Embedding Constraint Reasoning in Machine Learning to Build Generalist Systems

Yexiang Xue[1], Maxwell Jacobson[1], Nan Jiang[1], Maosen Zhang[2], Willem-Jan van Hoeve[3].

[1] Dept. of Computer Science, Purdue University. {yexiang, jacobs57, jiang631}@purdue.edu
[2] ByteDance. zhangmaosen.pku@bytedance.com
[3] Tepper School of Business, Carnegie Mellon University. vanhoeve@andrew.cmu.edu

Abstract

Building next generation generalist systems requires machine learning to capture data distribution and constraint reasoning to ensure structure validity. Nevertheless, effective approaches are lacking in bridging constraint satisfaction and machine learning. We propose **CO**nstraint **RE**asoning embedded structured learning (CORE), a scalable constraint reasoning and machine learning integrated approach for learning over structured domains. We propose to embed the reasoning module as a layer in the sequential neural networks for structured prediction and content generation. We evaluate CORE on several applications: vehicle dispatching service planning, if-then program synthesis, text2SQL generation, and constrained image generation. The proposed CORE module demonstrates superior performance over state-of-the-art approaches in all the applications. The structures generated with CORE satisfy 100% of the constraints, when using exact decision diagrams.

Introduction

The emergence of large-scale constraint reasoning and machine learning technologies have impacted virtually all application domains, including linguistics, operations, and vision. Constraint reasoning has traditionally been applied to building *prescriptive* models that generate solutions for strategic, tactical, or operational use (Xue, Choi, and Darwiche 2012). It requires a precise problem description and is usually difficult to be made flexible to the evolving data distributions. Machine learning, on the other hand, has been applied primarily to build *predictive* models, such as classifications or regressions (Bishop 2007). While the structure of a machine learning model (like a neural net) must be designed, the actual model parameters are learned via gradient descent. This gives machine learning models the flexibility to adapt to the evolving data distributions. Nevertheless, it is difficult to enforce constraints on the output of machine learning models. Many real-world applications are beyond the reach of constraint reasoning or machine learning alone.

In our recent line of work (Jiang et al. 2022; Xue and Hoeve 2019; Jacobson and Xue 2022), we have been focusing on structured prediction and content generation problems. Both learning problems require a tight integration of constraint reasoning and machine learning. Structured prediction and content generation have diverse application domains, ranging from natural language processing (Socher et al. 2013) to scene generation (Deng et al. 2021; Arad Hudson and Zitnick 2021).

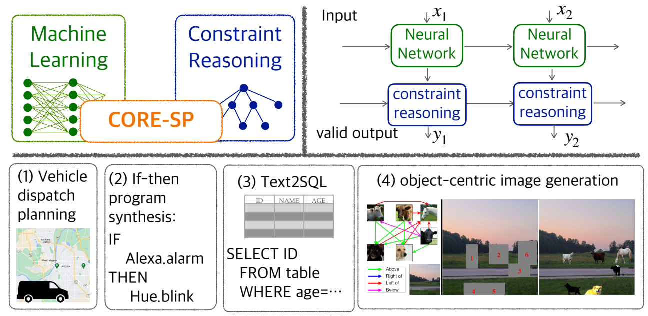

Figure 1: **(Up)** Our proposed CORE framework embeds constraint reasoning in into structured learning problems. At a high level, CORE (in blue colored box) is a fully differentiable layer that filters out the infeasible output from the structured output to ensure constraint satisfaction. **(Bottom)** We demonstrate the effectiveness of CORE on vehicle dispatching service, if-then program synthesis, Text2SQL generation, and constrained image generation tasks.

We propose **Co**nstraint **Re**asoning embedded **S**tructured **P**rediction (CORE), a scalable constraint reasoning and machine learning integrated approach for learning over the structured domains. The main idea is to augment structured predictive models with a constraint reasoning module that represents physical and operational requirements. See Figure 1 for our proposed CORE model, which integrates constraint reasoning and machine learning for all four applications. In the first three applications, we embed decision diagrams (Akers 1978) as a differentiable module into neural networks that can enforce constraint satisfaction of the outputs during training and testing. A decision diagram (DD) encodes each solution (an assignment of values to variables satisfying the constraints) as a path from the root to the terminal in the diagram. DD regards neural network predictions as the simulation of descending along a path in the decision diagram. DD filters out variable assignments from the neural network predictions that violate constraints. For the last application, we embed a tree search algorithm which carries out step-by-step reasoning of the spatial positions of every object into content generation. The spatial reasoning module decides the objects' positions following the output of a Recurrent Neural Network (RNN) in a process of iterative refinement. The RNN outputs the bounding boxes for each object to be generated. When determining one coordinate of the bounding box, the RNN iteratively halves the range of the coordinate until it is sufficiently small. During learning, the RNN is trained to learn implicit spatial knowledge, such as trees growing from the ground and birds flying in the sky. During inference, explicit constraints can be enforced by a forward-checking tree search algorithm, which removes all those position plans leading to constraint violation.

The applications we consider in this paper all require tight integration of constraint reasoning and machine learning. Our first application, *vehicle dispatching service planning*, recommends a route that satisfies daily service needs while meeting driver preferences. Historical data may reveal that the drivers do not follow common stylized objectives such as minimizing distance or time. Therefore, standard constraint reasoning tools, *e.g.*, solvers for the traveling salesman problem, cannot be applied. While we need machine learning to capture the drivers' implicit objective functions, pure machine learning-based approaches are insufficient because they often generate routes that violate delivery requests. Our second and third applications are *program synthesis from natural language*, which clearly requires machine learning to generate structured programs. Nevertheless, a pure learning approach cannot enforce the syntactic and grammar rules of those programs. Our last application is *constrained image generation*, which must generate feasible positions for a set of objects to be rendered in the image under user-defined positional constraints. A deep diffusion model can then inpaint the scene with background and position-organized objects, generating a "realistic" image. The pure deep generative model tends to fail on the number of objects, as well as the positional constraints of the objects, while our reasoning approach does not.

Our proposed CORE framework demonstrates superior performance against state-of-the-art approaches in all applications. First, the structures generated by CORE are better in constraint satisfaction. In vehicle service dispatching, all CORE generated routes are valid, while a conditional generative adversarial network (cGAN) without CORE generates on average less than 1% of valid routes when handling medium-sized delivery requests. For if-then program synthesis, the percentage of valid programs produced increased from 88% to 100% with the CORE module incorporated into the baseline. For Text2SQL, the percentage of valid SQL queries increased from 83.7% to 100% with CORE incorporated into the baseline on the testing set. For constrained image generation, the locations of the placed objects satisfy the given specifications. CORE also improves the learning performance of structured prediction models. We show that the routes generated by CORE better fulfill drivers' preferences than cGAN without CORE. In if-then program synthesis, CORE module leads to approximately 2.0% improvement in accuracy compared with the baseline. In Text2SQL generation, the CORE module improves around 4.2% in execution accuracy. In constrained image generation, CORE can generate realistic scene images with all the objects present and positioned at their natural locations. CORE also works well in zero-shot transfer learning: it generates good-quality scenes satisfying constraints unseen from the training set without retraining or fine-tuning.

References

Akers, S. B. 1978. Binary Decision Diagrams. *IEEE Trans. Computers*, 27(6): 509–516.

Arad Hudson, D.; and Zitnick, L. 2021. Compositional transformers for scene generation. *Advances in Neural Information Processing Systems*, 34: 9506–9520.

Bishop, C. M. 2007. *Pattern recognition and machine learning, 5th Edition*. Information science and statistics. Springer.

Deng, F.; Zhi, Z.; Lee, D.; and Ahn, S. 2021. Generative Scene Graph Networks. In *International Conference on Learning Representations*.

Jacobson, M. J.; and Xue, Y. 2022. Spatial Reasoning Network for Zero-shot Constrained Scene Generation. In *Under Review*.

Jiang, N.; Zhang, M.; van Hoeve, W.-J.; and Xue, Y. 2022. Constraint Reasoning Embedded Structured Prediction. *Journal of Machine Learning Research (JMLR)*. https://jmlr.org/papers/v23/21-1484.html.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *EMNLP*, 1631–1642. ACL.

Xue, Y.; Choi, A.; and Darwiche, A. 2012. Basing Decisions on Sentences in Decision Diagrams. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI)*. http://reasoning.cs.ucla.edu/fetch.php?id=123&type=pdf.

Xue, Y.; and Hoeve, W.-J. V. 2019. Embedding Decision Diagrams into GANs. In *Proc. of the Sixteenth International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR)*. https://link.springer.com/chapter/10.1007%2F978-3-030-19212-9_41.