the KL divergence between $q_\theta(V_H)$ and $p_w(V_H|V_O)$, during which the knowledge preserved by the logic rules can be effectively distilled into the learned embeddings. In M-step, we fix the learned embedding model q_θ and update p_w to maximize the log likelihood of all triples, i.e. $E_{q_\theta(V_H)}[log\ p_w(V_O, V_H)]$.In this way, the knowledge graph embedding model provides extra supervision for learning weights. We illustrate the details of each step in the following sections.

3.3 E-step Inference

In E-step, we aim to infer the posterior distribution of hidden variables, e.g. $p_w(V_H|V_O)$. We adopt the mean-field method to approximate the true posterior distribution $p_w(V_H|V_O)$ with a variational distribution $q_\theta(V_H)$. The key idea of mean-field method is to assume independence over all hidden variables, so the joint probability can be factorized as product of each variable distribution, i.e. $q_\theta(V_H) = \prod_{(h,r,t)\in H} q_\theta(v_{(h,r,t)})$. We assume the Beta distribution for each hidden variable and parametrize it with a knowledge graph embedding model. To sum up, the variational distribution $q_\theta(V_H)$ is given by Eqn (4), where $f(x_h, x_r, x_t)$ is a scoring function of KGE models defined on triplets.

$$\begin{aligned} q_\theta(V_H) &= \prod_{(h,r,t)\in H} q_\theta(v_{(h,r,t)}) \\ &= \prod_{(h,r,t)\in H} Beta(v_{(h,r,t)}|f(x_h, x_r, x_t)) \end{aligned} \tag{4}$$

The reason we choose the Beta distribution is that, its domain is within range $[0, 1]$, which is the same as confidence scores. Specifically, the probability density function is given by $Beta(x|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$. Given a fixed Beta distribution, the inferred confidence score should be the mode, which maximizes the probability density function. As the confidence score for each triple is unique, we need to guarantee the Beta distribution to have a unique mode. Therefore, both α and β should be greater or equal to 1. Meanwhile they should not all equal to 1, otherwise the distribution would degenerate to uniform distribution. To satisfy all the requirements, we set $\alpha = f(x_h, x_r, x_t) + 1$ and $\alpha + \beta = 3$, where $f(x_h, x_r, x_t)$ is the score function of KGE models.

The optimal variational distribution $q_\theta(V_H)$ for approximation should be as close to the the true posterior distribution $p_w(V_H|V_O)$ as possible. By minimizing the KL divergence between them, we get the fixed-point condition for optimal $q_\theta(V_H)$ as in Eqn (5). Here $MB(h,r,t)$ is the Markov Blanket of (h,r,t). It contains all the triplets that appear in any groundings together with (h,r,t).

$$\begin{aligned} &logq_\theta(v_{(h,r,t)}) \simeq E_{q_\theta(v_{MB(h,r,t)})} log(p_w(v_{(h,r,t)}|v_{MB(h,r,t)})) \\ &\simeq E_{q_\theta(v_{MB(h,r,t)})} log(p_w(v_{(h,r,t)}, v_{MB(h,r,t)})) \end{aligned} \tag{5}$$

To get the optimal $q_\theta(v_{(h,r,t)})$ that follows Eqn (5), calculation of expectation is required. However, this requires integration over all possible values within $[0, 1]$ which is computationally-expensive. To simplify the condition, we follow stochastic variational inference and estimate the expectation with a sample $\widehat{v}_{MB(h,r,t)}$ as in Eqn (6). Specifically, we check each triplet in the Markov Blanket $(h', r', t') \in MB(h,r,t)$: if it is observed, we set the value as ground-truth confidence score, i.e. $\widehat{v}_{(h',r',t')} = s$; if it is hidden, we sample the value following

$q_\theta(v_{(h',r',t')})$, i.e. $\widehat{v}_{(h',r',t')} \sim q_\theta(v_{(h',r',t')})$.

$$\widehat{v}_{MB(h,r,t)} = \{\widehat{v}_{(h',r',t')}\}_{(h',r',t') \in MB(h,r,t)}$$

$$\widehat{v}_{(h',r',t')} \begin{cases} = s_{(h',r',t')} & \text{if } (h',r',t') \text{ is observed} \\ \sim q_\theta(v_{(h',r',t')}) & \text{if } (h',r',t') \text{ is hidden} \end{cases} \quad (6)$$

By doing such, Eqn (5) can be approximated as $log(q_\theta(v_{(h,r,t)})) \simeq log(p_w(v_{(h,r,t)}|\widehat{v}_{MB(h,r,t)}))$, which can be further simplified as:

$$\begin{aligned} q_\theta(v_{(h,r,t)}) &\simeq p_w(v_{(h,r,t)}|\widehat{v}_{MB(h,r,t)}) \\ &= p_w(v_{(h,r,t)}, \widehat{v}_{MB(h,r,t)}) \end{aligned} \quad (7)$$

The intuition behind behind Eqn (7) is that the optimal knowledge graph embedding model should reach a consensus with the logic rules on the distribution of each hidden triplets. As such, the domain knowledge preserved by logic rules can be effectively distilled into the learned knowledge graph embedding.

To find the optimal $q_\theta(v_{(h,r,t)})$ as in Eqn (7), we start by using the current θ to compute $p_w(v_{(h,r,t)}, \widehat{v}_{MB(h,r,t)})$, which is similar to the method in [8]. Then the value is fixed as target, and we update θ by minimizing the reverse KL divergence between $p_w(v_{(h,r,t)}, \widehat{v}_{MB(h,r,t)})$ and $q_\theta(v_{(h,r,t)})$. This is equivalent to maximize the following objective function.

$$\begin{aligned} O_{\theta,U} = &- \sum_{(h,r,t)\in H} E_{p_w(v_{(h,r,t)}|\widehat{v}_{MB(h,r,t)})} \\ &[log \frac{p_w(v_{(h,r,t)}|\widehat{v}_{MB(h,r,t)})}{q_\theta(v_{(h,r,t)})}] \\ \sim &\sum_{(h,r,t)\in H} E_{p_w(v_{(h,r,t)}|\widehat{v}_{MB(h,r,t)})} log(q_\theta(v_{(h,r,t)})) \\ \sim &\sum_{(h,r,t)\in H} log(q_\theta(\widehat{v}_{(h,r,t)})) \end{aligned} \quad (8)$$

The third line is that, we further estimate the expectation in line 2 by drawing a sample $\widehat{v}_{(h,r,t)}$ from $p_w(v_{(h,r,t)}|\widehat{v}_{MB(h,r,t)})$, similar as in Eqn (6). More specifically, we sample $\widehat{v}_{(h,r,t)} \sim p_w(v_{(h,r,t)}|\widehat{v}_{MB(h,r,t)})$.

Additionally, we can also use observed triplets in O as positive samples to help the training of knowledge graph embedding model $q_\theta(v_{(h,r,t)})$. Therefore, we also try to maximize the following supervised objective function:

$$O_{\theta,L} = \sum_{(h,r,t)\in O} log(q_\theta(v_{(h,r,t)} = s_{(h,r,t)})) \quad (9)$$

By adding Eqn (8) and Eqn (9) together, we get our final objective function for optimizing θ:

$$O_\theta = O_{\theta,U} + O_{\theta,L} \quad (10)$$

3.4 M-step Learning

In M-step, we aim to learn the parameter w in p_w. Specifically, we will fix q_θ and further update logic weights w by maximizing ELBO as in Eqn (3). However, directly maximizing the log-likelihood function is difficult, as we need to compute the partition function Z in Eqn (1). Therefore, we instead follow existing studies and maximize the pseudolikelihood function: where the second equation is derived from the independence property of the MLN in Eqn (1).

$$\begin{aligned} L_{PL}(w) &= E_{q_\theta(V_H)}[\sum_{h,r,t} log\, p_w(v_{(h,r,t)}|v_{O\cup H\setminus(h,r,t)})] \\ &= E_{q_\theta(V_H)}[\sum_{h,r,t} log\, p_w(v_{(h,r,t)}|v_{MB(h,r,t)})] \end{aligned} \quad (11)$$

We again estimate the expectation in Eqn (11) by drawing a sample from $q_\theta(V_H)$: if (h,r,t) is an observed triplet, we set $\widehat{v}_{(h,r,t)} = s_{(h,r,t)}$; if it is hidden, we set $\widehat{v}_{(h,r,t)} \sim q_\theta(v_{(h,r,t)}$. Afterwards, we can optimize parameter w by maximizing the following objective function:

$$O_w = \sum_{(h,r,t)\in O\cup H} log\, p_w(v_{(h,r,t)} = \widehat{v}_{(h,r,t)}|\widehat{v}_{MB(h,r,t)}) \quad (12)$$

The intuition behind Eqn (12) is that: for each observed triplet $(h,r,t) \in O$, we seek to maximize $p_w(v_{(h,r,t} = s_{(h,r,t)}|\widehat{v}_{MB(h,r,t)})$; for each hidden triplet $(h,r,t) \in H$, we sample a value $\widehat{v}_{(h,r,t)}$ from $q_\theta(v_{(h,r,t)}$ and use the value as target for updating the probability $p_w(v_{h,t,t} = \widehat{v}_{h,t,t}|\widehat{v}_{MB(h,r,t)})$. In this way, the knowledge graph embedding model q_θ actually adds supervision to facilitate the learning process of logic rule weights.

3.5 Optimization and Inference

To optimize our approach, we iteratively perform E-step and M-step until convergence. As the mutual influence of Markov Logic Network and knowledge graph embedding model is through two posterior disrtibution over each triplet, during training we only use hidden triplets that can be predicted by both of them[1]. Afterwards, both p_w and q_θ can be employed to infer the confident score of hidden triples. In practice we find that q_θ consistently outperforms p_w and q_θ can predict hidden triplets

[1] see appendix for details about hidden triplet generation

that cannot be inferred from logic rules. Therefore, we use q_θ to predict confidence scores for hidden triplets. In the Experiment section, we show such phenomena with the results of incorporating rule-based p_w in prediction as pLogiNet [8], by introducing λ as hyper-parameter to control the contribution of MLN rules.

4 Experiments

4.1 Set Up

Dataset. CN15k is a subgraph of the commonsense KG ConceptNet, containing 15,000 entities and 241,158 uncertain relation facts in English. This dataset was adapted from the UKGE paper, which was normalized such that the confidence scores are between 0.1 and 1.0. We will evaluate using the same evaluation method as that was used in the UKGE paper.

Evaluation Tasks. We evaluate on the confidence score prediction task. The problem definition is as follows. Each relation fact l has a true confidence score s_l associated with it, denoted (l, s_l). We would predict the confidence score of l by answering queries $(l, ?s_l)$ in the test set and report the mean squared error (MSE) and mean absolute error (MAE).

Baselines. Since our model aims to improve the uncertainty KG embedding model by combining MLN rule-based approach [15], we would compare to two types of baselines, (i) an uncertain graph embedding model URGE [16], and (ii) UKGE [14] and its simplified variations UKGE_{n^-} and UKGE_{p^-}.

- Uncertain Graph Embedding Model. URGE was proposed to embed uncertain graphs. However, it fails to deal with different types of relations that are so common and essential in KG domain. URGE also only produces node embeddings. We included this result (from UKGE paper) for completion purpose.
- UKGE and its variations. UKGE is the first uncertain KG embedding model that is able to capture both entities and relations information in KE embedding space. We first compare our model to two of its simplified versions UKGE_{n^-} and UKGE_{p^-}. UKGE_{n^-} does not employ negative sampling and it was trained with observed facts only. UKGE_{p^-} uses MSE loss for unseen facts without utilizing PSL. UKGE_{rect} and UKGE_{logi} both uses PSL to minimize the distance to satisfaction of unseen relation facts, but with different mapping functions, namely bounded rectifier and logistic function, from plausibility score to confidence score for a give triplet.

4.2 Results

As Table 1 shows, PKGE_{kge} outperforms UKGE in confidence score prediction for MSE. On CN15k for MAE, PKGE was slightly worse than UKGE. A plausible explanation is that PKGE tends to produce higher confidence scores for triples instead of underestimating them, which results in an increase in the MAE. PKGE's overestimation of the confidence scores due to the noise in the CN15k data may have lead to worsened performance in MAE especially for CN15k.

PKGE_{kge} uses q_θ from the KGE model without using p_w of MLN. PKGE_λ corresponds to how much weight we give it to rules in our prediction of score. $\lambda = 1$ means only logical rules are used. Because there are triples in the test set that cannot be predicted using MLN (as we constructed MLN in such a way that all triples in the premise of a rule must be observed triples), the result of $\lambda = 1$ was based on the ones that are hypothesized according to our constraint. It is interesting to see that the model's performance is increased by the help of logical rules in the training; however, during testing the model is better off without using the logical rules. Such results reveal that PKGE is capable of distilling information from both KG confidence scores and logical rules into the KG embedding space.

5 Conclusions

In this paper, we introduced PKGE, which successfully utilizes logical rules to improve the performance of the UKGE model. PKGE is trained using the variational EM algorithm, and is evaluated on relation fact confidence score prediction. Our experiment results demonstrate effective performance of PKGE compared to the baseline models of URGE, and UKGE and its variant models.

CN15k		
Model	MSE	MAE
URGE	10.32	22.72
$\texttt{UKGE}_{n-}$	23.96	30.38
$\texttt{UKGE}_{p-}$	9.02	20.05
$\texttt{UKGE}_{rect}$	8.61	**19.90**
$\texttt{UKGE}_{logi}$	9.86	20.74
$\text{PKGE}_{\lambda\,=\,1}$	36.33	53.39
$\text{PKGE}_{\lambda\,=\,0.05}$	6.93	21.18
PKGE_{kge}	**6.84**	20.93

Table 1: Mean Square Error (MSE) and Mean Absolute Error (MAE) for the CN15k dataset ($\times 10^{-2}$).

References

[1] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *In SIGMOD Conference*, pages 1247–1250, 2008.

[2] M. Jakob A. Jentzsch D. Kontokostas P. N. Mendes S. Hellmann M. Morsey P. van Kleef S. Auer J. Lehmann, R. Isele. Dbpedia: A large-scale, multilingual knowledge base extracted from wikipedia. In *Semantic Web Journal*, pages 167–195, 2015.

[3] Q. Wang, Z. Mao, B. Wang, and L. Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017.

[4] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, October 2013.

[5] Sherzod Hakimov, Salih Atilay Oto, and Erdogan Dogdu. Named entity recognition and disambiguation using linked data and graph-based centrality scoring. In *Proceedings of the 4th International Workshop on Semantic Web Information Management*, 2012.

[6] Zhicheng Zheng, Xiance Si, Fangtao Li, Edward Y. Chang, and Xiaoyan Zhu. Entity disambiguation with freebase. In *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '12, page 82–89, 2012.

[7] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 541–550, 2011.

[8] Meng Qu and Jian Tang. Probabilistic logic neural networks for reasoning. In *Advances in neural information processing systems*, 2019.

[9] 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 26*, pages 2787–2795. 2013.

[10] Bishan Yang, Wen tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and

relations for learning and inference in knowledge bases. *ICLR2015*, abs/1412.6575, 2014.

[11] 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 International Conference on Machine Learning - Volume 48*, ICML'16, page 2071–2080. JMLR.org, 2016.

[12] Parag Singla and Pedro Domingos. Discriminative training of markov logic networks. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2*, AAAI'05, page 868–873, 2005.

[13] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4444–4451, 2017.

[14] Xuelu Chen, Muhao Chen, Weijia Shi, Yizhou Sun, and Carlo Zaniolo. Embedding uncertain knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3363–3370, 2019.

[15] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine learning*, 62(1-2):107–136, 2006.

[16] Jiafeng Hu, Reynold Cheng, Zhipeng Huang, Yixang Fang, and Siqiang Luo. On embedding uncertain graphs. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 157–166, 2017.

Acknowledgements

We would like to thank Prof. Guy Van den Broeck for stimulating lectures on the topic of relational learning and probabilistic graphical models. We also thank our research advisor, Prof. Yizohu Sun, for her valuable feedbacks and suggestions on this work. We would also like to thank Vivian Cheng and Shirley Chen from the Scalable Analytics Institute (Scai Lab) at UCLA for their valuable discussions.

Appendix

5.1 Experiment Setup

We split each dataset into three parts: 85% for training, 7% for validation, and 8% for testing. We use Adam optimizer for training, for which we set the exponential decay rates $\beta_1 = 0.9$ and $\beta_2 = 0.99$. Results of baselines and other models were taken directly from the `UKGE` paper. The result of our model was based on the best hyper-parameter of $\{lr = 0.001, k = 300\}$ and 5 iterations of variational EM. The logic rule generation process was the same according to Chen et el. [14], in which only the logic rules with *hit ratio* greater than a threshold is used.

5.2 Future Work

For future work, we plan to investigate three areas of interest. First, we plan to investigate different methods for logical rule generation. In our model, as well as `UKGE` and pLogicNet, only the unseen relation facts inferred from observed facts are included in the analysis. Examining on hidden relation facts inferred from partially but not all observed facts could be interesting. Second, we plan to evaluate our model on more KG tasks, specifically on ranking relation facts in the forms of $(h, r, \underline{?t})$, or relational learning $(h, \underline{?r}, t)$. Third, unlike previous KGE models, we also plan to investigate using node attributes in addition to edge features for UKGE model enhancement. Also we are going to test our performance on more datasets, such as NL27k and PPI5k from the research work of [14].

5.3 Hidden Triplet Construction

Since KGE model can predict every triplet while MLN can only predict triplets inferred by logic rules, we construct hidden triplets that can be predicted by MLN. Specifically, an unobserved triplet (h, r, t) is added to H if we can find a grouding $[premise] \longrightarrow [hypothesis]$, where the hypothesis is (h, r, t) and premise only contains triplets in the observed set O. We construct H with brute-force search as in [8] .

5.4 Course Relevance

Our project is closely-related to this course, from both data and modeling aspects:

- The dataset we use is a relational graph dataset where the relations have confidence scores that introduce uncertainty.
- From a modeling aspect, we will use a relational modeling framework that is directly related to this course:
    1. We are going to use a Markov Logic Network, which is related to the logic rules we have been discussing in the course.
    2. Our research project uses KG embedding models which help to enhance reasoning on KGs, a topic that we have also be introduced to in this course.
    3. In the future, this model could be adapted to probabilistic programming languages, such as DeepProbLog, that we've learned in class.

5.5 Feedback

Our research project is closely related to this course and required a lot of thought and discussion about our model. We had to really understand all of the related work surrounding this topic to thoroughly design our model. Despite this challenge, our project was well-worth the immense effort that we put in. It is also exciting that our project was successful as indicated by the experiment results. We hope to work on the ideas we listed as part of future work, and aim towards submitting this research as a workshop or conference paper.

Thank you to the instructor and TAs for a great quarter!