Learning Predictive Categories Using
*Lifted Relational Neural Networks*

Gustav Sourek¹, Suresh Manandhar², Filip Zelezný¹, Steven Schockaert³, and Ondrej Kuzelka³

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

1. What are Lifted Relational Neural Networks?
2. Learning Predictive Categories with LRNNs
Motivation

• How to learn with relational or graph-structured data in the presence of uncertainty?
  ➢ Lifted graphical models, e.g. Markov Logic

• How to efficiently learn latent concepts?
  ➢ Neural Networks (propositional concepts)

➢ How about latent relational concept learning?
  ➢ Lifted Relational Neural Networks
What is LRNN?

• Syntactically: **Set** of weighted first-order Horn clauses
  
  • 0.5 : water :- bondOH(X,Y)
  
  • 1.0 : bondOH(X,Y) :- H(X), O(Y), bond(X,Y)

  • LRNN encoding looks familiar - like a weighted Prolog program…

• Semantically: **Template** for **neural network construction**

  ➢ *We turn the template’s Herbrand models into NNs as follows.*
Network Construction

1. Every ground proposition (atom) which can be derived \* from a given LRNN model corresponds to an atom neuron.

2. Every ground rule \( h \leftarrow (b_1, \ldots, b_k) \) such that \((b_1, \ldots, b_k)\) can be derived \* from a given LRNN corresponds to a rule neuron.

3. To aggregate different groundings derived with the same rule’s ground head \{\(h \leftarrow (b_1, \ldots, b_k), \ldots, h \leftarrow (b_n, \ldots, b_n)\}\} there is an aggregation neuron

\* meaning it is present in the least Herbrand model
Putting it all together...
Weight Learning

• LRNN model := grounding of \{sample, template\} clauses

  ➢ Different samples result in different ground networks

  ➢ This induces weight sharing across ground networks as their neurons are tied to the same template rules

• Different aggregation functions are used as neurons’ activations so as to reflect the (fuzzy) logic of disjunction, conjunction, and different forms of aggregative reasoning over relational patterns

• Stochastic Gradient Descend can be used for training
1. What are Lifted Relational Neural Networks?

2. Learning Predictive Categories with LRNNs
Learning Predictive Categories

We consider a following (learning) scenario with latent categories:

1. Entities
   - a) Have properties, b) Belong to categories
     - **Categories** largely determine belonging entities’ properties

2. Properties
   - a) Belong to entities, b) Belong to categories
     - **Categories** largely determine entities satisfying a property
Learning Predictive Categories

1. **Given**: a set of entities and corresponding lists of their properties

2. **Assumption**: there exists some latent hierarchy of categories that are predictive of their corresponding object’s properties
   - The hierarchy should allow for property inheritance
     - Similarly we induce latent hierarchy on properties

3. **Goal**: Learn suitable category structures from data
Encoding in LRNN

• Given input samples : \{1/0 \text{HasProperty}(e, p)\}

• Membership to categories : \(w_{ec} : \text{IsA}(e, c)\)

• Category hierarchy : \(w_{c1c2} : \text{IsA}(c_1, c_2)\)

• Category properties : \(w_{cecp} : \text{HasProperty}(c_e, c_p)\)

• Transitivity : \(w_{isa} : \text{IsA}(A, C) \leftarrow \text{IsA}(A, B), \text{IsA}(B, C)\)

• Categories determine their entities’ properties : \(w'_{cecp} : \text{HasProperty}(A, B) \leftarrow \text{IsA}(A, c_e), \text{IsA}(B, c_p), \text{HasProperty}(c_e, c_p)\)
Learning Setting

• We minimize MSE of the query atom neuron outputs and their targets \(\{1/0 \text{HasProperty}(e, p)\}\) via SGD.

• The activation functions used were:
  
  • Conjunction \(\land_{(b_1, \ldots, b_k)} = \text{sigm} \left( \sum_{i=1}^{k} b_i - k + b_0 \right)\)
  
  • Disjunction \(\lor_{(b_1, \ldots, b_k)} = \text{sigm} \left( \sum_{i=1}^{k} b_i + b_0 \right)\)
  
  • Aggregation \(\ast_{(b_1, \ldots, b_m)} = \max_i b_i\)

• We set up 2 level hierarchy with \([3, 2]\) hidden categories for both objects and properties.
Evaluation

- **Animals dataset** (https://alchemy.cs.washington.edu/data/animals)
  - 50 animals + 65 properties (e.g., large, smelly, strong, …)

- Predictive ability: AUC PR 0.8, AUC ROC 0.86

- Same as with second order Markov Logic Networks, reported in (Statistical Predicate Invention, Kok and Domingos, 2007)

- Which is related to the introduced model, while jointly clustering objects and relations
Embeddings of entities
Embeddings of properties
Outlook

- We have learned implicit similarity measure via latent category membership degrees

- We might also incorporate explicit similarities as
  \[ w_l : \text{HasProperty}(A, B) \leftarrow \text{HasProperty}(C, B), \text{Similar}(A, C, l) \]

  - Where \( l \) denotes some level of similarity, e.g. based on externally obtained embeddings

  - With that we might emulate 1-NN or kernel regression

- Also whole triples of (subject, \textit{predicate}, object) might be considered to learn soft categories of predicates, too
Conclusions

• LRNNs are a flexible framework to easily encode non-trivial SRL scenarios
  • e.g., a joint learning of predictive categories of entities and their properties
  • More complicated settings might be easily reached with just mild extensions of the template
    • e.g., semi-supervised learning, embeddings, etc.
  • We also plan for thorough comparison with MLNs and incorporation of LRNNs into NLP tasks pipelines
Thank you!

See “Lifted Relational Neural Networks” at arXiv.org for more details
Convolutional NN

\[
\begin{align*}
  w_0^{(1)} & : f_1 & \leftarrow & \text{left}(A), \text{mid}(B), \text{right}(C), \text{next}(A, B), \text{next}(B, C) \\
  w_1^{(2)} & : \text{left}(X) & \leftarrow & f_0(X) \\
  w_2^{(2)} & : \text{mid}(X) & \leftarrow & f_0(X) \\
  w_3^{(2)} & : \text{right}(X) & \leftarrow & f_0(X)
\end{align*}
\]