A Unified Approach for Learning the Parameters of Sum-Product Networks

Han Zhao[†], Pascal Poupart[⋆] and Geoff Gordon[†]
[†]{han.zhao, ggordon}@cs.cmu.edu, [⋆]ppoupart@uwaterloo.ca
[†]Machine Learning Department, Carnegie Mellon University
[⋆]David R. Cheriton School of Computer Science, University of Waterloo

Introduction

- We present a unified approach for learning the parameters of Sum-Product networks (SPNs).
- We construct a more efficient factorization of complete and decomposable SPN into a mixture of trees, with each tree being a product of univariate distributions.
- We show that the MLE problem for SPNs can be formulated as a signomial program.
- We construct two parameter learning algorithms for SPNs by using sequential monomial approximations (SMA) and the concave-convex procedure (CCCP). Both SMA and CCCP admit multiplicative weight updates.
- We prove the convergence of CCCP on SPNs.

Background

Sum-Product Networks (SPNs):

- Rooted directed acyclic graph of univariate distributions, sum nodes and product nodes.
- We focus on discrete SPNs, but the proposed algorithms work for continuous ones as well.

Recursive computation of the network:

$$V_k(\mathbf{x} \mid \mathbf{w}) = \begin{cases} p(X_i = \mathbf{x}_i) & k \text{ is a leaf node over } X_i \\ \prod_{j \in \mathrm{Ch}(k)} V_j(\mathbf{x} \mid \mathbf{w}) & k \text{ is a product node} \\ \sum_{j \in \mathrm{Ch}(k)} w_{kj} V_j(\mathbf{x} \mid \mathbf{w}) & k \text{ is a sum node} \end{cases}$$

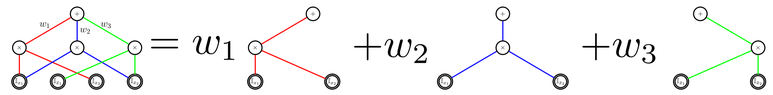
Scope: The set of variables that have univariate distributions among the node's descendants.

Complete: An SPN is *complete* iff each sum node has children with the same scope.

Decomposable: An SPN is decomposable iff for every product node v, $\mathrm{scope}(v_i) \cap \mathrm{scope}(v_j) = \varnothing$ where $v_i, v_j \in Ch(v), i \neq j$.

A Unified Framework for Learning

(SPNs as a Mixture of Trees) Theorem 1:
Every complete and decomposable SPN $\mathcal{S}$ can be factorized into a sum of $\Omega(2^h)$ induced trees (sub-graphs), where each tree corresponds to a product of univariate distributions. h is the height of $\mathcal{S}$.



Maximum Likelihood Estimation as Signomial Program:
The MLE optimization is:

$$\begin{aligned} \text{maximize}_{\mathbf{w}} \quad & \frac{f_{\mathcal{S}}(\mathbf{x}|\mathbf{w})}{f_{\mathcal{S}}(\mathbf{1}|\mathbf{w})} = \frac{\sum_{t=1}^{\tau} \prod_{n=1}^{N} \mathbb{I}_{x_n}^{(t)} \prod_{d=1}^{D} w_d^{\mathbb{I}_{w_d \in \mathcal{T}_t}}}{\sum_{t=1}^{\tau} \prod_{d=1}^{D} w_d^{\mathbb{I}_{w_d \in \mathcal{T}_t}}} \\ \text{subject to} \quad & \mathbf{w} \in \mathbb{R}_{++}^{D} \end{aligned}$$

$\tau = \Omega(2^h)$. D is the number of parameters in $\mathcal{S}$. N is the number of random variables modeled by $\mathcal{S}$. $\mathcal{T}_t$ is the t-th induced tree.

Proposition 2:
The MLE problem for SPNs is a signomial program.
Logarithmic transformation leads to a **difference of convex functions**:

$$\text{maximize } \log\left(\sum_{t=1}^{\tau(\mathbf{x})} \exp\left(\sum_{d=1}^{D} y_d \mathbb{I}_{y_d \in \mathcal{T}_t}\right)\right) - \log\left(\sum_{t=1}^{\tau} \exp\left(\sum_{d=1}^{D} y_d \mathbb{I}_{y_d \in \mathcal{T}_t}\right)\right)$$

Sequential Monomial Approximation (SMA): Optimal linear approximation in log-space, corresponds to the optimal monomial function approximation to the original signomial.

Concave-Convex Procedure (CCCP): Sequential convex relaxation by linearizing the first term, with efficient $O(|\mathcal{S}|)$ closed form solver for each convex sub-problem.

(Convergence of CCCP) Theorem 2:
Let $\{\mathbf{w}^{(k)}\}_{k=1}^{\infty}$ be any sequence generated by CCCP from any feasible initial point. Then all the limiting points of $\{\mathbf{w}^{(k)}\}_{k=1}^{\infty}$ are stationary points of the difference of convex functions program (DCP). In addition, $\lim_{k\to\infty} f(\mathbf{y}^{(k)}) = f(\mathbf{y}^*)$, where $\mathbf{y}^*$ is some stationary point of the DCP, i.e., the sequence of objective function values converges.

Algo	Update Type	Update Formula
PGD	Additive	$w_d^{(k+1)} \leftarrow P_{\mathbb{R}^{\epsilon}_{++}}\{w_d^{(k)} + \gamma \nabla_{w_d} f(\mathbf{w}^{(k)})\}$
EG	Multiplicative	$w_d^{(k+1)} \leftarrow w_d^{(k)} \exp\{\gamma \nabla_{w_d} f(\mathbf{w}^{(k)})\}$
SMA	Multiplicative	$w_d^{(k+1)} \leftarrow w_d^{(k)} \exp\{\gamma w_d^{(k)} \nabla_{w_d} f(\mathbf{w}^{(k)})\}$
CCCP	Multiplicative	$w_{ij}^{(k+1)} \propto w_{ij}^{(k)} \cdot \nabla_{v_i} f_{\mathcal{S}}(\mathbf{w}^{(k)}) \cdot f_{v_j}(\mathbf{w}^{(k)})$

Experiments

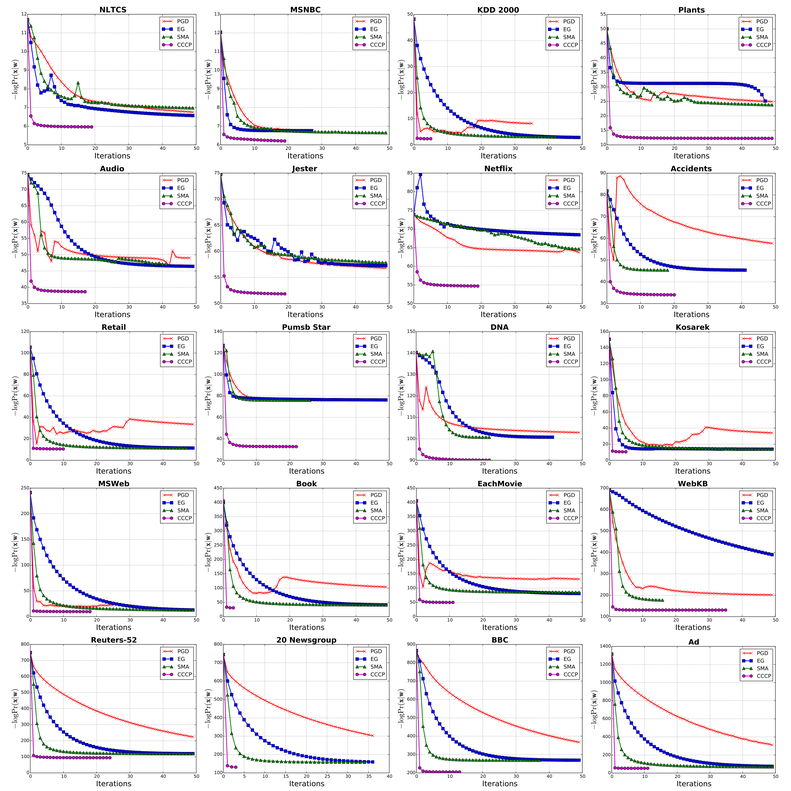


Figure 1: Negative log-likelihood values versus number of iterations for PGD, EG, SMA and CCCP on 20 benchmark datasets.

- CCCP consistently outperforms all the other three algorithms.
- We suggest CCCP for maximum likelihood estimation, and CVB for Bayesian learning of SPNs (See our ICML 2016 paper).