Learning Embeddings for Approximate Lifted Inference in MLNs

Mohammad Maminur Islam
Department of Computer Science
The University of Memphis
mislam3@memphis.edu

Somdeb Sarkhel
Adobe Research
sarkhel@adobe.com

Deepak Venugopal
Department of Computer Science
The University of Memphis
dvngopal@memphis.edu

Abstract

We present a dense representation for Markov Logic Networks (MLNs) that encodes symmetries in the MLN. Such a representation is particularly important in the context of lifted inference algorithms that scale up by exploiting symmetries. By leveraging advances in neural networks, we learn a novel representation for symmetries that is hard to specify explicitly using hand-crafted features. Specifically, we learn an embedding for MLN objects that predicts the context of an object, i.e., objects that appear along with it in formulas of the MLN, since common contexts indicate symmetry in the distribution. Importantly, using such a formulation we can adapt existing skip-gram models to learn symmetries efficiently. In this paper, we present an overview of our approach and some experimental results that show its promise in improving inference algorithms for MLNs.

1 Introduction

Neural embeddings have been extremely successful as a general approach to learn efficient and effective representations for a variety of real-world domains including words [16, 22], sentences [21], knowledge graphs [18], images [12], etc. Inspired by these successes, in this paper, we present a novel representation for Markov Logic Networks (MLNs) [8] using neural embeddings to represent symmetries in the model. Our main motivation for such a representation stems from the fact that over the last several years, it has been widely recognized that exploiting symmetries in MLNs (and in other statistical relational models such as PSL [4]) yields exponential improvements in the scalability of inference algorithms. Thus, several algorithms that are collectively referred to as *lifted inference* [23] algorithms have been proposed that exploit symmetries in the MLN.

However, identifying symmetries in the MLN efficiently and effectively is non-trivial. Several previous approaches [7, 33, 9, 29, 31, 34, 6, 17] have been proposed that exploit both exact and approximate symmetries in inference. However, these methods are either restrictive in nature (i.e., limited to exact symmetries) [9], or in other cases require hand-crafted features [34], or work on the graph structure [6, 17] which can be very large in practical domains such as NLP. In this paper we seek to leverage advances in neural networks to learn more powerful symmetry-based representations for MLNs. Specifically, we describe a distributed representation for objects in the MLN based on the premise that if two objects are symmetrical, they are exchangeable in ground formulas of the MLN. Thus, a possible representation is to vectorize objects using ground formulas and learn a dense embedding from these vectors. However, learning from vectors that directly encode ground formulas is not scalable since the input representation is as big as the ground Markov network. Therefore, inspired by the successful *skip-gram* model, we propose a novel, more scalable approach that creates an embedding based on local context information for objects. The embeddings layer will then learn similar representations for objects that have similar contexts. Importantly, using this formulation, we can adapt skip-gram model architectures [16] to learn the representation efficiently. Our brief ex-

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

perimental evaluation shows that, performing inference in the MLN based on such a representation yields scalable and accurate results.

2 Background

2.1 Markov Logic Networks

Markov logic networks (MLNs) are template models that define uncertain, relational knowledge as first-order formulas with weights. Weights encode uncertainty in a formula. Given a set of constants/objects that represent the domains of variables in the MLN, an MLN represents a factored probability distribution over the possible worlds, in the form of a Markov network, where the potentials are defined from the formulas grounded with the domain objects. The two common inference problems over MLNs are marginal inference and MAP inference which are both intractable problems.

2.2 Skip-Gram Models

Skip-gram models are used to learn an embedding over words based on the context in which they appear in the training data (i.e., nearby words). Word2vec [16] is a popular model of this type, where we train a neural network based on pairs of words seen in the training data. That is, we learn to predict a word based on its context or nearby words. The hidden layer typically has a much smaller number of dimensions as compared to the input/output layers. Thus, the hidden-layer learns a low-dimensional embedding that is capable of mapping words to their contexts. Typically the hidden-layer output is used as features for other text processing tasks, as opposed to using handcrafted features.

2.3 Related Work

Lifted inference [23, 7, 9, 33] is the predominant approach to improving the scalability of inference in relational models. Our approach is more closely related to pre-processing approaches that exploit approximate symmetries by changing the evidence using binary matrix factorization [31] or clustering. Other approaches for lifted inference identify automorphisms in graphs [6]. These methods have been applied to marginal as well as MAP inference [3, 17]. However, since these methods compute symmetries on the graph structure, for large practical problems, it becomes infeasible to create the entire graph structure. More recently, Anand et al. [1] developed methods for identifying contextual symmetries for probabilistic graphical models, and our proposed approach can be viewed as extending this to first-order models by learning a distributed representation for these symmetries. Finally, Rocktaschel and Riedel [25] developed subsymbolic representations and learning for logical inference operators. Specifically, they developed vector representations for logical symbols and used them within theorem proving. Our approach can be viewed as developing representations for probabilistic reasoning taking advantage of distributional symmetries.

3 Symmetry-based Representation Learning

Previous works have defined symmetry in terms of *orbits* in the automorphism groups of variables (or atoms) in the Markov network underlying the MLN [6]. However, the number of variables in complex MLNs tend to be extremely large. Therefore, here, we characterize symmetry of objects in the domain of the MLN. Specifically, our task is to learn a representation such that two objects that are symmetrical have a similar representation. One way of measuring this symmetry is to check if we can exchange the objects in the ground formulas of the MLN without changing the distribution represented by the MLN. Specifically, given the evidence, if the ground formulas in the MLN have a similar truth assignment before and after the exchange, we can safely exchange the objects.

Using the above perspective, we can encode objects in terms of the truth assignment to ground formulas and compare these encodings to determine symmetry between objects. A simple encoding is therefore a vector that specifies the truth assignment of each ground formula. Specifically, each object X is represented as a vector v_X that specifies whether each ground formula is `True`/`False`/`unknown`. For groundings where X does not appear, the vector component has a

default value unknown, and for groundings where X appears, its truth value can be computed as True/False/unknown given the evidence database. However, such an encoding can be extremely large since the number of ground formulas can potentially be orders of magnitude larger than the number of objects. For example, if the MLN formula is $\mathtt{R}(x, y) \wedge \mathtt{R}(y, z) \Rightarrow \mathtt{R}(z, x)$, if the domain-size of the variables x, y and z is equal to 100, the vector to encode the truth assignment to the ground formulas will have dimensionality equal to 1 million. Essentially, such an encoding explicitly represents all the potentials of the ground Markov network in vector form which is equivalent to constructing the ground Markov network. Thus, learning a dense representation from such an encoding (since the encoding will naturally be very sparse), leads to an extremely large input layer in the neural network and is not a scalable solution.

Therefore, instead of encoding objects as a vector over all ground formulas, we learn the dense representation in a more scalable way inspired by the *skip-gram* model which is widely used in word embeddings. The main idea in our approach is to train a neural network that predicts objects from other objects. Specifically, borrowing terminology from skip-gram models, we seek to predict the *context* of an object. The hidden layer of the neural network learns to represent the input object vectors in a reduced dimension such that objects that appear in similar contexts are placed close together in the embedded space. To do this, we vectorize an object X as a one-hot encoding represented by vector v_X. Note that, this encoding will be orders of magnitude smaller than a vector representing the ground formulas of the MLN. That is, the size of this encoding is bounded by the largest domain-size in the MLN.

To learn an embedding from the one-hot encoding of objects, for every ground formula satisfied by the evidence, we predict a vector corresponding to an object in that formula given a vector of another object in that formula. To draw an analogy to skip-gram models, we refer to objects that appear together in satisfied formulas as appearing in the context of each other. Specifically, let $f_1 \ldots f_K$ denote the ground formulas, let $\mathbf{O}_i$ represent the objects in f_i, and let o_{ij} represent the j-th object in the i-th formula (according to some canonical ordering of predicates in the formulas). The representation learner seeks to maximize,

$$\sum_{i=1}^{K} \sum_{j=1}^{|\mathbf{O}_i|} \sum_{-c \leq k \leq c; k \neq 0} \log P(o_{ij+k} | o_{ij})$$

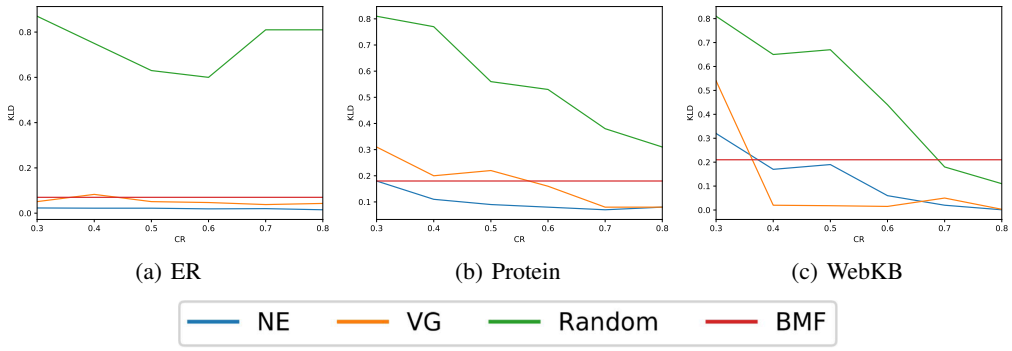
Specifically, c defines a sliding window-size over the objects in $\mathbf{O}_i$. Here, $P(o_{ij+k}|o_{ij})$ is defined as a softmax function proportional to $\exp(v'^{\top}_{o_{ij+k}} v_{o_{ij}})$, where v_o refers to the input vector representation for object o, and v'_o refers to its output vector representation. An example of specifying the training data to learn the embedding is shown below.

Example 1. *Consider a simple formula* $\mathtt{R}(x) \wedge \mathtt{S}(x, y)$. *Let* $\Delta_x = \{X_1, X_2, X_3\}$ *and* $\Delta_y = \{Y_1, Y_2\}$. *Assume a closed world with the evidence database* $\mathtt{R}(X_1)$, $\mathtt{R}(X_2)$, $\mathtt{R}(X_3)$, $\mathtt{S}(X_1, Y_1)$, $\mathtt{S}(X_2, Y_1)$, $\mathtt{S}(X_3, Y_2)$. *The training instances include,* v_{X_1}, v_{Y_1}; v_{X_2}, v_{Y_1}; v_{X_3}, v_{Y_2};. *That is, given* X_1 *or* X_2 *at the input layer, we predict* Y_1 *at the output layer, and given* X_3 *as input we predict* Y_2 *as the output layer. This means that the hidden layer in the model will derive features such that* X_1 *and* X_2 *will make common predictions at the output layer. At the same time, since* X_3 *has a different context, it needs to predict* Y_2 *at the output layer, and therefore, the hidden layer encoding for* X_3 *will be different from that of* X_1 *and* X_2.

However, defining context of an object only in satisfied ground formulas has limitations when the evidence is very sparse. In such cases, very few ground formulas may be satisfied. For example, consider an MLN $\mathtt{R}(x) \wedge \mathtt{S}(x, y) \wedge \mathtt{T}(y)$, with evidence $\mathtt{R}(X_1)$, $\mathtt{S}(X_1, Y_1)$, $\mathtt{R}(X_2)$, $\mathtt{S}(X_2, Y_1)$, $\mathtt{S}(X_2, Y_2)$. Since none of the ground formulas are satisfied by the evidence, we are unable to detect any symmetries in this case, even though symmetries may exist on partially satisfied formulas. To account for sparse evidences, we make a *closed world* assumption, where unknown atoms are assumed false in partially satisfied formulas, and then compute the contexts as before.

4 Experiments

We present some brief results of our experiments to evalute the effectiveness of the embedding in the context of inference. Specifically, we used the Gensim package [24] to learn the object embed-



(a) ER (b) Protein (c) WebKB

Figure 1: Average KL-divergence between the original marginals (before domain reduction) and approximate marginals (after domain reduction)

dings. We then sampled objects from the domains of the MLN by selecting an object and removing all its neighbors in the embedded space. Essentially, the selected object represents all its neighboring objects in the embedded space. We reduce the size of the MLN domains, and modify the evidence database based on these new domains. We then performed inference using this modified evidence, and projected the inference results obtained on the modified MLN to the variables of the original MLN (we skip the details due to lack of space). To perform inference, we used Tuffy [20], a state-of-the-art inference system for MLNs to compute marginal probabilities of query variables, and projected the marginal probabilities to the original MLN variables. We compared our approach (NE) with Venugopal and Gogate's [34] approach (VG) that compresses the MLN using K-Means clustering with features based on counts of atoms satisfied by the evidence and Binary Matrix Factorization (BMF) [31] that pre-processes binary evidence with a low-rank approximation. We also added a baseline method that reduces the evidence database by randomly sampling the evidence atoms in the evidence, which we refer to as Random. For NE, we set the hidden layer to have 300 neurons (a typical size recommended for word embeddings [16]).

We computed the average KL-Divergence for the marginal probabilities computed for the query variables, where the divergence is measured w.r.t the marginal probabilities obtained when we perform inference using the full evidence (no domain reduction). We measured the KL-Divergence for different amounts of compressions (CR), i.e., the average of the ratios of original-domain-size and new-domain-size (after domain-reduction) across all domains. For BMF, achieving the right CR was not possible since changing rank does not change the CR. Therefore for BMF, we varied rank from 20 to 100 (30 is the fault value in the NIMFA implementation of BMF), which is similar to the ranks used in [31], and report the best results across the ranks. Fig. 1 shows our results for three benchmarks taken from Alchemy [13]. However, Tuffy gets struck in grounding the MLN when the amount of evidence is very large to process. Therefore, we sample the evidences in the benchmarks and subsample it (10% of their original evidence database) to obtain our results. As shown in Fig. 1, NE outperformed the the other methods on two of the benchmarks in terms of accuracy. On the Webkb benchmark, NE performed worse than VG at lower values of CR, but was more consistent in terms of trade-off between CR and accuracy, with accuracy improving as we increased CR. Note that, randomly sampling the domains of the MLN without considering their symmetries performs very poorly as compared to all methods that take advantage of symmetries when modifying the original evidence.

5 Conclusion

In this paper, we proposed a novel subsymbolic representation for MLNs that is based on symmetries in the underlying model. The main motivation for this representation was that leveraging symmetries is crucial to scaling up inference in MLNs. We proposed an efficient way to learn the symmetry-based representation by predicting objects in the context of other objects in the MLN akin to *skip-gram* based word embeddings. Though our experiments demonstrated the promise of the proposed approach in lifted inference, the general idea of learning an embedding-based representation for MLNs can be useful in weight learning, transfer learning, etc.

References

[1] Anand, A.; Grover, A.; Mausam; and Singla, P. 2016. Contextual symmetries in probabilistic graphical models. In *IJCAI*, 3560–3568.

[2] Anand, A.; Noothigattu, R.; Singla, P.; and Mausam. 2017. Non-count symmetries in boolean & multi-valued prob. graphical models. In *AISTATS*, 1541–1549.

[3] Apsel, U.; Kersting, K.; and Mladenov, M. 2014. Lifting relational map-lps using cluster signatures. In *UAI*, 2403–2409.

[4] Bach, S. H.; Broecheler, M.; Huang, B.; and Getoor, L. 2017. Hinge-loss markov random fields and probabilistic soft logic. *J. Mach. Learn. Res.* 18(1):3846–3912.

[5] Bordes, A.; Weston, J.; Collobert, R.; and Bengio, Y. 2011. Learning structured embeddings of knowledge bases. In *AAAI*, 301–306.

[6] Bui, H. H.; Huynh, T. N.; and Riedel, S. 2013. Automorphism groups of graphical models and lifted variational inference. In *UAI*, 132–141.

[7] de Salvo Braz, R. 2007. *Lifted First-Order Probabilistic Inference*. Ph.D. Dissertation, University of Illinois, Urbana-Champaign, IL.

[8] Domingos, P., and Lowd, D. 2009. *Markov Logic: An Interface Layer for Artificial Intelligence*. San Rafael, CA: Morgan & Claypool.

[9] Gogate, V., and Domingos, P. 2011. Probabilistic Theorem Proving. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence*, 256–265. AUAI Press.

[10] Kautz, H.; Selman, B.; and Jiang, Y. 1997. A General Stochastic Approach to Solving Problems with Hard and Soft Constraints. In Gu, D.; Du, J.; and Pardalos, P., eds., *The Satisfiability Problem: Theory and Applications*. New York, NY: American Mathematical Society. 573–586.

[11] Khot, T.; Balasubramanian, N.; Gribkoff, E.; Sabharwal, A.; Clark, P.; and Etzioni, O. 2015. Exploring markov logic networks for question answering. In *EMNLP*, 685–694.

[12] Kiela, D., and Bottou, L. 2014. Learning image embeddings using convolutional neural networks for improved multi-modal semantics. In *EMNLP*, 36–45.

[13] Kok, S.; Sumner, M.; Richardson, M.; Singla, P.; Poon, H.; and Domingos, P. 2006. The Alchemy System for Statistical Relational AI. Technical report, Department of Computer Science and Engineering, University of Washington, Seattle, WA. http://alchemy.cs.washington.edu.

[14] Kopp, T.; Singla, P.; and Kautz, H. A. 2015. Lifted symmetry detection and breaking for MAP inference. In *NIPS*, 1315–1323.

[15] Mckay, B. D., and Piperno, A. 2014. Practical graph isomorphism, ii. *J. Symb. Comput.* 60:94–112.

[16] Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*, 3111–3119.

[17] Mladenov, M., and Kersting, K. 2015. Equitable partitions of concave free energies. In *UAI*, 602–611.

[18] Nickel, M.; Murphy, K.; Tresp, V.; and Gabrilovich, E. 2016. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE* 104(1):11–33.

[19] Niepert, M. 2012. Markov chains on orbits of permutation groups. In *UAI*, 624–633. AUAI Press.

[20] Niu, F.; Ré, C.; Doan, A.; and Shavlik, J. W. 2011. Tuffy: Scaling up statistical inference in markov logic networks using an rdbms. *PVLDB* 4(6):373–384.

[21] Palangi, H.; Deng, L.; Shen, Y.; Gao, J.; He, X.; Chen, J.; Song, X.; and Ward, R. 2016. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.* 24(4):694–707.

[22] Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *EMNLP*.

[23] Poole, D. 2003. First-Order Probabilistic Inference. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, 985–991. Acapulco, Mexico: Morgan Kaufmann.

[24] Řehůřek, R., and Sojka, P. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 45–50.

[25] Rocktäschel, T., and Riedel, S. 2017. End-to-end differentiable proving. In *NIPS*, 3791–3803.

[26] Rocktäschel, T.; Singh, S.; and Riedel, S. 2015. Injecting logical background knowledge into embeddings for relation extraction. In *NAACL-HLT*, 1119–1129.

[27] Sarkhel, S.; Singla, P.; and Gogate, V. 2015. Fast lifted map inference via partitioning. In *NIPS*, 3222–3230.

[28] Singla, P., and Domingos, P. 2008. Lifted First-Order Belief Propagation. In *AAAI*, 1094–1099.

[29] Singla, P.; Nath, A.; and Domingos, P. 2014. Approximate Lifting Techniques for Belief Propagation. In *AAAI*, 2497–2504.

[30] Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; and Bouchard, G. 2016. Complex embeddings for simple link prediction. In *ICML*, 2071–2080.

[31] van den Broeck, G., and Darwiche, A. 2013. On the complexity and approximation of binary evidence in lifted inference. In *Advances in Neural Information Processing Systems 26*, 2868–2876.

[32] Van den Broeck, G., and Niepert, M. 2015. Lifted probabilistic inference for asymmetric graphical models. In *Proceedings of the 29th Conference on Artificial Intelligence (AAAI)*.

[33] Van den Broeck, G.; Taghipour, N.; Meert, W.; Davis, J.; and De Raedt, L. 2011. Lifted Probabilistic Inference by First-Order Knowledge Compilation. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, 2178–2185.

[34] Venugopal, D., and Gogate, V. 2014. Evidence-based clustering for scalable inference in markov logic. In *ECML PKDD*.

[35] Venugopal, D.; Chen, C.; Gogate, V.; and Ng, V. 2014. Relieving the computational bottleneck: Joint inference for event extraction with high-dimensional features. In *EMNLP*, 831–843. ACL.

[36] Venugopal, D.; S.Sarkhel; and Gogate, V. 2016. Magician: Scalable Inference and Learning in Markov logic using Approximate Symmetries. Technical report, Department of Computer Science, The University of Memphis, Memphis, TN. https://github.com/dvngp/CD-Learn.