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. In this document we assume instances as images. However, the Poon-Domingos structure allows for any object with local dependencies.

1.1. Overview

The Poon-Domingos structure models a probability distribution over a set of variables with local dependencies. On the plain, one could argue it models rectangular neighborhoods for each point in the space. In the article [PD11], Poon and Domingos use images as a dataset, with dependencies being rectangular pixel neighborhoods. Images are an example of local dependencies, since a pixel has possible dependencies with their neighbors.

Dennis and Ventura explain an intuition of how the Poon-Domingos structure algorithm works [DV12]. We expand on this intuition, giving insights on how such an algorithm is built and showing a pseudo-code visualization of it. Once we have shown how to build the SPN structure, we describe generative learning through gradient descent, and later expectation-maximization.

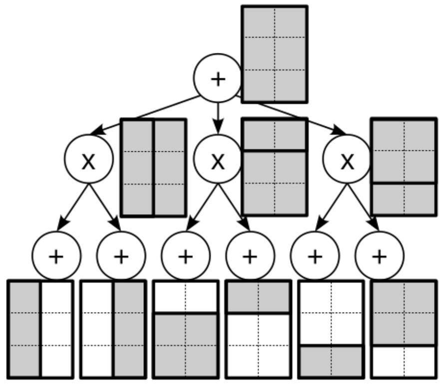

FIGURE 1. The Poon architecture with $r = 1$ resolution and $k = 1$ sum nodes per region on a 2×3 image. At each r resolution axis-aligned rectangular decomposition, we create k sum nodes.[DV12]

1.2. Definitions and properties

Definition 1.1 (Region)**.** *A Region $\mathcal{R}$ is a rectangular part of an image. Let $p_0 = (x_0, y_0)$ and $p_1 = (x_1, y_1)$ be the top-left and bottom-right pixels of $\mathcal{R}$ relative to the image. These are called the coordinates of $\mathcal{R}$.*

Definition 1.2 (Region Node)**.** *A Region Node R has a one-to-one and onto mapping with a Region $\mathcal{R}$. R has k internal nodes associated with it. If $\mathcal{R}$ is over an $r \times r$ set of pixels (i.e. the atomic unit), then R has k leaf nodes (e.g. k-mixture of gaussians). Else, R has k sum nodes.*

Definition 1.3 (Decomposition)**.** *Let $\mathcal{R}$ be a Region. A Decomposition $\mathcal{D}$ is an axis-aligned partitioning of $\mathcal{R}$ into two Regions $\mathcal{R}_1$ and $\mathcal{R}_2$.*

The decomposition $\mathcal{D}$ of a Region $\mathcal{R}$ involves a few steps. Let $\mathcal{R}_1$ and $\mathcal{R}_2$ be the resulting subregions product of the decomposition. The resulting subgraph of the SPN S of this decomposition is a DAG G. If $\mathcal{R}$ is the entire image, then the root of G is a single sum node and $G = S$. Otherwise, then the root of G is a region node and thus the root of G is a set of k sum nodes. Let R be the root node of G. Region nodes R_1 and R_2 will both have k sum nodes (or univariate distributions for leaves). We shall denote as R_i^j the j-th sum node of region node i. For each pairing of sum nodes (R_1^i, R_2^j), we create a product node π and add R_1^i and R_2^j as children of π. The set Π of these product nodes are the decomposition nodes of $\mathcal{R}$ into $\mathcal{R}_\infty$ and $\mathcal{R}_\in$. Once we have created all product nodes in this set, we add all of them as children of R. If R is a region node, then adding Π as children of R means, for every sum node σ in R, set product node $\pi \in \Pi$ as a child of σ.

Since a region node R is unique, we may have different decompositions in which the same region appears more than once. For this reason we should create a single Region Node for each possible region. We need a map function that takes the

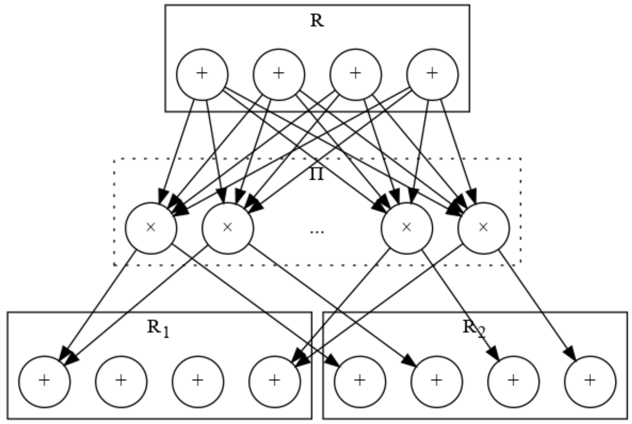

FIGURE 2. A decomposition of a Region $\mathcal{R}$ into two subregions $\mathcal{R}_1$ and $\mathcal{R}_2$. The set Π of product nodes are the decomposition nodes that connect the unsplit image to the partitions. Each element $\pi \in \Pi$ connects a pairing of a sum node of $\mathcal{R}_1$ and of $\mathcal{R}_2$.

top-left and bottom-right pixel positions of a region and maps it to a number for storage. Since every region is unique, we need a one-to-one and onto function.

Definition 1.4 (Region map function)**.** *A region map function is a function that maps a region into an integer. We define it as*

$$f : \mathbb{Z}_m \times \mathbb{Z}_n \times \mathbb{Z}_m \times \mathbb{Z}_n \to \mathbb{Z}_{m^2n^2}$$
$$f(x_1, y_1, x_2, y_2) = ((y_1 m + x_1)m + x_2)n + y_2$$

where $x_1, x_2 \in \mathbb{Z}_m$ *and* $y_1, y_2 \in \mathbb{Z}_n$.

Proposition 1.1. *The region map function is one-to-one and onto.*

Proof. We first prove f is one-to-one. If f is injective, then $f(x_1, y_1, x_2, y_2) = f(x'_1, y'_1, x'_2, y'_2) \Rightarrow (x_1, y_1, x_2, y_2) = (x'_1, y'_1, x'_2, y'_2)$. Suppose $f(x_1, y_1, x_2, y_2) = f(x'_1, y'_1, x'_2, y'_2)$ for some $x_i \in \mathbb{Z}_m$ and $y_i \in \mathbb{Z}_n$. Then we have:

$$((y_1 m + x_1)m + x_2)n + y_2 = ((y'_1 m + x'_1)m + x'_2)n + y'_2$$
$$(m^2 y_1 + m x_1 + x_2)n + y_2 = (m^2 y'_1 + m x'_1 + x'_2)n + y'_2$$
$$m^2 n y_1 + mn x_1 + n x_2 + y_2 = m^2 n y'_1 + mn x'_1 + n x'_2 + y'_2$$
$$m^2 n(y_1 - y'_1) + mn(x_1 - x'_1) + n(x_2 - x'_2) + (y_2 - y'_2) = 0$$

But $m, n > 0$. Therefore, it is easy to see that $x_i - x'_i = 0$ and $y_i - y'_i = 0$ is necessary for the equation to hold. Proof of surjection is simple. Since we know f is one-to-one and that $\mathbb{Z}_m \times \mathbb{Z}_n \times \mathbb{Z}_m \times \mathbb{Z}_n$ has the same number of elements as $\mathbb{Z}_{m^2n^2}$, than it follows that f must be onto. □

Bijection of the region map function is necessary since we need the inverse function f^{-1} to be symmetrical to f. That is, we must be able to encode a region into a number and later be able to find what region a number represents. We define the inverse function of f below.

Definition 1.5 (Inverse region map function)**.** *The inverse of the region map function is given by the decomposition of an integer* $r \in \mathbb{Z}_{m^2n^2}$ *into a tuple* $(x_1, y_1, x_2, y_2) \in \mathbb{Z}_m \times \mathbb{Z}_n \times \mathbb{Z}_m \mathbb{Z}_n$*. Let* $g = f^{-1}$*. We define* g *as an algorithm as follows*

Algorithm 1 Function $g = f^{-1}$

Input $r \in \mathbb{Z}_{m^2n^2}$
Output $(x_1, y_1, x_2, y_2) \in \mathbb{Z}_m \times \mathbb{Z}_n \times \mathbb{Z}_m \times \mathbb{Z}_n$

$y_2 \leftarrow i \mod n$
Let $c \in \mathbb{Z}_{m^2n^2}$
$c \leftarrow \frac{(r-y_2)}{n}$
$x_2 \leftarrow c \mod m$
$c \leftarrow \frac{c-x_2}{m}$
$x_1 \leftarrow c \mod m$
$y_1 \leftarrow \frac{c-x_1}{w}$
return (x_1, y_1, x_2, y_2)

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).