An Introduction to Sum-Product Networks

A collection of studies on properties, structure, inference and learning on Sum-Product Networks

Student: Renato Lui Geh

Supervisor: Denis Deratani Mauá (DCC IME-USP)

University of São Paulo / Universidade de São Paulo (USP)

Institute of Mathematics and Statistics / Instituto de Matemática e Estatística (IME)

Abstract

This work is a collection of ongoing studies I am working on for my undergraduate research project on automatic learning of Sum-Product Networks. The main objective of this work is logging my study notes on this subject in an instructive and uncomplicated way. Most scientific papers are cluttered with intricate names and require extensive background on the subject in order for the reader to understand what is going on. In this paper we seek to provide an easy reference and introductory reading material to those who intend to work with Sum-Product Networks.

This study is divided into five main sections. We start with an introductory section regarding probabilistic graphical models and why Sum-Product Networks are so interesting. Next we talk about the structure of the model. Thirdly, we analyse some properties and theorems. Fourthly, we look on how to perform exact tractable inference. And finally we take a look at how to perform learning.

Contents

1 Introduction

We assume the reader has already read the notation A and has the mathematical background required B defined in the Appendix.

In this section we show what the usual problems with probabilistic graphical models are and what led to the creation of Sum-Product Networks. Additionally, we show some results from experiments Poon and Domingos performed on the inaugural Sum-Product Network article *Sum-Product Networks: A New Deep Architecture* [PD11].

1.1 Motivation

Probabilistic Graphical Models (PGMs) perform inference through posterior probabilities on the query and evidence. Thus, inference would look roughly like this:

$$P(X|\mathbf{e} = e_1, \ldots, e_q)$$

Where X is called the variable query and $\mathbf{e}$ the evidence, that is, the observed instances of the variables.

Using the definition of conditional probability,

$$P(X|\mathbf{e}) = \frac{P(X, \mathbf{e})}{P(\mathbf{e})}$$

We get the following equation:

$$P(X|\mathbf{e}) = \frac{P(X, \mathbf{e})}{P(\mathbf{e})} = \alpha P(X, \mathbf{e}) = \alpha \sum_{\mathbf{y}} P(X, \mathbf{e}, \mathbf{y}) \tag{1}$$

Where $\mathbf{y}$ is a hidden variable. That is, let $\mathbf{X}$ be the complete set of variables. Then $\mathbf{X} = \{X\} \cup \mathbf{E} \cup \mathbf{Y}$, where X is the query, $\mathbf{E}$ is the set of evidence variables and $\mathbf{Y}$ is a set of non-query non-evidence variables. Thus $\mathbf{y}$ is an instance of $\mathbf{Y}$.

We can see that $P(X, \mathbf{e}, \mathbf{y})$ is actually a subset of the full joint distribution. Since we are summing out the hidden variables, we are actually discarding all the possible values of $\mathbf{y}$ and taking into account all the possibilities where the query given the evidence occur.

Now consider a Bayesian network as the PGM of our choice. We know that Bayesian networks have the property of representing the full joint distribution as a product of conditional probabilities:

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|Par(X_i)) \tag{2}$$

Where $Par(X_i)$ are the values of the parents of X_i. From this property we know that we can now compute inference by applying Equation (2) on Equation (1). By doing that we get inference by computing the sum of products of conditional probabilities from the network. This is fundamental to Adnan Darwiche's *network polynomial* [Dar03; Dar09], a concept that is the core of Sum-Product Networks.

We know that we can compute inference by summing out the hidden variables and then multiplying the remaining factors, but this process relies on adding and then multiplying an exponential number of probabilities. In fact, if we don't take the order of the terms in the summation into account, the complexity reaches $O(np^n)$, where p is the number of possible values a variable may take. If we move the independent terms from the summation the complexity is then $O(p^n)$. This is obviously intractable, and a reason why approximate inference is often the best solution.

Bayesian networks are not the only model that have intractable exact inference. Most PGMs suffer from intractability of inference, and hence intractability of learning. However Domingos and Poon argument that "classes of graphical models where inference is tractable exist [. . .], but are quite limited in the distributions they can represent compactly." [PD11].

Sum-Product Networks provide a graphical model where inference is both tractable and exact whilst still being more general than existing tractable models.

1.2 Background

In Adnan Darwiche's *A Differential Approach to Inference in Bayesian Networks* [Dar03] and *Modeling and Reasoning with Bayesian Networks* [Dar09], Darwiche presents a new way of representing full joint distributions through a *network polynomial.* In this subsection we will show what a network polynomial is.

Consider the following Bayesian network:

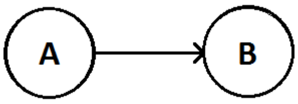

Figure 1: A Bayesian network $A \to B$, that is, the variable B depends on A.

A	Θ_A
a	$\theta_a = 0.3$
$\overline{a}$	$\theta_{\overline{a}} = 0.7$

A	B	$\Theta_{B\|A}$
a	b	$\theta_{b\|a} = 0.1$
a	$\overline{b}$	$\theta_{\overline{b}\|a} = 0.9$
$\overline{a}$	b	$\theta_{b\|\overline{a}} = 0.8$
$\overline{a}$	$\overline{b}$	$\theta_{\overline{b}\|\overline{a}} = 0.2$

Tables 1 and 2

The two tables above describe the Bayesian network in Figure 1. From these tables we can construct the full joint distribution table by applying the definition of conditional probability we saw in the previous subsection.

A	B	$P(A, B)$
a	b	$\Theta_a\Theta_{b\|a}$
a	$\overline{b}$	$\Theta_a\Theta_{\overline{b}\|a}$
$\overline{a}$	b	$\Theta_{\overline{a}}\Theta_{b\|\overline{a}}$
$\overline{a}$	$\overline{b}$	$\Theta_{\overline{a}}\Theta_{\overline{b}\|\overline{a}}$

1.3 Experiments

A Notation

In this section we show the notations we use throughout this paper.

A.1 Letters

We use an uppercase letter to denote a variable. A lowercase letter denotes an instance of a variable. A bold fonted letter is a set. A bold fonted uppercase letter is a set of variables. For instance:

$$\mathbf{X} = \{X_1 = x_1, \ldots, X_n = x_n\}$$

A.2 Events and evidence

The letter 'e', regardless of case, is reserved for events and evidence. An uppercase 'E' is an evidence variable. An uppercase bold fonted '$\mathbf{E}$' is the set of evidence variables. A lowercase 'e' is a particular observed event variable. A bold fonted lowercase '$\mathbf{e}$' is the set of variables of a particular observed event.

A.3 Probabilities

All functions of the form $P(\cdot)$ are probability functions. Joint probability distributions have the variables separated by commas $P(X, Y)$ instead of $P(X \wedge Y)$. We call prior probabilities the probability functions of the form $P(X)$. Posterior probabilities are of the form $P(\mathbf{X}|\mathbf{Y})$.

A.4 Arrows

An arrow pointing to the right may have two possible meanings:

- Dependency
  - $A \rightarrow B$ is read as B *depends of* A.
- Directed edge connectivity
  - $A \rightarrow B$ is read as *there exists an edge from node* A *to node* B.

Meanings will be clear from context.

B Mathematical background

References

[Dar03] Adnan Darwiche. “A Differential Approach to Inference in Bayesian Networks”. In: (2003).

[Dar09] Adnan Darwiche. *Modeling and Reasoning with Bayesian Networks*. 1st Edition. Cambridge University Press, 2009.

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