Human-in-the-loop Learning for Probabilistic Programming

Sriraam Natarajan
University of Texas-Dallas

Phillip Odom
Georgia Institute of Technology

Tushar Khot
Allen Institute for AI

Kristian Kersting
TU Dortmund

Jude Shavlik
University of Wisconsin-Madison

Abstract

We present our BoostSRL system, a Java-based learning system that inductively learns probabilistic logic clauses from data. Our system is capable of learning different types of models, handling modeling of hidden data, learning with preferences from humans, scaling with large amounts of data by approximate counting and modeling temporal data. We review these capabilities briefly in this short paper.

1 BoostSRL system

Most learning methods inside Probabilistic Programming and the related Statistical Relational AI communities have focused on learning the parameters (probabilities/weights) given the program and the data. Our gradient boosted approach, presented in Figure 1 [9, 10], instead relies on the intuition that learning a set of weak partial programs (similar to ensemble methods in classical machine learning), and learns both the rules/programs and the parameters of the rules simultaneously. As shown in the figure, at a fairly high-level, the algorithm proceeds as follows: during every iteration, the gradients are computed for every example and these gradients become the regression values on the examples, a new logical (i.e., parameterized) regression tree is learned that fit these examples, the tree is added to the model and the process is repeated. Table 1 shows several rules from a probabilistic program learned for predicting the winner of a football game based on yards earned by each team, the number of turnovers as well as the season the games were played. The entire code base with full documentation is available at https://starling.utdallas.edu/software/boostsrl/. We have developed varied extensions for different models, distributions, learning settings and applications.

1.1 Multiple models:

Our framework and the system can learn different types of probabilistic programming models. It can learn Markov Logic Networks [2], undirected graphical models that employs the use of weighted logic clauses. It can learn Relational Dependency networks [10], cyclic lifted graphical models and can learn with hidden data [3]. In addition, this was employed in the context of sequential decision-making, *imitation learning*, and has been used to learn relational policies [7]. Finally, we extended our algorithm to learn over continuous time [17], the first of its kind in logical models.

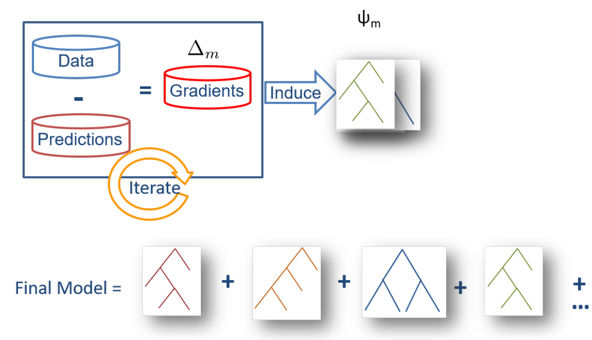

Figure 1. Relational Function Gradient Boosting: A boosted algorithm that iteratively fits the current error and adds it to the model. The approach is similar to that of the standard xg-Boost with one key difference, the gradients that are added i.e., the trees/partial programs are essentially first-order i.e., parameterized. Hence the successive learning of relational regression models allow for a compact representation of the learned concepts. This algorithm is the backbone on which several extensions have been developed - different distributions, different types of probabilistic programs, learning with human advice in the loop, scaling based on databases and finally, adapting them to several real tasks.

Table 1. Probabilistic program for predicting the winner of football games. Positive weights indicate that team T is more likely to win the game, while negative values indicate the opponent is more likely to win.

RULE	WEIGHT
winner(T, G) ← yds(T, G, High), turnovers(T, G, Low)	0.85
winner(T, G) ← yds(T, G, High), turnovers(T, G, High)	0.69
winner(T, G) ← yds(T, G, High), year(G, 1997)	0.45
winner(T, G) ← yds(Opponent,G,High)	-0.14

ProbProg'18, October 04–06, 2018, Boston, MA, USA
2018.

1.2 Multiple distributions:

Our system can learn different types of distributions - multinomials, exponential, Gaussian and Dirichlet distributions. We are currently extending the system to learn fully hybrid relational models.

1.3 Learning with human advice:

Most research inside probabilistic logic models either employed the full model and performed inference or learned only the parameters given the rules themselves. Our previous work on boosting instead learned the rules from the data. In essence, in our prior work, the humans were restricted to be "mere experts". Consequently, we have extended our system to include human advice in the form of *preferences* [12], *precision vs recall tradeoff* [16], *qualitative constraints* [14] and more recently *privileged information* [14].

1.4 Closing the loop:

While BoostSRL system can take human advice as input, the input is typically taken *before* learning occurs. Inspired by active learning, we have extended the system to solicit advice as needed. It explicitly computes the uncertainty in its model and solicits advice from the expert as needed [13]. Our BoostSRL system is capable of active advice seeking for the different types of logical models specified above.

1.5 Scaling:

Modern challenges require systems capable of handling large-scale data. The BoostSRL system has has been implemented on top of a relational database [4]. Malec et al. [4] demonstrate the efficiency gains of combining databases with effective background knowledge in inductive logic programming. Das et al. [1] introduce an efficient approximate counting technique which can be used as a part of our BoostSRL system when learning MLNs by converting the partial programs into a graph. We have recently generalized this work to now learn hypergraphs that allows for effectively learning multi-arity predicates in an efficient manner.

1.6 Applications:

As part of the BoostSRL system, we have developed a NLP pipeline that have been employed for relation extraction previously [5, 11, 15]. The key aspect of this pipeline is that it reads text documents as inputs, runs it through Stanford NLP pipeline, creates corresponding logic facts and rule base thus making it ready for learning logical models. This pipeline has been used in combination with BoostSRL to identify adverse drug effects from PubMed abstracts [5] and knowledge base relations [15].

Medical applications: We used the BoostSRL system to to classify patients into normal, mild cognitive impairment (onset of Alzheimer's that is hard to detect) and Alzheimer's using relational features from MRI scans [6]. We show part of the program in Table 2, where the system has identified special regions and features of the brain associated with Alzheimer's. We have also used our system to predict coronary artery calcification levels based on historical data [8].

Table 2. Part of the probabilistic program for predicting Alzheimer's based on regions in the MRI scan. The system identifies important regions of the brain such as the hippocampus and important features of the MRI image such as the cerebrospinal fluid (CSF) and grey matter (GM) levels.

RULE	WEIGHT
alz(B) ← CSF(B, Left parahippocampal gyrus) > 0.19, CSF(B, Lobule IX of vermis) > 0.13	0.57
alz(B) ← CSF(B, Left parahippocampal gyrus) > 0.19, CSF(B, Lobule IX of vermis) ≤ 0.13 CSF(B, Left hippocampus) > 0.37	0.70
. . .	
alz(B) ← CSF(B, Left parahippocampal gyrus) ≤ 0.19, CSF(B, Right superior frontal gyrus) ≤ 0.25 GM(B, Left hippocampus) ≤ 0.56	0.68

2 Conclusion

BoostSRL is a versatile system that is capable of learning different types of models of multiple distributions. Its ability to learn with different types of human knowledge and effectively scale up to large data enable it to be applicable to many challenging tasks.

3 Acknowledgements

The authors acknowledge the contributions of the other students and collaborators in the lab who have developed this system. In no particular order, we thank, Gautam Kunapuli, Ameet Soni, Shuo Yang, Mayukh Das, Devendra Dhami, Nandini Ramanan, Srijita Das, Navdeep Kaur, Alex Hayes, Kaushik Roy, Dileep Viswanathan, Anurag Wazalwar, Vishal Bangera, and Raksha Kumaraswamy. We also acknowledge the support of DARPA, AFOSR, NSF, AFRL and ARO for their support. Kristian Kersting acknowledges the support by the German Science Foundation project âĂIJCAML: Argumentative Machine LearningâĂİ (KE1686/3-1) as part of the SPP 1999 (RATIO). Finally, we acknowledge the contributions of all the co-authors of the papers that have significantly improved the applicability of the system.

References

[1] Mayukh Das, Yuqing Wu, Tushar Khot, Kristian Kersting, and Sriraam Natarajan. 2016. Scaling Lifted Probabilistic Inference and Learning Via Graph Databases. In *SIAM International Conference on Data Mining (SDM)*.

[2] Tushar Khot, Sriraam Natarajan, Kristian Kersting, and Jude Shavlik. 2011. Learning markov logic networks via functional gradient boosting. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*. IEEE, 320–329.

[3] Tushar Khot, Sriraam Natarajan, Kristian Kersting, and Jude Shavlik. 2015. Gradient-based Boosting for Statistical Relational Learning: The

Markov Logic Network and Missing Data Cases. *MLJ* (2015).

[4] Marcin Malec, Tushar Khot, James Nagy, Erik Blask, and Sriraam Natarajan. 2017. Inductive Logic Programming Meets Relational Databases: Efficient Learning of Markov Logic Networks. In *International Conference on Inductive Logic Programming (ILP)*. Springer International Publishing, 14–26.

[5] Sriraam Natarajan, Vishal Bangera, Tushar Khot, Jose Picado, Anurag Wazalwar, Vitor Santos Costa, David Page, and Michael Caldwell. 2016. Markov Logic Networks for Adverse Drug Event Extraction from Text. *Knowledge and Information Systems* (2016).

[6] Sriraam Natarajan, Saket Joshi, Baidya Saha, Adam Edwards, Tushar Khot, Elizabeth Moody, Kristian Kersting, Christopher Whitlow, and Joseph Maldjian. 2013. Relational Learning helps in Three-way Classification of Alzheimer Patients from Structural Magnetic Resonance Images of the Brain. *International Journal of Machine Learning and Cybernetics* (2013).

[7] Sriraam Natarajan, Saket Joshi, Prasad Tadepalli, Kristian Kersting, and Jude Shavlik. 2011. Imitation Learning in Relational Domains: A Functional-Gradient Boosting Approach. In *IJCAI*. 1414–1420.

[8] Sriraam Natarajan, Kristian Kersting, Edward Ip, David Jacobs, and Jeffrey Carr. 2013. Early Prediction of Coronary Artery Calcification Levels Using Machine Learning. In *Innovative Applications in AI*.

[9] Sriraam Natarajan, Kristian Kersting, Tushar Khot, and Jude Shavlik. 2015. *Boosted Statistical Relational Learners: From Benchmarks to Data-Driven Medicine*. Springer.

[10] Sriraam Natarajan, Tushar Khot, Kristian Kersting, Bernd Gutmann, and Jude Shavlik. 2012. Gradient-based Boosting for Statistical Relational Learning: The Relational Dependency Network Case. *MLJ* (2012).

[11] Sriraam Natarajan, Ameet Soni, Anurag Wazalwar, Dileep Viswanathan, and Kristian Kersting. 2016. *Deep Distant Supervision: Learning Statistical Relational Models for Weak Supervision in Natural Language Extraction*. LNAI.

[12] Phillip Odom, Tushar Khot, Reid Porter, and Sriraam Natarajan. 2015. Knowledge-Based Probabilistic Logic Learning. In *AAAI*.

[13] Phillip Odom and Sriraam Natarajan. 2016. Actively Interacting with Experts: A Probabilistic Logic Approach. In *ECML*.

[14] Phillip Odom and Sriraam Natarajan. 2018. Human-Guided Learning for Probabilistic Logic Models. *Frontiers in Robotics and AI journal special issue on Statistical Relational Artificial Intelligence* (2018).

[15] Ameet Soni, Dileep Viswanathan, Jude Shavlik, and Sriraam Natarajan. 2016. Learning Relational Dependency Networks for Relation Extraction. In *ILP*.

[16] Shuo Yang, Tushar Khot, Kristian Kersting, Gautam Kunapuli, Kris Hauser, and Sriraam Natarajan. 2014. Learning from Imbalanced Data in Relational Domains: A Soft Margin Approach. In *ICDM*.

[17] Shuo Yang, Tushar Khot, Kristian Kersting, and Sriraam Natarajan. 2016. Learning Continuous-Time Bayesian Networks in Relational Domains: A Non-Parametric Approach. In *AAAI*.