

Sum-Product Network and Its Application



Yilin ZHENG

11510506@mail.sustech.edu.cn

Southern University of Science and Technology

April 4, 2019

Introduction

Sum-Product Network

SPN on Image Completion

Progress

Introduction

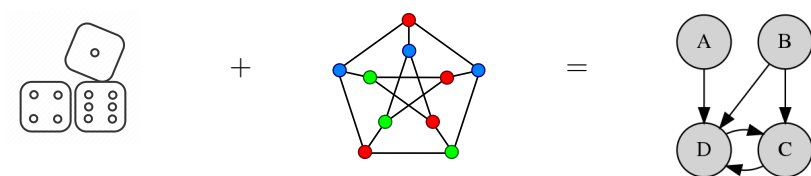
Sum-Product Network

SPN on Image Completion

Progress

Probabilistic graphical models(PGMs):

- Probability Theory
- Graph Theory



- Bayesian Network(BN): $p_{\mathcal{B}}(\mathbf{X}) = \prod_{X \in \mathbf{X}} p(X|\mathbf{par}(X))$
- Markov Random Field(MRF):

$$p_{\mathcal{M}}(\mathbf{X}) = \frac{1}{Z_{\mathcal{M}}} \prod_{l=1}^L \Psi_{\mathbf{C}_l}(\mathbf{X}_{\mathbf{C}_l})$$

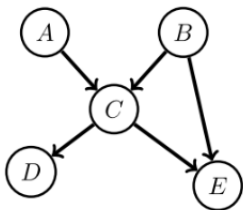


Figure: Example of a BN

$$p_{\mathcal{B}}(A, B, C, D, E) = p(A)p(B)p(C|A, B)p(D|C)p(E|B, C)$$

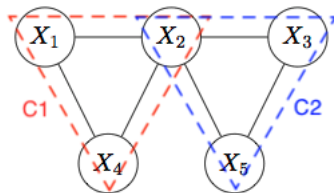


Figure: Example of a MRF

$$p_{\mathcal{M}}(X_1, X_2, X_3, X_4, X_5) = \frac{1}{Z} \Psi_{\mathbf{C}_1}(X_1, X_2, X_4) \Psi_{\mathbf{C}_2}(X_2, X_3, X_5)$$

- ▶ Representation: joint distribution probability
- ▶ Learning: structure and parameters
- ▶ Inference: computing posterior marginal distributions

Classical PGMs suffer from some problems:

1. Scales unproportionally to the complexity of the model
2. Approximate learning yields unpredictable results
3. Intractable after slight modification
4. Separation in learning and inference

Sum-Product Network: Poon and Domingos(2011)

1. Scales up linearly
2. Exact learning
3. Tractability
4. Combination of learning and inference

- ▶ Reproduce the SPN application on image completion
- ▶ Optimization of program
- ▶ Analysis the results

1. Literature review
2. Implementation of an SPN
3. Reproducing the experiment of image completion
4. Optimization and Analysis
5. Thesis draft and Defence preparation

Introduction

Sum-Product Network

SPN on Image Completion

Progress

Definition (Sum-Product Network)

- ▶ $\mathcal{S} = (\mathcal{G}, \Phi(\mathbf{X}))$, \mathcal{G} is a rooted DAG and $\Phi(\mathbf{X})$ is a set of nonnegative parameters over random variables \mathbf{X}
- ▶ Leaves are numeric input, called *indicator variables*(IVs)
- ▶ Types of internal nodes: **sum** nodes or **product** nodes
- ▶ **Root** is a sum node
- ▶ Probability is computed at root
- ▶ Network polynomials of SPN:

$$f_{\mathcal{S}}(\mathbf{X}) = \sum_{X \in \mathbf{X}} \Phi(X) \prod_{X \in \mathbf{X}} (X)$$

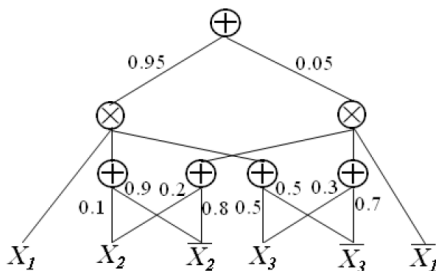


Figure: Example of an SPN with 6 indicator variables (\bar{X} represents negation of X)

$$f_S(X_1, X_2, X_3, \bar{X}_1, \bar{X}_2, \bar{X}_3) = 0.95X_1(0.1X_2 + 0.9\bar{X}_2)(0.5X_3 + 0.5\bar{X}_3) + 0.05(0.2X_2 + 0.8\bar{X}_2)(0.3X_3 + 0.7\bar{X}_3)\bar{X}_1$$

Definition (Scope of a Node(**sc**))

$$\mathbf{sc}(N) \begin{cases} \{X\} & \text{if } N \text{ is a leaf} \\ \cup_{C \in \mathbf{chi}(N)} \mathbf{sc}(C) & \text{if } N \text{ is an internal node} \end{cases}$$

Properties:

- ▶ Completeness: a sum node's children have the same scope
- ▶ Consistency: a variable and its negation are in the same scope of a product node
- ▶ Validity: completeness + consistency
- ▶ Decomposability: No overlap of scopes between different nodes

Types of learning:

- ▶ Structure learning:
 - ▶ Poon-Domingos Architecture: Rectangle regions
 - ▶ Dennis-Venture Architecture: Any shape regions
- ▶ Parameter learning:
 - ▶ Gradient method: maximize log-likelihood
 - ▶ EM algorithm: introduce latent variables, inference in E-step, update weights in M-step

Differential approach:

1. Compute the marginal distribution via network polynomials
2. Interpret the probability distribution through derivatives

Introduction

Sum-Product Network

SPN on Image Completion

Progress

Learn the half image and recover the left half image by learning and inference over the SPN constructed on given dataset.

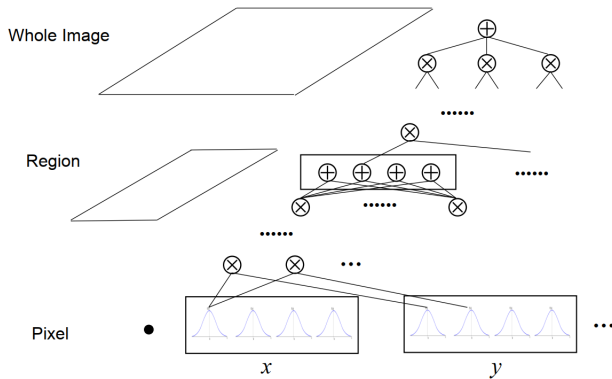


Figure: Poon-Domingos Architecture

- ▶ **common:** Contain some helper functions to provide time management, messaging between progress(MPI), parameter settings for training on clusters, and some utilities.
- ▶ **evaluation:** Process the dataset, apply network to dataset to output models, and evaluate the results generated from the models.
- ▶ **spn:** Contain the SPN architecture, including node definition, computation functions, the learning, and inference.

Files: 16 .cpp, total code: 3605 lines

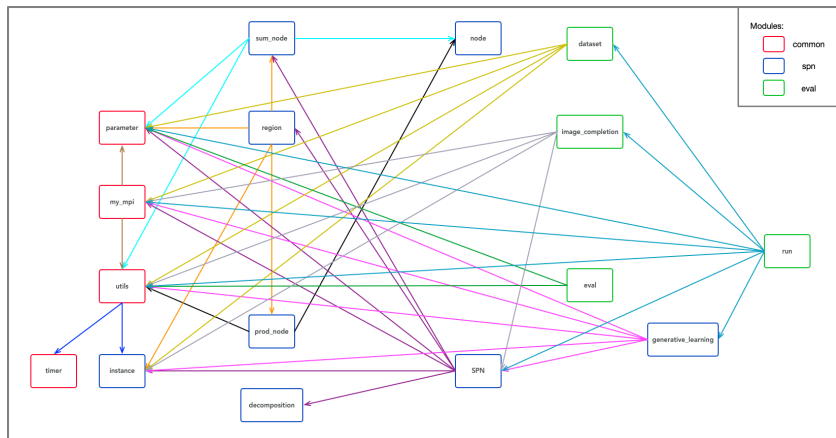


Figure: Callgraph of Experiment Program

Platform: TaiYi cluster

Cores:

- ▶ Caltech: 120
- ▶ Olivetti: 80

Caltech:

- ▶ 101 categories, from 40 to 800 images per category
- ▶ about 300×200 pixels, rescaled to 100×64 pixels in the experiment.

Olivetti:

- ▶ face images taken between April 1992 and April 1994 at AT&T Laboratories Cambridge
- ▶ image size 64×64 pixels

Introduction

Sum-Product Network

SPN on Image Completion

Progress

- ▶ A brief review of the SPN
- ▶ Monthly report till Mar. 2019
- ▶ Implementation of SPN architecture
- ▶ Unit Tests of the Program
- ▶ Interim Report

- ▶ Debug of learning part of the program
- ▶ Reproduce the results of image completions
- ▶ Draft the thesis

- ▶ Optimization of codes
- ▶ Compare and analysis the results

Thank you for your listening! 😊