Neural Probabilistic Logic Programming in Discrete-Continuous Domains

Lennert De Smet[1][0000−0003−3136−0634], Pedro Zuidberg Dos Martires[2][0000−0001−5834−0188], Robin Manhaeve[1][0000−0001−9907−7486], Giuseppe Marra[1][0000−0001−5940−9562], Angelika Kimmig[1][0000−0002−6742−4057], and Luc De Raedt[1,2][0000−0002−6860−6303]

[1] Department of Computer Science, KU Leuven, Belgium
[2] Center for Applied Autonomous Systems, Örebro, Sweden

Neural-Symbolic AI (NeSy) [5] allows neural networks to exploit symbolic background knowledge in the form of logic by combining the reasoning power of logical representations with the learning capabilities of neural networks. Its advantages are many; learning converges more rapidly and performs better in the limited data regime. Additionally, NeSy models are robust and facilitate inference on out-of-distribution data [11, 15, 8]. One of the challenges of NeSy lies in combining logical symbols with continuous and differentiable neural representations. Such a combination has only been realised for discrete random variables by interpreting the outputs of neural networks as the weights of these variables. These weights can then be given either a fuzzy semantics [1, 7] or a probabilistic semantics [9, 17]. The latter is also used in neural probabilistic logic programming (NPLP), where neural networks parametrise probabilistic logic programs.

In contrast to existing probabilistic NeSy approaches, deep probabilistic programming (DPP) [14, 4] can capture models integrating continuous random variables. However, it is unclear whether DPP can be generalised to enable logical and relational reasoning. Moreover, reasoning on hybrid domains is crucial for safety-critical applications in robotics and reinforcement learning [16]. Hence, an important gap exists between DPP and NeSy as reasoning is a fundamental component of the latter. We contribute to closing this DPP-NeSy gap by introducing DeepSeaProbLog, an NPLP language with support for discrete-continuous random variables that retains logical and relational reasoning capabilities.

Syntax

DeepSeaProbLog is based on two fundamental concepts; the *neural distributional fact (NDF)* and the *probabilistic comparison formula (PCF)*. NDFs allow the definition of discrete-continuous random variables and comprise the interface between neural networks and logical symbols. For example, the uncertain location of an object `X` can be modelled as a normal distribution regressed by the network `regressor` via the expression `loc(X) ~ normal(regressor(X))`. PCFs then translate these random variables into Boolean statements usable in logical formulae. These statements can take the form of certain safety constraints such as `distance(loc(car1), loc(car2)) > 10`, expressing that `car1` and `car2` should be at least 10 meters apart.

Semantics, Inference and Learning

DeepSeaProbLog programs are used to compute the probability that a logical statement is true. This probability follows from the semantics of the DeepSeaProbLog program itself, which we base on the distribution semantics [13].

Inference in DeepSeaProbLog is based on weighted model integration (WMI) [3], which intuitively integrates over all models of a *satisfiability modulo theory (SMT)* formula [2] weighted by their probability. We formally prove that a DeepSeaProbLog program can be mapped onto an SMT formula and that the WMI of this formula yields the probability that the formula is true. As a result, DeepSeaProbLog inference is given by an expression of the form

$$\int \sum \prod \mathbb{1}(c(\boldsymbol{x}))\ p_{\Lambda}(\boldsymbol{x})\ \mathrm{d}\boldsymbol{x}, \tag{1}$$

where each $c(\boldsymbol{x})$ corresponds to a constraint defined by a PCF and every $p_{\Lambda}(\boldsymbol{x})$ is the probability density of a continuous random variable defined by a NDF. In practice, the integration is approximated by sampling values, exploiting DPP.

Unfortunately, this approach to inference introduces two obstacles for learning; the indicator functions in Equation (1) are not differentiable and sampling blocks the flow of gradients. The former is resolved by applying relaxations [10] and we formally prove that these relaxations lead to asymptotically unbiased gradients. The latter can be dealt with by applying the reparametrisation trick [12]. Together, these solutions turn DeepSeaProbLog into an end-to-end differentiable programming language. *More details can be found in the original publication* [6].

Experiments

The introduction of neurally parametrised continuous random variables allows DeepSeaProbLog to tackle problems beyond the scope of existing methods. One such problem is *distant object detection*, where objects need to be both located and classified without explicit supervision on the location. An example task of this nature is the detection of handwritten years from just the labels of the digit classes (Figure 1). Our results clearly illustrate that DeepSeaProbLog outperforms state-of-the-art DPP and non-probabilistic NeSy baselines.

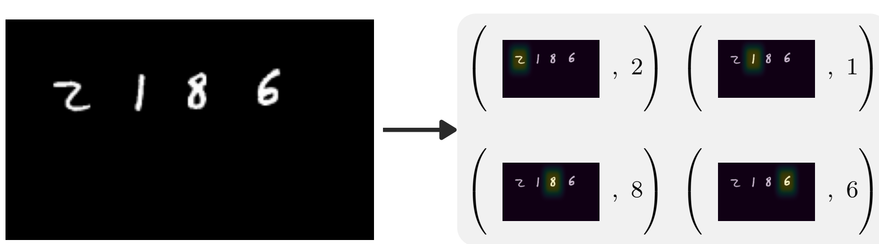

Fig. 1. Example of detecting a handwritten year from only its digit labels.

References

[1] Badreddine, S., Garcez, A.d., Serafini, L., Spranger, M.: Logic tensor networks. Artificial Intelligence (2022)
[2] Barrett, C., Tinelli, C.: Satisfiability modulo theories. In: Handbook of model checking. Springer (2018)
[3] Belle, V., Passerini, A., Van den Broeck, G.: Probabilistic inference in hybrid domains by weighted model integration. In: IJCAI (2015)
[4] Bingham, E., Chen, J.P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P., Horsfall, P., Goodman, N.D.: Pyro: Deep universal probabilistic programming. The Journal of Machine Learning Research (2019)
[5] De Raedt, L., Dumančić, S., Manhaeve, R., Marra, G.: From statistical relational to neural-symbolic artificial intelligence. In: IJCAI (2021)
[6] De Smet, L., Dos Martires, P.Z., Manhaeve, R., Marra, G., Kimmig, A., De Readt, L.: Neural probabilistic logic programming in discrete-continuous domains. In: Uncertainty in Artificial Intelligence. pp. 529–538. PMLR (2023)
[7] Diligenti, M., Gori, M., Saccà, C.: Semantic-based regularization for learning and inference. Artif. Intell. (2017)
[8] Evans, R., Grefenstette, E.: Learning explanatory rules from noisy data. Journal of Artificial Intelligence Research (2018)
[9] Manhaeve, R., Dumančić, S., Kimmig, A., Demeester, T., De Raedt, L.: Neural probabilistic logic programming in deepproblog. Artificial Intelligence (2021)
[10] Petersen, F., Borgelt, C., Kuehne, H., Deussen, O.: Learning with algorithmic supervision via continuous relaxations. NeurIPS (2021)
[11] Rocktäschel, T., Riedel, S.: End-to-end differentiable proving. NeurIPS (2017)
[12] Ruiz, F.J.R., Titsias, M.K., Blei, D.M.: The generalized reparameterization gradient. NeurIPS (2016)
[13] Sato, T.: A statistical learning method for logic programs with distribution semantics. In: ICLP (1995)
[14] Tran, D., Hoffman, M.D., Saurous, R.A., Brevdo, E., Murphy, K., Blei, D.M.: Deep probabilistic programming. In: ICLR (2017)
[15] Xu, J., Zhang, Z., Friedman, T., Liang, Y., Van den Broeck, G.: A semantic loss function for deep learning with symbolic knowledge. In: ICML (2018)
[16] Yang, W., Marra, G., Rens, G., Raedt, L.D.: Safe reinforcement learning via probabilistic logic shields. In: IJCAI 2023 (2023)
[17] Yang, Z., Ishay, A., Lee, J.: Neurasp: Embracing neural networks into answer set programming. In: IJCAI (2020)