CS – 549

Paper Review – Neural Probabilistic Logic Programming in Discrete-Continuous Domains

Akhil Sai Chintala (934409906)

- **What is the problem discussed in the paper?**
  Neural-symbolic AI (NeSy) enables neural networks to utilize logically-based symbolic background information. It enables inferring from out-of-distribution data and promotes learning in the context of limited data. Logic and probability theory are combined with neural networks in probabilistic NeSy, enabling learning under uncertainty as well. Deep Probabilistic Programming (DPP) also includes neural network and continuous random variable models. DPP and NeSy differ because DPP has not been extended to support logical and relational reasoning. DeepSeaProbLog, a Neural Probabilistic Logic Programming (NPLP) language with support for discrete-continuous random variables and retention of logical and relational reasoning skills, is presented to fill this gap.

- **Why is it important?**
  Current probabilistic NeSy systems, like DeepProbLog, have a significant drawback in that they are restricted to discrete random variables with limited probability distributions. On the other hand, modeling and optimizing continuous probability distributions are strong suits of DPP. As a result, DeepSeaProbLog is presented, which integrates DPP methods into NeSy. With features like: DeepSeaProbLog generalizes a variety of current PLP languages. A weighted model integration (WMI)-based inference technique with a well-defined probabilistic semantics that can handle unbounded sample spaces
  DeepSeaProbLog is transformed into a differentiable, discrete-continuous NPLP language using a proved asymptotically unbiased gradient estimate for WMI.
  An experimental assessment demonstrating the adaptability and effectiveness of this method of thinking.
- **What are the main ideas of the proposed solution for the problem?**
  DeepSeaProbLog is a novel neural-symbolic probabilistic logic programming language that integrates hybrid probabilistic logic and neural networks. This integration is achieved by allowing arbitrary and differentiable probability distributions expressed in a modern DPP language while combining knowledge compilation and learning facilitation with the reparameterization trick and continuous relaxations of non-differentiable logic components. Inference is dealt with efficiently through approximate weighted model integration. DeepSeaProbLog is capable of intricate probabilistic modeling allowing for meaningful weak supervision while maintaining strong out-of-distribution performance. Experiments demonstrate how hybrid probabilistic logic can be used as a flexible structuring formalism for the neural paradigm that can effectively optimize and reuse neural components in different tasks. The implementation of DeepSeaProbLog utilizes Tensorflow Probability to sample from a continuous joint

probability distribution, effectively using knowledge compilation as a differentiable bridge between logical and probabilistic reasoning.

- **What are the shortcomings of the proposed solution?**
  The main limitation of DeepSeaProbLog is computational tractability which it inherits from probabilistic logic in general. Apart from this, other limitations include:

  The sampling strategy used remains ignorant of the comparison formulae that are approximated.
  More intricate inference strategies exist within the field of weighted model integration, yet they currently lack the differentiability property to be integrated in DeepSeaProbLog's gradient-based optimization.
  Performing successful joint inference and gradient-based learning under general comparisons is still open for improvement.
  Future advances in the field of approximate inference will be beneficial as exact knowledge compilation prevents the scaling of DeepSeaProbLog to larger problem instances.
  Avenue for scaling up DeepSeaProbLog inference using further continuous relaxation schemes, more specifically, replacing discrete random variables with relaxed categorical variables which might allow, for instance, to forego the knowledge compilation step while still being able to pass around training signals.