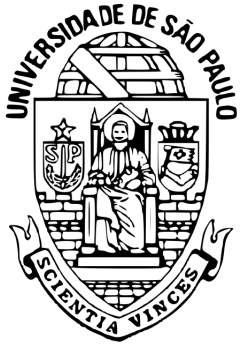

Mobile Robot Self-Driving Through Image Classification Using Discriminative Learning of Sum-Product Networks

Undergraduate Thesis

Student: Renato Lui Geh
Advisor: Prof. Denis Deratani Mauá

INSTITUTE OF MATHEMATICS AND STATISTICS
UNIVERSITY OF SÃO PAULO

São Paulo, Brazil

2018

Acknowledgements

I would like to greatly thank my advisor, Prof. Denis Deratani Mauá, for the support and attention, but most of all for the patience of having to read countless reports for both my undergraduate research and undergraduate thesis, many of which were not short.

To my parents, Chen and Luiz, for making sure I wanted for nothing, for giving me the best education possible, and for always assuring me of my abilities.

A special thank you to Maria Clara Cardoso, my best friend and confidant, whose support and company helped me immensely through the last years. Our conversations are always filled with laughter and I will always hold them dear.

A warm thanks to my friends and colleagues Ricardo and Yan, whose camaraderie and friendship I cherish deeply.

I'd also like to express great gratitude to all professors I had during my undergraduate, whose trade is often overlooked and underappreciated, yet manage to teach us so much and inspire us to always do our best.

This work was partially supported by CNPq grant PIBIC 800585/2016.

Abstract

GEH, L. R. **Mobile robot self-driving through image classification using discriminative learning of sum-product networks**. Institute of Mathematics and Statistics, University of São Paulo, São Paulo, Brazil, 2018.

Driving has proven to be a very difficult task for machines to emulate, not only due to the inherent complexity of the problem but also because of the need for accurate real-time predictions. Nonetheless, recent advances in computer vision and machine learning have shown promising results in the real-world. Mobile robots are low-cost miniature computers with limited processing power and memory. The problem of self-driving can be similarly applied to the mobile robot domain as a down-scaled version of the same task, with an additional hardware constraint. Sum-product networks are probabilistic graphical models capable of representing tractable probability distributions containing a great number of variables. Exact inference is asymptotically linear to the number of edges in the network's graph, and its deep architecture is capable of representing a wide range of distributions. In this work, we attempt to model autonomous driving by using sum-product networks on a small mobile robot. We model this task as an imitation learning problem through image classification. We present accuracy results on an artificial self-driving dataset for different sum-product network learning algorithms, providing a comparative study not only for different network architectures, but also discriminative and generative models. Finally, we provide a real-world mobile robot implementation on a miniature computer.

Keywords: sum-product networks, probabilistic graphical models, machine learning, robotics

Abbreviations

DAG	Directed acyclic graph
EM	Expectation-maximization
GD	Gradient descent
IV	Indicator variable
MAP	Maximum a posteriori probability
MPE	Most probable explanation
MPN	Max-product network
PGM	Probabilistic graphical model
RV	Random variable
SGD	Stochastic gradient descent
SPN	Sum-product network
SPT	Sum-product tree

Symbols and notations

μ	Gaussian distribution mean
σ	Gaussian distribution standard deviation
X	Random variable
x	Instance of random variable X
$\mathbf{X}$	Set of random variables
$\mathcal{X}$	Sample space of X
Z	Partition function
ϕ	Factor (potential)
$[X = x]$	Indicator function for RV valuation $X = x$

List of Figures

List of Tables

Contents

Chapter 1

Introduction

In this chapter we first describe the motivations and objectives of this thesis. Next, we describe the structure of this document.

1.1 Motivation and objectives

Self-driving is a challenging computer vision task, mainly due to its inherent complexity and the necessity for real-time decision making. Although there have been many promising results the past few years on autonomous driving, the task still relies on the underlying problem of following a pathway through visual cues (usually road markings). A possible approach to this task is through imitation learning by means of image classification. That is, the agent tasked with driving should be able to reliably mimic human behavior by correctly classifying whether to turn, stop or go straight given an image captured in front of the car.

Mobile robots are low cost machines capable of movement. These robots are usually small, and because of their size and cost, often don't have the same performance capabilities as a desktop computer. However, these domain traits make mobile robot self-driving a very similar analogue to real-world autonomous cars. Processing power and memory constraints play a big role in this case, and translate well to embedded systems present in a self-driving car.

Sum-product networks (SPNs) are probabilistic graphical models that are able to represent a wide range of tractable probability distributions of many variables. SPNs have shown impressive results in several domains, and particularly that of image classification. Their deep architecture seems to capture features and contexts well, and since inference is computed in time linear to the network's edges, SPNs are promising models for fast inference in self-driving.

In this work, we attempt to model self-driving of mobile robots through image classification. For the task of classification our objective is to use sum-product networks learned discriminatively, though we also give results for generative SPNs, comparing not only generative and discriminative learning, but also different SPN architectures.

1.2 Thesis structure

This thesis is structured as follows. In chapter 2, we first provide background on sum-product networks, where we formally define an SPN, present key properties on their structure, explain how to compute exact inference and find an approximation of the maximum a posteriori probability (MAP).

In **??**, we show how to compute the partial derivatives with respect to a sub-SPN and to its weights, leading on how to perform gradient descent and then on learning the weights of the network through gradient descent both generatively and discriminatively.

?? is dedicated to algorithms for learning the structure of an SPN. We explain the two structural learning algorithms that were used in the experiments.

For **??**, we first show how we model self-driving as an image classification problem. We then specify the architecture of the robot used in the experiments, giving specifications on the hardware and software used. Furthermore, we describe some concepts of control we use for navigation.

In **??**, we provide classification results on many image classification datasets from various domains with different learning algorithms. We then describe the self-driving dataset used for training, and give in-dataset accuracies as well as real-world empirical results on the mobile robot itself.

Finally, in **??** we give our conclusions and provide some discussion of the results.

There is an additional section of this thesis in which we give a brief subjective insight on the work done for this thesis. We also list subjects we deemed important for the work done in this thesis.

Furthermore, **??** contains all proofs done in this thesis.

Chapter 2

Sum-product networks

In this chapter we provide some background concepts needed for defining a sum-product network. Once this is covered, we formally define an SPN, list some interesting properties on their structure, and describe how to perform exact inference (i.e. extract the probability of evidence of some valuation) and how to find an approximation of the maximum a posteriori probability.

We leave all proofs in **??**.

2.1 Background

The objective of probabilistic modelling is to compactly represent a probability distribution, be able to find a good approximation to the real function and be able to efficiently compute both the marginals and modes. Probabilistic graphical models (PGMs) attempt to solve this through the use of graphs, representing distributions as a normalized product of factors (Pearl, 1988).

$$P(X = x) = \frac{1}{Z} \prod_k \phi_k(x_{\{k\}})$$

Where $x \in \mathcal{X}$ is a d-dimensional vector valuation of RVs $\mathbf{X}$ on sample space $\mathcal{X}$, and factor (also called a potential) ϕ_k is a function mapping instantiations of X to a non-negative number. Z is the partition function $Z = \sum_{x \in \mathcal{X}} \prod_k \phi_k(x_{\{k\}})$ that sums out all variables and normalizes the term above it to the $[0, 1]$ range.

A downside of this representation is that inference is exponential on the worst case, which makes learning also exponential, as it uses inference as a subroutine. To get around this problem, Darwiche proposed in Darwiche, 2003 the notion of *network polynomial*.

A network polynomial is a function over the probabilities of each instantiation. Let $\Phi(x)$ be a probability distribution. The network polynomial of $\Phi(x)$ is the function $f = \sum_{x \in \mathcal{X}} \Phi(x)\Pi(x)$, where $\Pi(x)$ is the product of the IVs of each variable on instantiation x,

where each indicator variable $[Y = y]$ has a value of zero if $Y \neq y$ in x and a value of one otherwise (i.e. if $Y = y$ in x or $Y \notin x$).

As an example, take the bayesian network $\mathcal{N} = A \rightarrow B$ with binary variables. Let λ_a, $\lambda_{\overline{a}}$, λ_b and $\lambda_{\overline{b}}$ be the indicator variables for when $A = 1$, $A = 0$, $B = 1$ and $B = 0$ respectively. The network polynomial of $\mathcal{N}$ is the expression

$$f_{\mathcal{N}} = P(a)P(b|a)\lambda_a\lambda_b + P(a)P(\overline{b}|a)\lambda_a\lambda_{\overline{b}} + P(\overline{a})P(b|\overline{a})\lambda_a\lambda_{\overline{b}} + P(\overline{a})P(\overline{b}|\overline{a})\lambda_{\overline{a}}\lambda_{\overline{b}}.$$

The main advantage of this representation is to avoid recomputing terms. For instance, take an instantiation of $x = \{A = 0\}$. Then, the network polynomial will be as follows.

$$\begin{aligned} f_{\mathcal{N}}(x) &= P(a)P(b|a) \cdot 0 \cdot 1 + P(a)P(\overline{b}|a) \cdot 0 \cdot 1 + P(\overline{a})P(b|\overline{a}) \cdot 1 \cdot 1 + P(\overline{a})P(\overline{b}|\overline{a}) \cdot 1 \cdot 1 = \\ &= P(\overline{a})P(b|\overline{a}) + P(\overline{a})P(\overline{b}|\overline{a}) \end{aligned}$$

Which means we can avoid computing values from the two first terms. We can also compute the network polynomial of some unnormalized probability distribution as long as we divide by the partition function, defined as the network polynomial with all indicators set to one. Although the network polynomial has exponential size in terms of variables, computing the probability of evidence is linear in its size. By representing the network polynomial as an arithmetic circuit of sums and products, one can prove that the cost of inference is indeed polynomial.

2.2 Definition

Sum-product networks borrow many concepts from network polynomials and arithmetic circuits. There are many definitions of SPNs and in this thesis we present two. The first definition is given by the seminal article Poon and Domingos, 2011, and can be seen as a more low-level approach to defining the network. The second, based on Gens and Domingos, 2013, is a stronger definition, but one which we will use more throughout this thesis, as it lends itself better to continous data.

Let $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$ be the set of all variables. We shall call this set the root scope. Let G be a graph G. The sets of vertices and edges of G will be denoted by $V(G)$ and $E(G)$. We will call $\mathrm{Ch}(n)$ and $\mathrm{Pa}(n)$ the sets of children and parents of node $n \in V(G)$.

Definition 2.1 (Sum-product network; Poon and Domingos, 2011)**.** *A sum-product network (SPN) over variables $X_1, X_2, \ldots, X_n$ is a DAG whose leaves are indicator variables $[X_1 = x_1^1], [X_2 = x_2^1], \ldots, [X_n = x_n^1], \ldots, [X_1 = x_1^d], [X_2 = x_2^d], \ldots, [X_n = x_n^d]$. Its internal nodes are weighted sums or products. Each edge coming out from a sum node n to another node j has a non-negative weight associated with it. We denote such weight by $w_{n,j}$. The value of a sum node n is $v_n = \sum_{j \in Ch(n)} w_{n,j} v_j$, where v_j is the value of node j. The value of a product node n is $v_n = \prod_{j \in Ch(n)} v_j$. The value of a leaf node is the value of the indicator variable. The value of the SPN is the value of its root node.*

Throughout this thesis, we denote by $S(X = x)$ the value of an SPN S given evidence x. A sub-SPN S_n of S is the subgraph of S rooted at n. The partition function of S is when all indicator variables are set to one, and is denoted by $S(*)$. The scope of an SPN S is the union set of all scopes of its children. A leaf's scope is the scope of its IV.

Bibliography

[Darwiche 2003] Adnan Darwiche. "A Differential Approach to Inference in Bayesian Networks". In: (2003) (cit. on p. 3).

[Gens and Domingos 2013] R. Gens and P. Domingos. "Learning the Structure of Sum-Product Networks". In: *ICML* 30 (2013) (cit. on p. 4).

[Poon and Domingos 2011] H. Poon and P. Domingos. "Sum-Product Networks: A New Deep Architecture". In: *UAI* 27 (2011) (cit. on p. 4).

[Pearl 1988] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988 (cit. on p. 3).