THE POON-DOMINGOS PARAMETER LEARNING ALGORITHM FOR IMAGE COMPLETION AND CLASSIFICATION ON SUM-PRODUCT NETWORKS

Renato Lui Geh
Computer Science
Institute of Mathematics and Statistics
University of São Paulo
renatolg@ime.usp.br

Abstract. In this document we describe the Poon-Domingos [PD11] parameter learning algorithm for image classification and completion.

1. Structure

The Poon-Domingos algorithm uses a fixed structure and then learns the weights through generative learning. We first give an overview on how to build the structure given an image and then provide a pseudo-code algorithm for building such structure.

1.1. Overview

The Poon architecture models a probability distribution over a set of images. It is constructed by taking all possible rectangular axis-aligned regions in the image and assigning product nodes to each of these regions. Two sum nodes are then added as children for each of these regions, representing all the possible pairings of subregions in each region determined by the axis-aligned division set in the previous step. We then add the product nodes to a single sum node that represents the undivided original area. We then recursively apply the same steps on each sum node we constructed this way, taking that sum node as the new root of the sub-SPN. The Poon structure accepts different multiple resolution levels. At every region splitting, we consider a step r that indicates how fine the granularity is for the architecture.

1.2. Definitions and properties

Let us organize in a clearer way what we have extracted from [DV12].

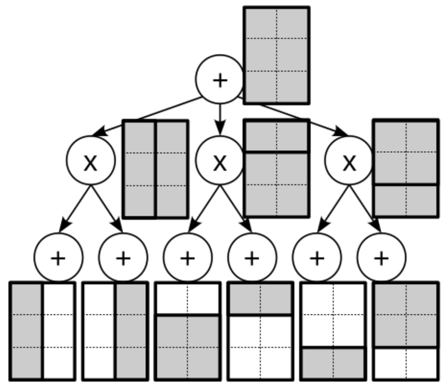

FIGURE 1. The Poon architecture with $r = 1$ resolution. At each product node division, we consider an r length division on each axis. Since $r = 1$ in this case, we have 3 product nodes, with each of their two possible subregions represented by sum nodes. Note how product nodes are always decomposable and sum nodes are always complete.

Definition 1.1 (Region). *A region π is a product node. Graphically, it represents an axis-aligned rectangular region of the image. We shall denote an SPN $S(\cdot)$ rooted at a region as $\pi(\cdot)$.*

Definition 1.2 (Subregion). *A subregion σ is a sum node. Given a region R, the children of R are the two possible rectangles that compose R. We shall denote an SPN rooted at a subregion as $\sigma(\cdot)$.*

The idea behind the Poon structure algorithm is to recursively take a rectangular subarea of an image and divide it into all k possible regions given a granularity r, with each region having two children (subregions) that represent the possible pairings of each region. We then recurse through each of these subregions.

Let $I(x_0, y_0, x_1, y_1)$ be the area of the image we are to create the structure for, with (x_0, y_0) and (x_1, y_1) being the top left and bottom right positions in the image. The entire image is given by $I(0, 0, w, h)$, where w, h are the width and height of the image respectively.

A subregion σ implicitly represents an area $I(x_0, y_0, x_1, y_1)$, whilst each region of σ is a possible subdivision of I by drawing a line, either horizontally or vertically, through one of the axis of I, partitioning it into two subregions.

We can clearly see that, for a subregion σ, there are $x_1 - x_0$ possible vertically divided regions and $y_1 - y_0$ horizontally divided regions, bringing the total to $(x_1 + y_1) - (x_0 + y_0)$ possible product nodes as children of σ for $r = 1$. For the general case, we can clearly see that we have $\lceil (x_1 - x_0)/r \rceil + \lceil (y_1 - y_0)/r \rceil$ possible regions.

1.3. Algorithm

The structure algorithm is recursive. It takes as parameters a sum node S as root of the SPN, the (x_0, y_0) top-left position of the subregion S relative to the original complete image, the (x_1, y_1) bottom-right position of the subregion S, the resolution granularity step k and a dataset $\mathcal{D}$ where $\mathcal{D}[X]$ is the set of instances of variable X.

Algorithm 1 `GenerateDenseSPN`

Input Root sum node S
Input Top-left position $p_0 = (x_0, y_0)$ of the underlying image area of S
Input Bottom-right position $p_1 = (x_1, y_1)$ of the underlying image area of S
Output A dense SPN structure

References

[DV12] 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).

[PD11] Hoifung Poon and Pedro Domingos. “Sum-Product Networks: A New Deep Architecture”. In: *Uncertainty in Artificial Intelligence* 27 (2011).