The Sum-Product Theorem: A Foundation for Learning Tractable Models

Abram Friesen and Pedro Domingos {afriesen, pedrod}@cs.uw.edu

Key inference problems in AI

Marginal probability	$= \sum_{x \in X} \prod_i P_i(x)$
MPE	$= \max_{x \in X} \prod_i P_i(x)$
Satisfiability	$= \bigvee_{x \in X} \bigwedge_i N_i(x)$
Constraint satisfaction	$= \bigvee_{x \in X} \bigwedge_i C_i(x)$
Integration	$= \int_{x \in X} \prod_i F_i(x)$
Nonconvex optimization	$= \min_{x \in X} \sum_i F_i(x)$
Database querying	$= \bigcup_{x \in X} \bowtie_i Q_i(x)$

These problems all have the same general form: summing a function over a semiring. [Bistarelli et al. (1997); Aji & McEliece (2000); Wilson (2005); Green et al. (2007); Dechter & Mateescu (2007)]

We identify and prove the **sum-product theorem**, which states a simple sufficient condition (decomposability) for tractable, high-treewidth, exact inference in any problem with this form.

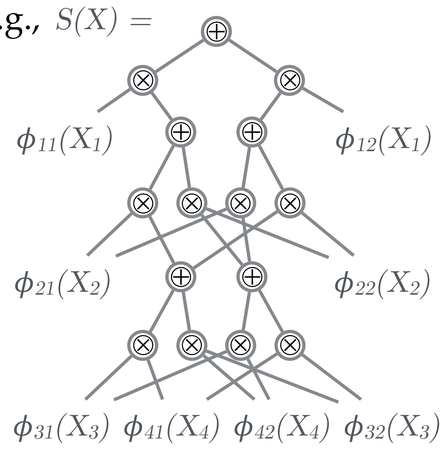
Based on it, we show how to define and learn tractable, high-treewidth representations for any such problem.

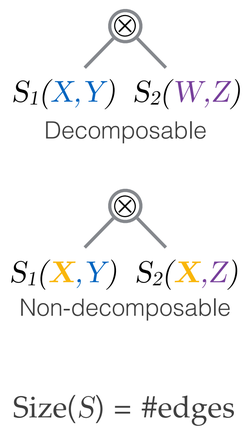
Inference, in general

We are interested in computing summations $\oplus_{x \in X} S(x)$ where $(R,\oplus,\otimes,0,1)$ is a commutative semiring, $S : X \to R$ is a function on that semiring, and $X = \{X_1, \ldots, X_n\}$ is a set of variables.

We refer to S as a sum-product function (SPF).

- e.g., $S(X) =$





In general, the cost of computing $\oplus_X S(X)$ is $O(exp(n))$.

Definition. A *(commutative) semiring* $(R,\oplus,\otimes,0,1)$ is a nonempty set R on which operations sum ($\oplus$) and product ($\otimes$) are associative and commutative and have identity elements $0,1 \in R$, such that

1) product ($\otimes$) distributes over sum ($\oplus$), and
2) $0 \neq 1$, $a \oplus 0 = a$, $a \otimes 1 = a$, and $a \otimes 0 = 0$ for all $a \in R$.

Definition. A *sum-product function (SPF)* $S : X \to R$ over (R, X, Φ) is any of

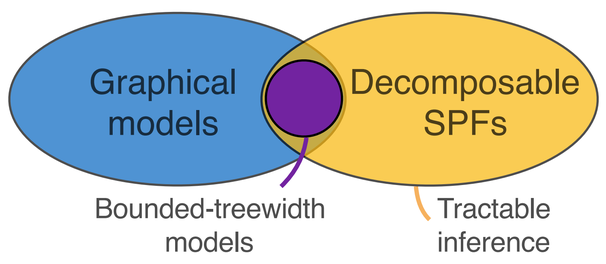
1) a function: $\phi_j(X_i)$, $\phi_j \in \Phi$
2) a product of SPFs: $S_1(X_1) \otimes S_2(X_2)$, or
3) a sum of SPFs: $S_1(X_1) \oplus S_2(X_2)$,

where R is a semiring, $X = \{X_1, \ldots, X_n\}$ is a set of variables with finite domains, and $\Phi = \{\phi_j\}$ is a set of constant and univariate functions.

Definition. A product node is *decomposable* iff the scopes of its children are disjoint. An SPF is *decomposable* iff all of its product nodes are decomposable.

The sum-product theorem

Decomposability is a simple condition that defines a class of expressive functions for which inference is tractable.



Theorem (sum-product theorem).
Every decomposable SPF S can be summed in time linear in its size.

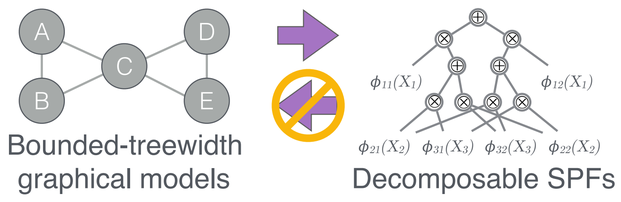
Proof: push outer summation to leaves.

- Sum nodes: $\oplus_X \oplus_i S_i(X_i) = \oplus_i (\oplus_{X_i} S_i(X_i) \otimes C)$
- Product nodes: $\underbrace{\oplus_X \otimes_i S_i(X_i)}_{\text{cost} = d^n} = \underbrace{\otimes_i \oplus_{X_i} S_i(X_i)}_{\text{cost} = d \cdot n}$
- Leaf nodes can be summed in constant time
- Sum S by summing each leaf node and then evaluating the remaining nodes bottom-up.

Where C is a constant and $i \in$ Children(node)

Corollary. Every SPF with n variables and treewidth bounded by a constant can be summed in time $O(n)$, but not every SPF that can be summed in time $O(n)$ has bounded treewidth.

Proof idea:



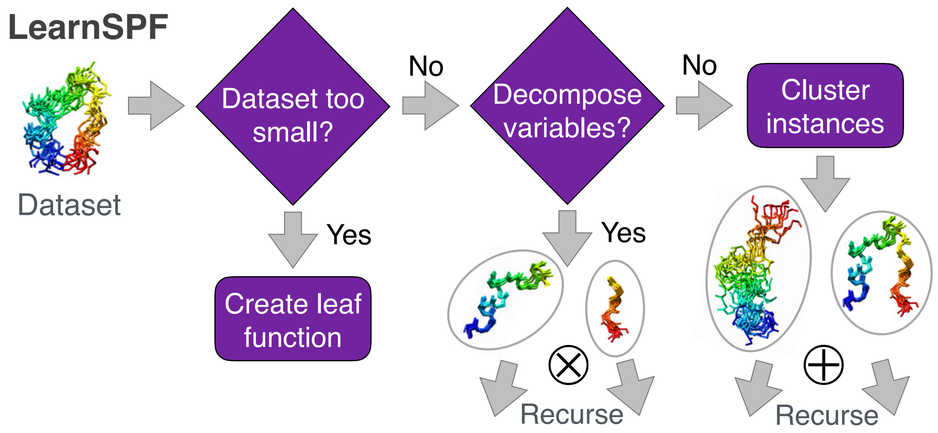
The SPT is the most recent step in a long line of work on tractable inference, which includes Darwiche (2001, 2003); Darwiche & Marquis (2002); Bacchus et al. (2002, 2009), Dechter & Mateescu (2007); Poon & Domingos (2011); and Gens & Domingos (2013).

Learning tractable representations with LearnSPF

From the sum-product theorem, identifying and exploiting decomposability is the key to learning a tractable model.

To decompose, an algorithm must find a partition of X into $\{X_1, X_2\}$ such that $\oplus_X S(X) \approx (\oplus_{X_1} S_1(X_1)) \otimes (\oplus_{X_2} S_2(X_2))$.

Based on this, LearnSPF is able to learn tractable, high-treewidth models in any semiring.



Specific choices for decomposition, clustering and leaf-creation subroutines depend on the domain. Instances with analogous decomposability structure can be clustered by virtually any algorithm, including naive Bayes and k-means. Correlation and independence tests can be used to identify decomposability.

Applications to specific semirings

Domain	Inference task	Semiring	Variables	Leaf functions
Logical inference	SAT	$(\mathbb{B}, \vee, \wedge, 0, 1)$	Boolean	Literals
	#SAT	$(\mathbb{N}, +, \times, 0, 1)$	Boolean	Literals
	MAX-SAT	$(\mathbb{N}_{-\infty}, \max, +, -\infty, 0)$	Boolean	Literals
Constraint satisfaction	CSPs	$(\mathbb{B}, \vee, \wedge, 0, 1)$	Discrete	Univariate constraints
	Fuzzy CSPs	$([0,1], \max, \min, 0, 1)$	Discrete	Univariate constraints
	Weighted CSPs	$(\mathbb{R}_{+,\infty}, \min, +, \infty, 0)$	Discrete	Univariate constraints
Probabilistic inference	Marginal	$(\mathbb{R}_+, +, \times, 0, 1)$	Discrete	Potentials
	MPE	$(\mathbb{R}_+, \max, \times, 0, 1)$	Discrete	Potentials
Continuous functions	Integration	$(\mathbb{R}_+, +, \times, 0, 1)$	Continuous	Univariate functions
	Optimization	$(\mathbb{R}_\infty, \min, +, \infty, 0)$	Continuous	Univariate functions
Relational databases	Unions of CQs	$(\mathbf{U}_m, \cup, \bowtie, \varnothing, \mathbf{1}_R)$	Sets of tuples	Unary tuples
	Provenance	$(\mathbb{N}[\mathbf{X}], +, \times, 0, 1)$	Discrete	K-relation tuples

Learning tractable knowledge bases

SPFs on the Boolean semiring correspond to negation normal form (NNF), and summation of an NNF F is $\bigvee_{x \in X} F(x) = \text{SAT}(F)$

Corollary (Darwiche, 2001)**.** The satisfiability of a decomposable NNF is decidable in time linear in its size.

Learned rule sets are typically encoded in large CNF knowledge bases (KBs), making reasoning over them highly intractable. In contrast, LearnSPF provides a method for directly learning large, complex KBs that are encoded in decomposable NNF and therefore support efficient querying.

Learning efficiently-optimizable nonconvex functions

SPFs can be extended to continuous (real) variables.

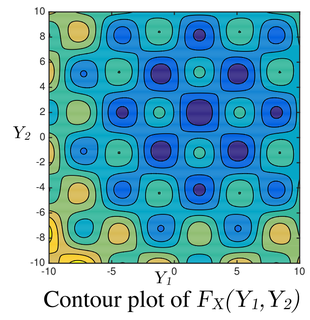
Corollary. The global minimum of a decomposable SPF on the min-sum semiring (i.e., a MSF) can be found in time linear in its size.

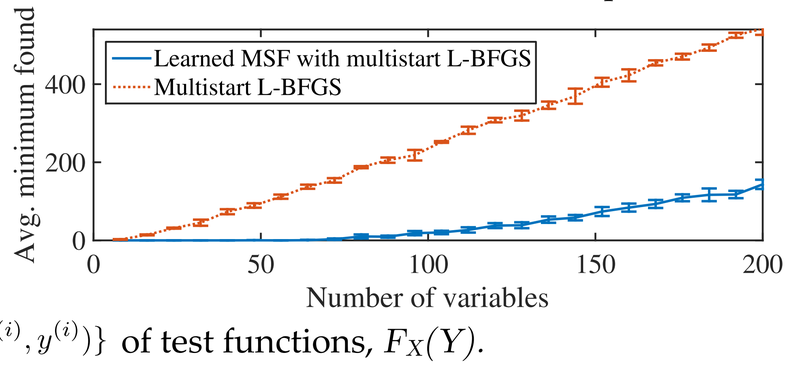
RDIS implicitly constructs a decomposable MSF. [Friesen & Domingos (2015)]

Typically, optimizing nonconvex functions is hard, and learning them is even harder; however, LearnSPF provides a method for effectively learning nonconvex functions that can then be efficiently optimized.

Experiment: learning and optimizing a decomposable MSF with LearnSPF versus learning and optimizing a nonconvex function.

- Problem is a continuous variant of structured prediction.
- LearnSPF used K-means to cluster and correlation to decompose.





- Sampled dataset $\{(x^{(i)}, y^{(i)})\}$ of test functions, $F_X(Y)$.
  - Parameters X dictate the decomposability of the variables Y.
  - Dataset: 300 train, 50 test, with labels $y^{(i)} = \arg\min_{y \in Y} F_{x^{(i)}}(y)$.

Department of Computer Science and Engineering, University of Washington