VISUALIZING GENERATIVE SUM-PRODUCT NETWORKS ON IMAGE RECONSTRUCTION

Renato L. Geh

University of São Paulo
Institute of Mathematics and Statistics
Rua do Matão, 1010 - São Paulo, SP, Brazil, 05508-090

ABSTRACT

Sum-Product Networks (SPNs) are fairly recent deep tractable probabilistic graphical models that are able to answer exact queries in linear time. Although there have been many advancements in practical problems, there is an absence in literature of visualizations on how SPNs represent learned data. In this paper we show how two structure learning algorithms can heavily impact on how SPNs treat data, particularly in the domain of image reconstruction. We show two coloring techniques to visualize sum and product nodes through their scopes. We then apply both techniques to generative SPNs learned from two distinct learning methods.

Index Terms— Sum-product networks, probabilistic graphical models, visualization, image reconstruction

1. INTRODUCTION

Image reconstruction is the task of accurately predicting, guessing and completing missing elements from an image. Density estimators that model a joint probability distribution can achieve this by learning the features of similar images and finding the valuation that most adequately fits the incomplete image. However, classical density estimators, such as Probabilistic Graphical Models (PGMs), suffer from exact inference intractability in the general case. This leads to approximate prediction and representation, as learning in PGMs often requires the use of inference as a subroutine.

Sum-Product Networks (SPNs) [1] are fairly recent tractable PGMs capable of representing distributions as a deep network of sums and products. Most importantly, SPNs are capable of exact inference in time linear to its graph's edges. There have been many advances on SPNs in the image domain, such as image classification and reconstruction [2, 3, 4], image segmentation [5] and activity recognition [6, 7, 8]. However, there have been little effort [9] so far to explore SPNs' semantics and representation power.

In this paper, we provide visualizations on SPNs learned from two structure learning algorithms. We present two techniques to perform this task. These techniques rely on a couple of properties SPNs must follow in order to correctly represent a probability distribution, and are highly dependent on the graph's structure. We first give a short background review of SPNs, relevant properties and scope definition. We follow this with an explanation on how we achieved the visualizations shown in this article. Finally, we show results and provide a conclusion of our findings.

2. BACKGROUND

An SPN can be seen as a DAG with restrictions with respect to its node types and weighted edges. Let n be a graph node. The set of nodes $\text{Pa}(n)$ and $\text{Ch}(n)$ are the parents and children of n. A weighted edge $i \to j$ is denoted by $w_{i,j}$.

Definition 1. *A sum-product network (SPN) is a directed acyclic graph. A node n of an SPN can either be a:*

1. *sum, where its value is given by $v_n = \sum_{j \in Ch(n)} w_{n,j} v_j$;*
2. *product, where its value is given by $v_n = \prod_{j \in Ch(n)} v_j$;*
3. *probability distribution, whose value is its probability of evidence.*

In this paper, we assume that all SPN leaves (i.e. node type 3) are tractable univariate distributions, that is, computing its mode or partition function takes constant time. The scope of a node $\text{Sc}(n)$ is the union set of the scope of its children.

Definition 2 (Completeness)**.** *An SPN is complete iff every child of a sum node has the same scope as its siblings.*

Definition 3 (Decomposability)**.** *An SPN is decomposable iff every child of a product node has disjoint scope with its siblings.*

A complete and decomposable SPN correctly computes the probability of evidence of the modeled distribution. An SPN that correctly represents a probability distribution is said to be valid. In fact, a complete and consistent (i.e. no two children of an SPN node have contradicting variable values) SPN is sufficient (though not necessary) for validity [1]. However,

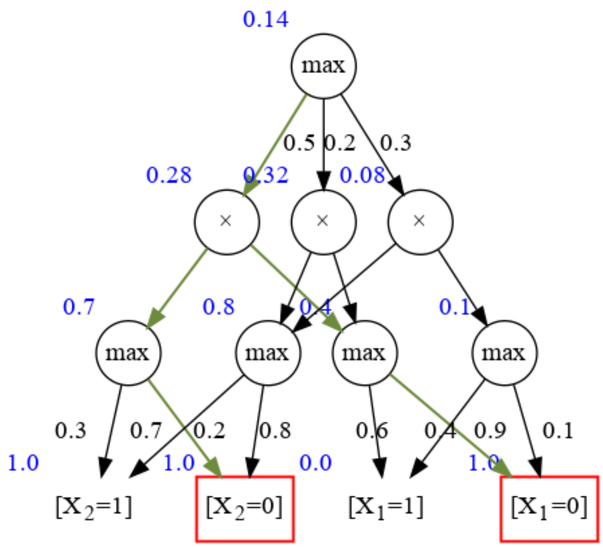

Fig. 1. Finding the MPE of an SPN given $\mathbf{X} = \{X_1 = 0\}$.

learning decomposable SPNs is easier, and it has been shown that decomposability is as expressive as consistency [10].

Let $\mathbf{X} = \{X_1 = x_1, X_2, = x_2, \ldots, X_n = x_n\}$ be a valuation and S an SPN. The value of S is the value of its root, and is denoted by $S(\mathbf{X})$. Inference in SPNs is done through a bottom-up evaluation. The value of a leaf node n is the probability of $\mathbf{X}$. If $\text{Sc}(n) \not\subset \mathbf{X}$, then n's value is the distribution's mode.

Finding the $\arg\max_{\mathbf{x}} S(\mathbf{X} = \mathbf{x})$, also called the Most Probable Explanation (MPE), of an SPN has been shown to be NP-hard [10, 11, 12]. Image reconstruction can be seen as an application of MPE, where each variable is a pixel, and values are pixel colors. Finding the MPE, and thus the reconstruction of an image given some initial evidence consists of finding the pixel values that are most likely to fit the model. Given that finding the exact valuation that maximizes the model is hard, we instead use an approximate method proposed in [1] called the Max-Product algorithm.

The Max-Product algorithm consists of replacing the original SPN with a Max-Product Network (MPN). An MPN of an SPN is simply the SPN with its sum nodes replaced with max nodes. The value of a max node is the maximum weighted child. Finding an MPE approximation on an MPN is done through a bottom-up evaluation similar to an SPN. Once all nodes have been computed, a top-down traversal is done, finding the max paths in the graph by choosing only the max edge in a max node and traversing all edges in product nodes, as shown in Figure 1.

3. VISUALIZING SPNS

Visualization in SPNs can be done through an analysis of the SPN's scope and structure. The definition of completeness lends itself naturally to an interpretation of sum nodes as layers of mixture models. A possible intuition for this interpretation is that sum nodes model latent variables in charge of explaining similar interactions between variables. Decomposability, on the other hand, models independence between sets of variables.

4. RESULTS

5. CONCLUSION

6. REFERENCES

[1] Hoifung Poon and Pedro Domingos, "Sum-product networks: A new deep architecture," *Uncertainty in Artificial Intelligence*, vol. 27, 2011.

[2] Robert Gens and Pedro Domingos, "Learning the structure of sum-product networks," *International Conference on Machine Learning*, vol. 30, 2013.

[3] Aaron Dennis and Dan Ventura, "Learning the architecture of sum-product networks using clustering on variables," *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[4] Robert Gens and Pedro Domingos, "Discriminative learning of sum-product networks," *Advances in Neural Information Processing Systems*, pp. 3239–3247, 2012.

[5] Zehuan Yuan, Hao Wang, Limin Wang, Tong Lu, Shivakumara Palaiahnakote, and Chew Lim Tan, "Modeling spatial layout for scene image understanding via a novel multiscale sum-product network," *Expert Systems with Applications*, vol. 63, pp. 231 – 240, 2016.

[6] M. R. Amer and S. Todorovic, "Sum-product networks for modeling activities with stochastic structure," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 1314–1321.

[7] M. R. Amer and S. Todorovic, "Sum product networks for activity recognition," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, April 2016, vol. 38, pp. 800–813.

[8] J. Wang and G. Wang, "Hierarchical spatial sum–product networks for action recognition in still images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 1, pp. 90–100, Jan 2018.

[9] Antonio Vergari, Nicola Di Mauro, and Esposito Floriana, "Visualizing and understanding sum-product networks," *Machine Learning*, 08 2016.

[10] Robert Peharz, Sebastian Tschiatschek, Franz Pernkopf, and Pedro M. Domingos, "On theoretical properties of sum-product networks," in *AISTATS*, 2015.

[11] Jun Mei, Yong Jiang, and Kewei Tu, "Maximum a posteriori inference in sum-product networks," in *AAAI*, 2018.

[12] Diarmaid Conaty, Denis D. Maua, and Casio P. de Campos, "Approximation complexity of maximum a posteriori inference in sum-product networks," in *Proceedings of The 33rd Conference on Uncertainty in Artificial Intelligence*. 8 2017, AUAI.