Discriminative Learning of Sum-Product Networks

Robert Gens **Pedro Domingos**
Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195-2350, U.S.A.
{rcg,pedrod}@cs.washington.edu

Discriminatively-trained probabilistic models often outperform their generative counterparts on challenging tasks in computer vision and NLP. Conditional random fields have been a predominant approach, but adapting them to richer structures further complicates learning with approximate inference [4]. Deep Networks have typically required both a generative and discriminative objective involving approximate inference. The discriminative objective is usually reflected in the use of backpropagation through softmax layers or support vector machines over network variables. The Sum-Product Network is a promising new deep architecture that can perform fast, exact inference on a representation that is broader than existing tractable graphical models [5].

For the first time, we combine the advantages of SPNs with the advantages of discriminative learning. We demonstrate the conditions under which an SPN represents the conditional partition function and provide a discriminative training algorithm. In this setting, SPN nodes that exclusively cover variables conditioned upon need not be consistent or complete. This admits a larger class of discriminative SPNs with speedy inference.

We define an SPN that models the joint probability of three disjoint sets of hidden, query, and given variables: H, Y, and X respectively. The conditional probability of the query given the data is naturally $P(Y|X) = \frac{P(Y,X)}{P(X)} = \frac{\sum_h P(h,Y,X)}{\sum_{h,y} P(h,y,X)}$. Each summation only requires a single linear-time evaluation of the SPN, in which we set all indicators for the marginalized variables to one.

Partial derivatives of this objective are just as easy as in the generative case. In previous work, we tried using backpropagation and "soft" expectation maximization to maximize the likelihood of the training data. We noted, however, that learning from the expectation over all possible complete subcircuits suffered from gradient diffusion [5, 1]. This led us to use "Hard" EM, which provided the best results at the time.

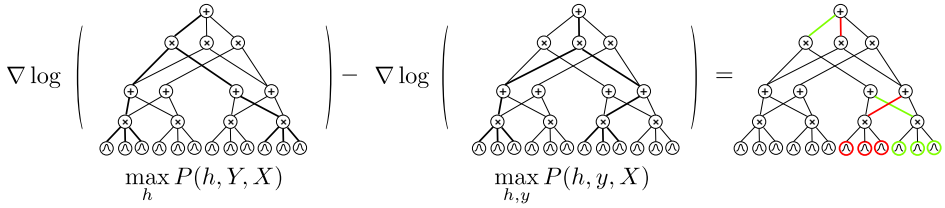
To optimize the conditional log likelihood, we use the same approximation in a technique we term "hard" gradient descent. For the purpose of this discussion, weights only appear on the edges from sum nodes to their children.

$$\begin{aligned}\frac{\partial}{\partial w}\log P(Y|X) &= \frac{\partial}{\partial w}\log\sum_h P(h,Y,X) - \frac{\partial}{\partial w}\log\sum_{h,y} P(h,y,X) \\ &\approx \frac{\partial}{\partial w}\log\max_h P(h,Y,X) - \frac{\partial}{\partial w}\log\max_{h,y} P(h,y,X)\end{aligned}$$

Finding the assignments that maximize the value of an SPN is also a linear-time operation. In a top-down pass we follow the winning child of a sum node and all children of a product node. This maximization produces a complete sub-circuit, a monomial in the network polynomial [3]. Since there are two maximizations in this discriminative objective, we have the gradient of the difference of two monomials. The partial derivative of a weight therefore depends on whether that weight appears in one, both, or none of the monomials.

$$\frac{\partial}{\partial w}\log P(Y|X) \quad \approx \quad \begin{cases} \frac{1}{w} & : w \in \underset{h}{\operatorname{argmax}}\, P(h,Y,X) \wedge w \notin \underset{h,y}{\operatorname{argmax}}\, P(h,y,X) \\ -\frac{1}{w} & : w \notin \underset{h}{\operatorname{argmax}}\, P(h,Y,X) \wedge w \in \underset{h,y}{\operatorname{argmax}}\, P(h,y,X) \\ 0 & : \text{otherwise} \end{cases}$$

The first two conditions correspond to a weight being part of only complete subcircuit. The last condition applies to when a weight is part of both or neither subcircuit. The subcircuit $\underset{h}{\operatorname{argmax}}\ P(h,Y,X)$ is the result of infering MAP assignments to the hidden variables when we provide ground-truth values for the labels. Likewise, $\underset{h,y}{\operatorname{argmax}}\ P(h,y,X)$ denotes the subcircuit for when we must infer both label and hidden variables, corresponding to the model at test time. We illustrate this objective below with bold lines denoting the subcircuit resulting from hard inference, green and red indicating positive and negative derivatives respectively.



We note that this learning rule is similar to voted perceptron [2]. To see this more clearly, one would just parameterize weights on sum nodes as e^w instead of w. This would change the above update rule from $\{\frac{1}{w}, -\frac{1}{w}, 0\}$ to $\{1, -1, 0\}$.

We are applying discriminative SPNs to several promising visual applications that will benefit from fast inference over millions of variables. Since SPNs can represent probabilistic context-free grammars, they are well-suited for learning image grammars. Discriminative SPNs are also appropriate for challenging classification tasks that rely on context. We are currently applying discriminative SPNs to Cifar-10, the Stanford Background dataset, emotion classification, etc. . We are investigating both static and learned architectures for these problems.

References

[1] Y. Bengio. Learning deep architectures for ai. *Foundations and Trends® in Machine Learning*, 2(1):1–127, 2009.

[2] M. Collins and N. Duffy. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 263–270. Association for Computational Linguistics, 2002.

[3] A. Darwiche. A differential approach to inference in bayesian networks. *Journal of the ACM (JACM)*, 50(3):280–305, 2003.

[4] A. Kulesza, F. Pereira, et al. Structured learning with approximate inference. *Advances in neural information processing systems*, 20:785–792, 2007.

[5] H. Poon and P. Domingos. Sum-product networks: A new deep architecture. In *Proc. 12th Conf. on Uncertainty in Artificial Intelligence*, pages 337–346, 2011.

Topic: learning algorithms **Preference:** oral