

# Automatic Learning of Sum-Product Networks

Renato Lui Geh (student), Denis Deratani Mauá (advisor)  
Institute of Mathematics and Statistics, University of São Paulo  
{renatolg, ddm}@ime.usp.br



IME-USP

## Introduction

Sum-product networks (SPNs) are probabilistic graphical models capable of representing tractable probability distributions with a great number of variables. In the last decade, SPNs have achieved impressive results in various fields, such as protein folding, signal modelling, image classification and completion, activity recognition and natural language.

Graphically, SPNs can be seen as DAGs whose leaves are tractable univariate probability distributions and internal nodes are either weighted sums or products. Its great representational power is given by its deep architecture, allowing for deep compact networks to represent the same family of functions as an exponential shallow one [1]. Moreover, computing exact inference in SPNs and extracting the approximate MAP states of the network is done in linear time to the number of the graph's edges, meaning that parameter learning is also tractable.

Despite these promising results, at present there are very few SPN inference and learning codebases publicly available, and no well maintained, documented and updated SPN library. In addition, there has been no comparative study on different SPN learning methods yet. This project sought to develop a free, open-source inference and learning SPN library, and to compare three different state-of-the-art SPN learning algorithms in the domain of image classification and completion.

## Materials and Methods

The Poon-Domingos architecture [4], Dennis-Ventura structure algorithm [2] and Gens-Domingos schema [3] were implemented as part of the GoSPN library (available at <https://github.com/RenatoGeh/gospn/>). The library, written in Go, also contains functions for finding the exact probability of evidence of some valuation, computing an approximation of the MAP probability and states, and extracting useful structural properties of a network, such as completeness, decomposability and scope.

We used four datasets for image classification and completion. Two datasets, DigitsX and MNIST, are handwritten digits dataset, with the former containing only one handwriting, and the latter 500. The Caltech-101 dataset contains images of 101 different object categories. We took only three categories (bike, cars and faces) due to memory and time constraints. The last dataset, Olivetti, contains 400 face images of 40 different people, with 10 images each.

## Development

During training, the Poon-Domingos algorithm struggled with the time and memory constraints established, often exceding either or both. The Gens-Domingos and Dennis-Ventura algorithms had similar running times for training, with both taking less than eight hours in all datasets. Tests were done in an Intel i7–4500U 1.8GHz with 16GB RAM, and were limited to 24 hours of training. Inference was very fast, taking about a second to compute the approximate MAP states.

Training was done without any additional feature extraction. SPNs were trained based solely on pixel values. For the Poon-Domingos and Dennis-Ventura structures, each pixel was represented by a mixture of gaussians with  $g$  components. Each pixel was then divided into  $g$  quantiles, with each component's mean and standard deviation set to that of its corresponding quantile. The Gens-Domingos algorithm represented each pixel as a discrete multinomial distribution.

Classification was done by taking a percentage  $p$  of the dataset as training set, and  $1 - p$  as test set. Training and test sets were balanced, i.e. every class had the same number of instances each. Predicting labels was done by finding the MPE of the classification variable through approximate MAP state extraction. We speculate that better results could be achieved by computing the exact probability of evidence of each possible valuation instead of the MAP way, albeit with longer inference time.

For image completion, half of the image was given as evidence to the model. The other half was then generated by the SPN by querying for the most probable variable valuations given the evidence.

## Results

Due to the issues mentioned earlier, the Poon-Domingos model either did not complete training or had unsatisfactory results. In this poster, we show only results achieved by the Dennis-Ventura (DV) and Gens-Domingos (GD) algorithms.

$p = 0.1$	DV	GD	$p = 0.2$	DV	GD	$p = 0.3$	DV	GD
DigitsX	92.85	91.27	DigitsX	98.57	96.78	DigitsX	99.18	96.93
Caltech	78.58	77.40	Caltech	78.49	85.00	Caltech	79.88	80.28
Olivetti	83.78	2.50	Olivetti	74.88	2.50	Olivetti	89.93	84.28
$p = 0.4$	DV	GD	$p = 0.5$	DV	GD	$p = 0.6$	DV	GD
DigitsX	98.81	98.09	DigitsX	99.42	97.14	DigitsX	99.28	97.85
Caltech	79.88	86.11	Caltech	81.38	88.66	Caltech	81.35	90.00
Olivetti	89.93	91.25	Olivetti	89.93	95.50	Olivetti	97.50	98.75
			MNIST	77.85	81.55			
$p = 0.7$	DV	GD	$p = 0.8$	DV	GD	$p = 0.9$	DV	GD
DigitsX	98.57	97.61	DigitsX	93.33	92.66	DigitsX	88.75	86.25
Caltech	75.45	92.22	Caltech	74.78	90.00	Caltech	75.75	84.84
Olivetti	92.89	81.93	Olivetti	50.00	81.59	Olivetti	60.93	100.0

Table 1: Classification accuracy (in %).

Image completion was done on the Olivetti dataset. The gray half of the image was given as evidence, and the other half in green is the completion generated by the model. The completion on the left was done on an SPN trained with the Gens-Domingos algorithm. The one on the right was generated by a Dennis-Ventura trained SPN.



Figure 1: Image completion on the Olivetti dataset.

## Conclusions

We achieved good classification results without any feature extraction on different applications, such as handwritten digit classification, object identification and face recognition. On the completion task, the algorithms were able to identify key features, such as nose, eyes and mouth on the Olivetti dataset. Both learning and inference code were documented and made available as part of the free open-source GoSPN library.

## References

- [1] Olivier Delalleau and Yoshua Bengio. “Shallow vs. Deep Sum-Product Networks”. In: *Advances in Neural Information Processing Systems 24 (NIPS 2011)* (2011).
- [2] Aaron Dennis and Dan Ventura. “Learning the Architecture of Sum-Product Networks Using Clustering on Variables”. In: *Advances in Neural Information Processing Systems 25* (2012).
- [3] Robert Gens and Pedro Domingos. “Learning the Structure of Sum-Product Networks”. In: *International Conference on Machine Learning* 30 (2013).
- [4] Hoifung Poon and Pedro Domingos. “Sum-Product Networks: A New Deep Architecture”. In: *Uncertainty in Artificial Intelligence* 27 (2011).