Learning Predictive Categories Using Lifted Relational Neural Networks

Gustav Šourek[1], Suresh Manandhar[2], Filip Železný[1], Steven Schockaert[3], and Ondřej Kuželka[3]

[1] Czech Technical University, Prague, Czech Republic
{souregus,zelezny}@fel.cvut.cz
[2] Department of Computer Science, University of York, York, UK
suresh.manandhar@york.ac.uk
[3] School of CS & Informatics, Cardiff University, Cardiff, UK
{SchockaertS1,KuzelkaO}@cardiff.ac.uk

Abstract. Lifted relational neural networks (LRNNs) are a flexible neural-symbolic framework based on the idea of lifted modelling. In this paper we show how LRNNs can be easily used to declaratively specify and solve a learning problem in which latent categories of entities and properties need to be jointly induced.

1 Introduction

Lifted models, such as Markov logic networks (MLNs [9]), are first-order representations that define patterns from which specific (ground) models can be unfolded. For example, in an MLN we may express the pattern that *friends of smokers tend to be smokers*, which then constrains the probabilistic relationships between specific individuals in the derived ground Markov network. Inspired by this idea, in [11] we introduced a method that uses weighted relational rules for learning feed-forward neural networks, called *Lifted Relational Neural Networks* (LRNNs). This approach differs from standard neural networks in two important ways: (i) the network structure is derived from symbolic rules and thus has an intuitive interpretation, and (ii) the weights of the network are tied to the first-order rules and are thus shared among different neurons.

In this paper, we show how LRNNs can be used to learn a latent category structure that is predictive in the sense that the properties of a given object can be largely determined by the category to which that object belongs, and dually, the objects satisfying a given property can be largely determined by the category to which that property belongs. This enables a form of transductive reasoning which is based on the idea that similar objects have similar properties. The proposed approach is similar in spirit to [5], which uses second-order MLNs instead. However, the use of LRNNs has an important advantage for learning latent concepts. This is because, being a non-probabilistic model, LRNNs do not need to invoke costly EM algorithms and they can therefore be more efficient.

2 Preliminaries: Lifted Relational Neural Networks

A lifted relational neural network (LRNN) $\mathcal{N}$ is a set of weighted definite first-order clauses. Let $\mathcal{H}_{\mathcal{N}}$ be the least Herbrand model of the classical theory $\{\alpha : (\alpha, w) \in \mathcal{N}\}$, with $\mathcal{N}$ an LRNN. We define the *grounding of* $\mathcal{N}$ as $\overline{\mathcal{N}} = \{(h\theta \leftarrow b_1\theta \wedge \cdots \wedge b_k\theta, w) : (h \leftarrow b_1 \wedge \cdots \wedge b_k, w) \in \mathcal{N} \text{ and } \{h\theta, b_1\theta, \ldots, b_k\theta\} \subseteq \mathcal{H}\}$.

Definition 1. *Let $\mathcal{N}$ be a LRNN, and let $\overline{\mathcal{N}}$ be its grounding. Let $g_\vee$, $g_\wedge$ and g_* be functions*[4] *from $\bigcup_{i=1}^{\infty} \mathbb{R}^i$ to $\mathbb{R}$. The* ground neural network *of $\mathcal{N}$ is a feed-forward neural network constructed as follows.*

- *For every ground atom h occurring in $\overline{\mathcal{N}}$, there is a neuron A_h with activation function $g_\vee$, called* atom neuron.
- *For every ground fact $(h, w) \in \overline{\mathcal{N}}$, there is a neuron $F_{(h,w)}$, called* fact neuron*, which has no input and always outputs the constant value w.*
- *For every ground rule $(h\theta \leftarrow b_1\theta \wedge \cdots \wedge b_k\theta, w) \in \overline{\mathcal{N}}$, there is a neuron $R_{h\theta \leftarrow b_1\theta \wedge \cdots \wedge b_k\theta}$ with activation function $g_\wedge$, called* rule neuron*. It has the atom neurons $A_{b_1\theta}, \ldots, A_{b_k\theta}$ as inputs, all with weight* 1*.*
- *For every rule $(h \leftarrow b_1 \wedge \cdots \wedge b_k, w) \in \mathcal{N}$ and every $h\theta \in \mathcal{H}$, there is a neuron $Agg^{h\theta}_{(h \leftarrow b_1 \wedge \cdots \wedge b_k, w)}$ with activation function g_*, called* aggregation neuron*. Its inputs are all rule neurons $R_{h\theta' \leftarrow b_1\theta' \wedge \cdots \wedge b_k\theta'}$ where $h\theta = h\theta'$ with all weights equal to* 1*.*
- *Inputs of an atom neuron $A_{h\theta}$ are the aggregation neurons $Agg^{h\theta}_{(h \leftarrow b_1 \wedge \cdots \wedge b_k, w)}$ and fact neurons $F_{(h\theta, w)}$, with the input weights determined by the outputs of the aggregation and fact neurons.*

Depending on the used families of activation functions $g_\wedge$, $g_\vee$ and g_*, we can obtain neural networks with different behavior. In this paper we will use:

$$g_\wedge(b_1, \ldots, b_k) = sigm\Big(\sum_{i=1}^{k} b_i - k + b_0\Big) \quad g_\vee(b_1, \ldots, b_k) = sigm\Big(\sum_{i=1}^{k} b_i + b_0\Big)$$

$$g_*(b_1, \ldots, b_m) = \max_i b_i$$

Note that $g_\wedge$ and $g_\vee$ are closely related to the conjunction and disjunction from Łukasiewicz logic, which is in accordance with the intuition that $g_\wedge$ should only have a high output if all its inputs are high, while $g_\vee$ should be high as soon as one of the inputs is high.

When the given LRNN contains *loops*, the resulting ground neural network will be recurrent. As recurrent neural networks are more difficult to train, we break the loops and partially expand the recursion; we omit the details.

3 Learning predictive categories

Let a set of entities be given, and for each entity, a list of properties that it satisfies. The basic assumption underlying our model is that there exist some

[4] These represent aggregation operators that can take a variable number of arguments.

(possibly overlapping) categories, such that every entity can be described accurately enough by its soft membership to each of these categories. We furthermore assume that these categories can themselves be organised in a set of higher-level categories. The idea is that the category hierarchy should allow us to predict which properties a given objects has, where the properties associated with higher-level categories are typically (but not necessarily) inherited by their sub-categories. To improve the generalization ability of our method, we assume that a dual category structure exists for properties. The main task we consider is to learn suitable (latent) category structures from the given input data.

To encode the above described model in a LRNN, we proceed as follows. We assume that the input is encoded as a set of facts of the form ($\mathit{HasProperty}(e, p)$, ± 1.0). For every entity e and for each category c at the lowest level of the category hierarchy, we construct the following ground rule:

$$w_{ec} : \mathit{IsA}(e, c)$$

Note that weight w_{ec} intuitively reflects the soft membership of e to the category c; it will be determined when training the ground network. Similarly, for each category c_1 at a given level and each category c_2 one level above, we add the following ground rule:

$$w_{c_1 c_2} : \mathit{IsA}(c_1, c_2)$$

In the same way, ground rules are added that link each property to a property category at the lowest level, and ground rules that link property categories to higher-level categories. To encode the idea that entity categories should be predictive of properties, we add the following ground rule for each entity category c_e and each property category c_p:

$$w_{c_e c_p} : \mathit{HasProperty}(c_e, c_p)$$

The weights $w_{c_e c_p}$ encode which entity categories are related to which property categories, and will again be determined when training the ground network. To encode transitivity of the is-a relationship, we simply add the following rule

$$w_{isa} : \mathit{IsA}(A, C) \leftarrow \mathit{IsA}(A, B), \mathit{IsA}(B, C)$$

To encode that the properties of entities are typically determined by their category, we add the next rule for each entity cluster c_e and property cluster c_p:

$$w'_{c_e c_p} : \mathit{HasProperty}(A, B) \leftarrow \mathit{IsA}(A, c_e), \mathit{IsA}(B, c_p), \mathit{HasProperty}(c_e, c_p)$$

We train the model using SGD as described in [11]. In particular, in a LRNN, there is a neuron for any ground literal which is logically entailed by the rules and facts in the LRNN. and the output of this neuron represents the truth value of this literal Therefore if we want to train the weights of the LRNN, we just optimize the weights of the network w.r.t. a loss function such as the mean squared error, where the loss function is computed from the desired truth values of the query literals and the outputs obtained from the respective atom neurons

4 Evaluation

To evaluate the potential of the proposed method, we have used the Animals dataset[5] which describes 50 animals in terms of 85 Boolean features, such as *fish, large, smelly, strong*, and *timid*. This dataset was originally crated in [7], and was used among others for evaluating a related learning task in [5]. For both objects and properties, we have used two levels of categories, with in both cases three categories at the lowest level and two categories at the highest level.

Figures 1 and 2 show the category membership degrees projected to first two principal components of a number of entities and properties, for the three lowest-level categories. Note that the membership degrees can be interpreted as defining a vector-space embedding. We can see, for instance, that sea mammals are clustered together, and that predators tend to be separated from herbivores. In Figure 2, we have highlighted two types of properties: colours and teeth-types. Note that these do not form clusters (e.g. a cluster of colours) but they represent, as prototypes, different clusters of properties which tend to occur together. For instance, *blue* is surrounded by properties which are typically held by water mammals, whereas white and red occur together with stripes, nocturnal, pads, and whereas gray occurs together with small and weak etc. We also evaluated the predictive ability of this model. We randomly divided the facts in the dataset into two halves, trained the model on one half and tested it on the other one, obtaining AUC ROC of 0.77. We also performed an experiment with a 90-10 splits, in order to be able to directly compare our results with the results from [5], and we obtained the same AUC PR 0.8 as reported therein (and AUC ROC 0.86).

5 Discussion

The proposed model essentially relies on the assumption that similar entities tend to have similar properties, for some similarity function which is learned implicitly in terms of category membership degrees. It is possible to augment this form of inference with other models of plausible reasoning, such as reasoning based on analogical (and other logical) proportions [6,8]. Moreover, as in [2], we could take into account externally obtained similarity degrees, using rules such as $w_l : \mathit{HasProperty}(A, B) \leftarrow \mathit{HasProperty}(C, B), \mathit{Similar}(A, C, l)$, where l denotes a certain level of similarity. Depending on the chosen activation function, such a network would behave similarly to 1-NN or similarly to kernel regression.Another natural extension would be to consider (subject, predicate, object) triples, and to learn soft categories for predicates, as well as for subjects and objects.

The model considered in this paper is related to statistical predicate invention [5] which relies on jointly clustering objects and relations. The dual representation of object and property categories is also reminiscent of formal concept analysis [4]. LRNNs themselves are also related to the long stream of research in neural-symbolic integration [1] and more recent approaches such as [10,3].

[5] Downloaded from `https://alchemy.cs.washington.edu/data/animals/`.

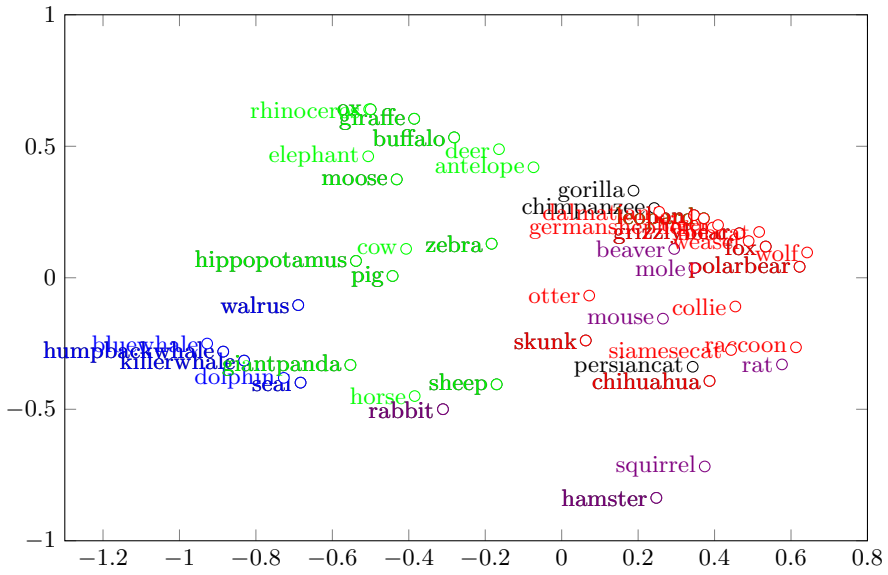

Fig. 1. Embedding of entities (animals, only a subset of entities is displayed). Several homogeneous groups of animals are highlighted: sea mammals (blue), large herbivores (green), rodents (violet), and other predators (red).

6 Conclusions and Outlook

We have illustrated how the declarative and flexible nature of LRNNs can be used to esily encode non-trivial learning scenarios. The model that we considered in this paper jointly learns predictive categories of entities and of their properties. The main strength of this approach lies in the ease with which the model can be extended to more complicated relational settings, which we plan to show in a longer version of this paper. We also plan to conduct a thorough experimental comparison of our method with state-of-the-art SRL methods such as MLNs. Finally, we also plan to use an extended version of this model as a component in an NLP pipeline.

Acknowledgments GS is supported by a CTU student grant competition project SGS16/093/OHK3/1T/13: "Developing novel machine learning methods with applications". OK is supported by a grant from the Leverhulme Trust (RPG-2014-164). SS is supported by ERC Starting Grant 637277.

References

1. Sebastian Bader and Pascal Hitzler. Dimensions of neural-symbolic integration-a structured survey. *arXiv preprint cs/0511042*, 2005.
2. Islam Beltagy, Cuong Chau, Gemma Boleda, Dan Garrette, Katrin Erk, and Raymond Mooney. Montague meets Markov: Deep semantics with probabilistic logical form. In *Proc. *SEM*, pages 11–21, 2013.

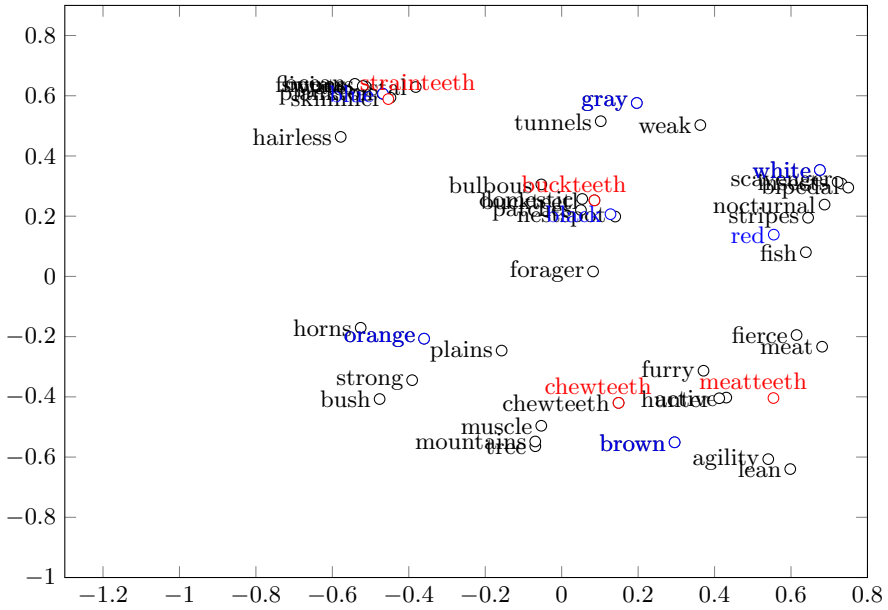

Fig. 2. Embedding of properties (only a subset of properties is displayed). Two representative groups of properties are shown in colour: colours (blue) and teeth-type (red).

3. William W Cohen. Tensorlog: A differentiable deductive database. *arXiv preprint arXiv:1605.06523*, 2016.
4. Bernhard Ganter, Gerd Stumme, and Rudolf Wille. *Formal concept analysis: foundations and applications*, volume 3626. springer, 2005.
5. Stanley Kok and Pedro Domingos. Statistical predicate invention. In *Proceedings of the 24th International Conference on Machine Learning*, pages 433–440, 2007.
6. Laurent Miclet, Sabri Bayoudh, and Arnaud Delhay. Analogical dissimilarity: definition, algorithms and two experiments in machine learning. *Journal of Artificial Intelligence Research*, 32:793–824, 2008.
7. Daniel N Osherson, Joshua Stern, Ormond Wilkie, Michael Stob, and Edward E Smith. Default probability. *Cognitive Science*, 15(2):251–269, 1991.
8. Henri Prade and Gilles Richard. Reasoning with logical proportions. In *Twelfth International Conference on the Principles of Knowledge Representation and Reasoning*, 2010.
9. Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine learning*, 62(1-2):107–136, 2006.
10. Tim Rocktäschel and Sebastian Riedel. Learning knowledge base inference with neural theorem provers. In *NAACL Workshop on Automated Knowledge Base Construction (AKBC)*, 2016.
11. Gustav Šourek, Vojtěch Aschenbrenner, Filip Železný, and Ondřej Kuželka. Lifted relational neural networks. In *Proceedings of the NIPS Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches*, 2015.