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

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

Institute of Mathematics and Statistics — University of São Paulo

In an Ideal world...

$$\overbrace{\overbrace{\overbrace{\textbf{Mobile Robot}}^{1}\ \overbrace{\textbf{Self-Driving}}^{2}}^{3}\ \textbf{Through}\ \overbrace{\textbf{Image Classification}}^{4}}^{5}$$

$$\textbf{Using}\ \underbrace{\underbrace{\textbf{Discriminative Learning}}_{6}\ \textbf{of}\ \underbrace{\textbf{Sum-Product Networks}}_{7}}_{8}$$

...but life is short

Section 2

Mobile Robot Self-Driving Through Image Classification

Using Discriminative Learning of Sum-Product Networks

Section 1

Section 3: (1+2)

Sum-Product Networks

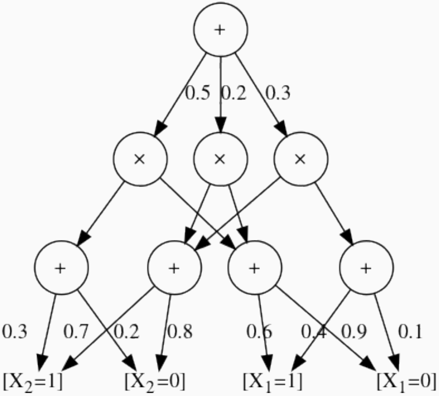
Definition 1 (Gens and Domingos 2013).

A sum-product network (SPN) is a DAG where each node can be defined recursively as follows.

1. A tractable univariate probability distribution is an SPN.
2. A product of SPNs with disjoint scopes is an SPN.
3. A weighted sum of SPNs with the same scope is an SPN, provided all weights are positive.
4. Nothing else is an SPN.

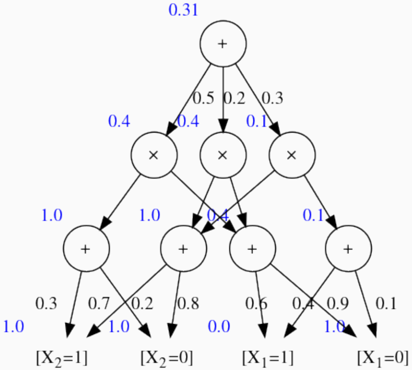
Sum-product network

The value $S(X)$ of an SPN is equal to $\phi(X)$, an unnormalized probability function, if it obeys certain properties. If all weights sum to one, $S(X) = P_\phi(X)$ (Poon and Domingos 2011).



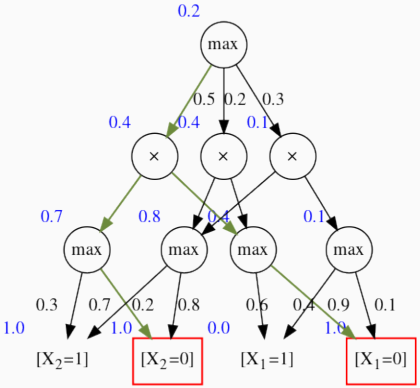
Probability of evidence

Single backward pass computes $S(X = \{X_1 = 0\}) = 0.31$. Linear on the number of edges



Maximum a posteriori probability

Replace sums with max nodes. Backward pass followed by forward pass computes most probable explanation, i.e. find $\arg\max_{x \in X} P(X, E)$.



Learning

Structure

- PD-Dense architecture (Poon and Domingos 2011)
- **Clustering on Variables** (Dennis and Ventura 2012)
- **Gens-Domingos LearnSPN** (Gens and Domingos 2013)
- Using deep learning techniques (Peharz et al. 2018)
- many others...

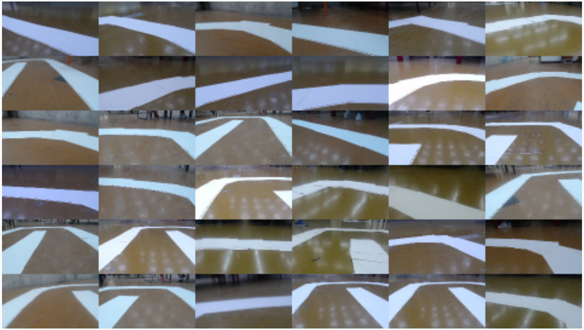
Weights

- **Generative and discriminative gradient descent**
- Generative Expectation-Maximization
- Extended Baum-Welch (Rashwan, Poupart, and Zhitang 2018)
- many others...

Self-Driving

Dataset

Dataset used: Moraes and Salvatore 2018



Lane tracking dataset with 80×45 RGB images. Each labeled with either UP, LEFT or RIGHT.

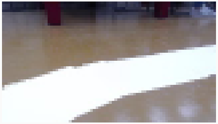
Self-driving as image classification

Let $X = \{X_0, X_1, \ldots, X_{n-1}\}$ be an **image**. Every $X_i = x_i$ refers to the i-th pixel with a grayscale intensity of x_i.

Let $Y = \{\text{UP}, \text{LEFT}, \text{RIGHT}\}$ be the **classification variable**.

LEFT

UP



RIGHT

The entire scope of variables is $X \cup \{Y\}$.

Objective: $\arg\max_{y \in Y} P(Y = y|X)$

Pre-processing

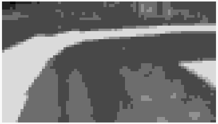
Pipeline:

original RGB image $\rightarrow$ grayscale $\rightarrow$ some T transformation.

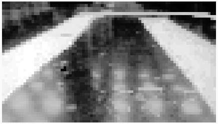
Three transformations tested:

1. Otsu binarization (Otsu 1979)
2. Quantization (resolution downscaling)
3. Histogram equalization

1



2



3

Raspberry Pi 3 Model B — Berry

CPU: Quad Core 1.2GHz Broadcom BCM2837 64bit ARMv7

Memory: 1GB RAM

Storage: 16GB SSD

Brick

Lego Mindstorms NXT v2 — Brick

CPU: Atmel AT91SAM7S256 48MHz 32bit ARMv4

Memory: 64KB RAM

Storage: 256KB Flash

Robot

Berry handles inference, passing predicted label to **Brick**.

Brick handles motors according to label received from **Berry**.

Message passing through USB cable.

Driving with SPNs

Modelling

Every pixel X_i is a variable in the distribution represented by the SPN, i.e. no additional feature extraction, end-to-end.

Two architectures:

GD: LearnSPN (Gens and Domingos 2013)

DV: Clustering on Variables (Dennis and Ventura 2012)

Three weight setups:

g: Generative gradient descent (Poon and Domingos 2011)

d: Discriminative gradient descent (Gens and Domingos 2012)

s: Proportional weights for GD, random weights for DV

Accuracy

Accuracy (%)	DV+g	DV+d	DV+s	GD+g	GD+d	GD+s
B	78.8	78.8	78.8	82.8	83.8	85.0
Q_2	78.6	78.0	78.0	78.6	80.4	79.4
$Q_2 + E$	76.6	76.6	76.8	79.6	82.8	81.8
Q_3	77.4	77.4	77.4	77.6	80.2	79.8
$Q_3 + E$	70.4	76.6	76.6	79.2	81.2	77.4
Q_4	78.2	78.4	78.2	76.0	**78.2**	76.4
$Q_4 + E$	76.6	76.6	76.8	76.0	74.6	80.6
Q_5	77.8	78.4	78.4	77.6	74.0	73.8
$Q_5 + E$	76.6	76.6	76.6	72.0	72.8	72.0
Q_6	77.4	78.4	78.4	75.2	**74.4**	72.0
$Q_6 + E$	76.0	76.4	76.4	73.0	75.0	73.6
Q_7	78.2	78.4	78.4	62.8	72.2	71.4
$Q_7 + E$	76.2	76.4	76.4	70.6	71.4	71.6
$\emptyset$	78.0	78.4	78.4	62.4	**62.4**	62.4
E	76.4	76.4	76.4	60.4	60.0	61.2

Inference time

Inference (secs)	DV+g	DV+d	DV+s	GD+g	GD+d	GD+s
B	0.23	0.25	0.25	0.38	0.37	0.31
Q_2	0.22	0.24	0.23	0.28	0.34	0.16
$Q_2 + E$	0.22	0.23	0.23	0.38	0.30	0.27
Q_3	0.22	0.23	0.22	0.22	0.32	0.17
$Q_3 + E$	0.22	0.23	0.22	0.34	0.32	0.31
Q_4	0.22	0.22	0.23	0.16	**0.17**	0.13
$Q_4 + E$	0.23	0.27	0.29	0.13	0.14	0.13
Q_5	0.22	0.26	0.28	0.07	0.05	0.02
$Q_5 + E$	0.22	0.29	0.25	0.05	0.05	0.02
Q_6	0.23	0.24	0.23	0.04	**0.05**	0.01
$Q_6 + E$	0.22	0.24	0.28	0.03	0.04	0.02
Q_7	0.23	0.23	0.26	0.03	0.01	0.01
$Q_7 + E$	0.22	0.26	0.24	0.01	0.01	0.01
$\emptyset$	0.22	0.26	0.23	0.02	**0.01**	0.01
E	0.23	0.23	0.22	0.01	0.01	0.02

Chosen models

Model 1: Q_4, **GD+d**

Accuracy: 78.2%

Desktop time: 170ms

Berry time: 700ms

Model 2: Q_6, **GD+d**

Accuracy: 74.4%

Desktop time: 50ms

Berry time: 150ms

Model 3: $\emptyset$, **GD+d**

Accuracy: 62.4%

Desktop time: $<$ 10ms

Berry time: 75ms

“Real world” scenario

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

(`https://youtu.be/vhpWQDX2cQU`)

Implementation

Inference and learning: GoSPN
(`https://github.com/RenatoGeh/gospn`)

Mobile robot implementation: GoDrive
(`https://github.com/RenatoGeh/godrive`)

Thank you.

Questions?

References i

- Dennis, Aaron and Dan Ventura (2012). "Learning the Architecture of Sum-Product Networks Using Clustering on Variables". In: *Advances in Neural Information Processing Systems* 25.
- Gens, Robert and Pedro Domingos (2012). "Discriminative Learning of Sum-Product Networks". In: *Advances in Neural Information Processing Systems 25 (NIPS 2012)*.
- – (2013). "Learning the Structure of Sum-Product Networks". In: *International Conference on Machine Learning* 30.
- Moraes, Paula and Felipe Salvatore (2018). *Self Driving Data*. URL: `https://github.com/felipessalvatore/self_driving_data`.

Otsu, Nobuyuki (1979). “A Threshold Selection Method from Gray-Level Histograms”. In: *IEEE Transactions on Systems, Man, and Cybernetics* 9.1, pp. 62–66.

Peharz, R. et al. (2018). “Probabilistic Deep Learning using Random Sum-Product Networks”. In: *ArXiv e-prints*.

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

Rashwan, Abdullah, Pascal Poupart, and Chen Zhitang (2018). “Discriminative Training of Sum-Product Networks by Extended Baum-Welch”. In: *Proceedings of the Ninth International Conference on Probabilistic Graphical Models*. Vol. 72. Proceedings of Machine Learning Research, pp. 356–367.