PSDDs

Kushagra Chandak

15th May, 2020

Introduction to PSDDs

- Circuit (DAG) representation of discrete joint probability distributions over binary variables. (Complete and canonical)
- Used when learning with domain constraints.
  - Data + constraints $\xrightarrow{\text{Learn}}$ Model
  - Learn a statistical model/distribution that assigns zero probability to instantiations (data) that violate the constraints.
- Useful when the probability space is structured.
  - Permutations, rankings, simple paths, other combinatorial objects.

Structured Probability Spaces

Courses:

- Logic (L)
- Knowledge Representation (K)
- Probability (P)
- Artificial Intelligence (A)

Data

L	K	P	A	Students
0	0	1	0	6
0	0	1	1	54
0	1	1	1	10
1	0	0	0	5
1	0	1	0	1
1	0	1	1	0
1	1	0	0	17
1	1	1	0	4
1	1	1	1	3

Prior Knowledge

- Must take at least one of Probability or Logic.
- Probability is a prerequisite for AI.
- The prerequisites for KR is either AI or Logic.

Source: Adnan Darwiche, Representation learning workshop, Simons Institute

Structured Probability Spaces

unstructured

L	K	P	A
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

- Must take at least one of Probability or Logic.
- Probability is a prerequisite for AI.
- The prerequisites for KR is either AI or Logic.

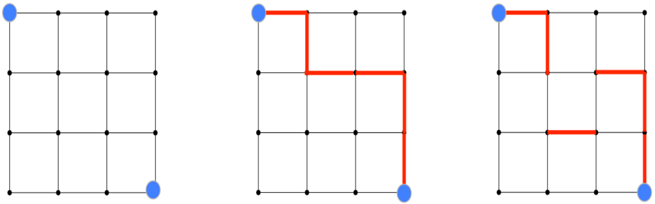
7 out of 16 instantiations are impossible

structured

L	K	P	A
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

Source: Adnan Darwiche, Representation learning workshop, Simons Institute

Structured Probability Spaces



Good variable assignment (represents route)

184

Bad variable assignment (does not represent route)

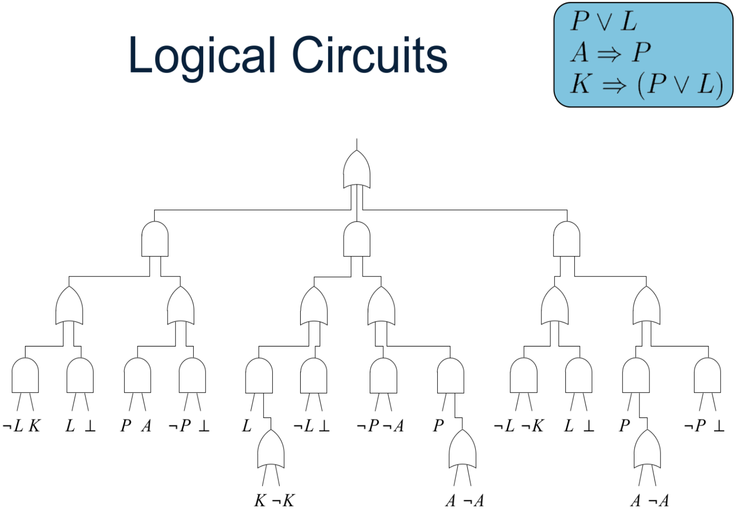
16,777,032

Space easily encoded in logical constraints

Unstructured probability space: $184 + 16{,}777{,}032 = 2^{24}$

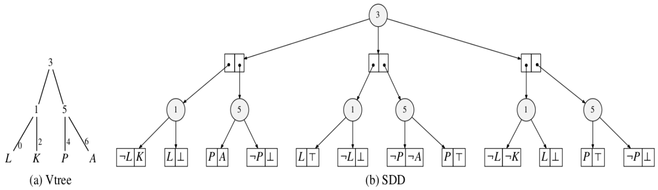
Source: Adnan Darwiche, Representation learning workshop, Simons Institute

Underlying circuit of PSDD: SDD



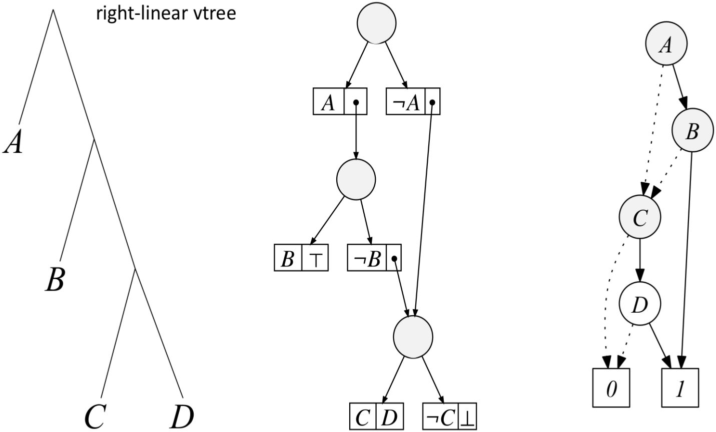
Source: Adnan Darwiche, Representation learning workshop, Simons Institute

Underlying circuit of PSDD: SDD



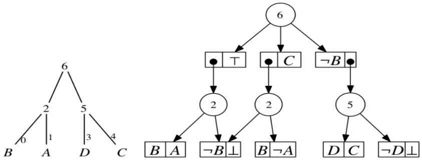
- Domain constraints (base) represented by SDD. Determined by a vtree: full binary tree with leaves corresponding to the variables.
- A decision $((p_1 \wedge s_1) \vee \ldots \vee (p_n \wedge s_n))$ or a terminal ($\top$ or $\bot$) node. (p_i, s_i) is called an element. A decision node *respects* a vtree node.
- Vtree basically describes the structure of the SDD and what variables will form leaves.
- Primes of a decision node are consistent, mutually exclusive and exhaustive.
- Structure that induces infinitely many probability distributions. Parameterize SDD $\rightarrow$ PSDD.

OBDDs are SDDs



Source: Adnan Darwiche, INFORMS

SDDs: Basing Decisions on Sentences



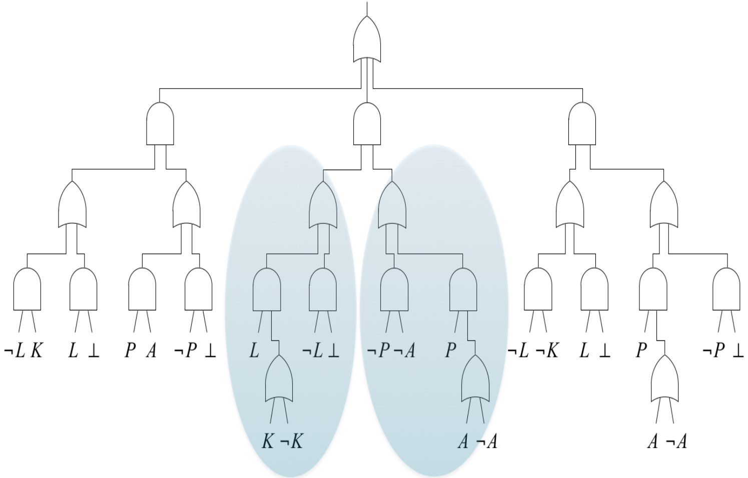
(a) vtree (b) Graphical depiction of an SDD

- Boolean functions. Generalization of OBDDs in two ways:
  - Characterstic of OBDDs: a variable order (linear). Characterstic of SDDs: a vtree (tree). SDD with a right linear vtree: OBDD.
  - Decision nodes of SDD: may not be binary. Branch over sentences.
- SDDs maintain key properties of OBDDs (canonicity and polytime apply): Due to XY partition.
- **XY partition:** Write $f(\mathbf{X}, \mathbf{Y})$ as $h_1(\mathbf{X})g_1(\mathbf{Y}) + \ldots + h_n(\mathbf{X})g_n(\mathbf{Y})$, where h_is form a partition (mutually exclusive and exhaustive; no h_i false). $\{(h_1, g_1), \ldots, (h_n, g_n)\}$: h_i: primes, g_i: subs.
- For OBDDs, where $\mathbf{X} = \{X\}$, XY partition is called Shannon decomposition.

More on XY partition and SDDs

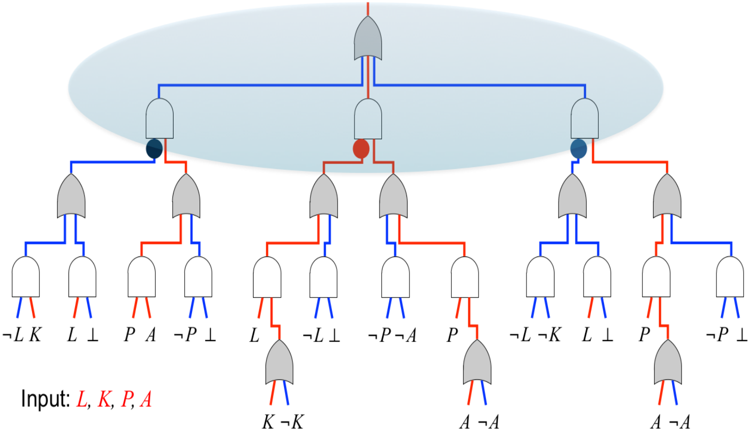
- SDDs are compressed if no two subs are equal. If equal subs, say $g1 = g2$, we can compress as $\{(h_1 + h_2, g_1), \ldots, (h_n, g_n)\}$. Every $f(\mathbf{X}, \mathbf{Y})$ has a unique compressed XY partition.
- Constructing SDD from Boolean function: Given a Boolean function, for a fixed vtree, the variables are partitioned (acc. to vtree) and we have a unique compressed XY partition which gives the root node of SDD. Recursively partition primes and subs using the vtree.
- SDDs more succinct than OBDDs: if there's an exponential sized OBDD for a variable order, dissecting it gives linear SDD.
- Search of OBDDs: searching a variable order space (permutations). Search of SDDs: searching in vtree space (trees).

Properties of SDD: Decomposibility



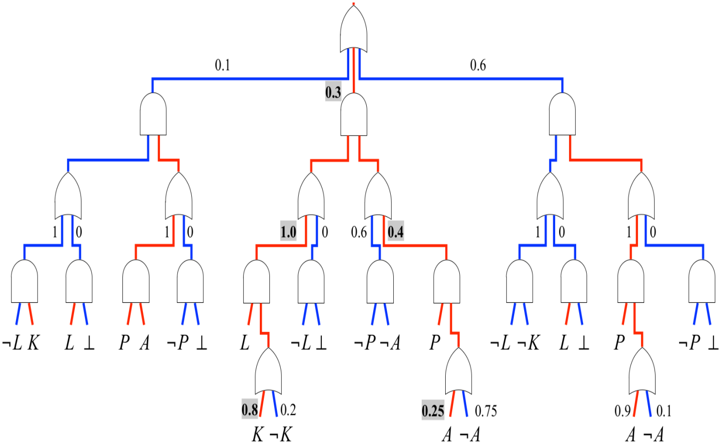
Source: Adnan Darwiche, Representation learning workshop, Simons Institute

Properties of SDD: Determinism



Source: Adnan Darwiche, Representation learning workshop, Simons Institute

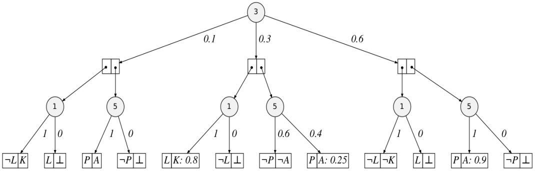
PSDD



Input: L, K, P, A $\Pr(L,K,P,A) = 0.3 \times 1.0 \times 0.8 \times 0.4 \times 0.25 = 0.024$

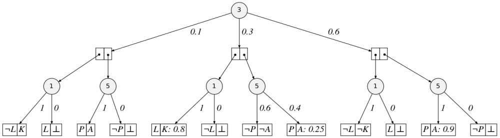
Source: Adnan Darwiche, Representation learning workshop, Simons Institute

Syntax and Semantics of PSDD



- PSDDs are based on normalized SDDs. Node n associated to vtree node v
  - $n =$ terminal node, then v is a leaf node containing variable of n.
  - $n =$ decision node, then primes and subs will be in left and right child of v.
  - $n =$ root SDD node, $v =$ root vtree node.
- **Syntax:**
  - For each decision node $(p_1, s_1), \ldots (p_k, s_k)$ and prime p_i, a positive parameter θ_i is supplied such that $\theta_1 + \ldots + \theta_k = 1$ and $\theta_i = 0$ iff $s_i = \bot$.
  - For each terminal node $\top$, a positive paramter θ is supplied such that $0 < \theta < 1$.

More Definitions



- **PSDD local (node) distribution:** $\Pr_n$ over variables of vtree v
  - If $n =$ terminal node and v has variable X, then

n	$\Pr_n(X)$	$\Pr_n(\neg X)$
$X:\theta$	θ	$1-\theta$
$\perp$	0	0
X	1	0
$\neg X$	0	1

  - If $n =$ decision node $(p_1, s_1, \theta_1), \ldots, (p_k, s_k, \theta_k)$ and v has left variables $\mathbf{X}$ and right variables $\mathbf{Y}$, then

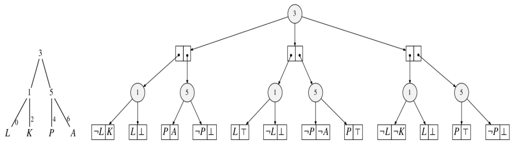
$$\Pr_n(\mathbf{xy}) = \Pr_{p_i}(\mathbf{x}).\Pr_{s_i}(\mathbf{y}).\theta_i$$

for i where $\mathbf{x} \models p_i$. E.g. $n = (\neg P, \neg A)(P, \perp)$

$\mathbf{x}$	$\mathbf{y}$	$\Pr_{p_i}(\mathbf{x})$	$\Pr_{s_i}(\mathbf{y})$	θ_i	$\Pr_n(\mathbf{xy})$
P	A	1	0.25	0.4	0.1
P	$\neg A$	1	0.75	0.4	0.3
$\neg P$	A	1	0	0.6	0.0
$\neg P$	$\neg A$	1	1	0.6	0.6

- **Parameter Semantics:** $\theta_i = \Pr_n(p_i)$

Relationship between local and global distributions



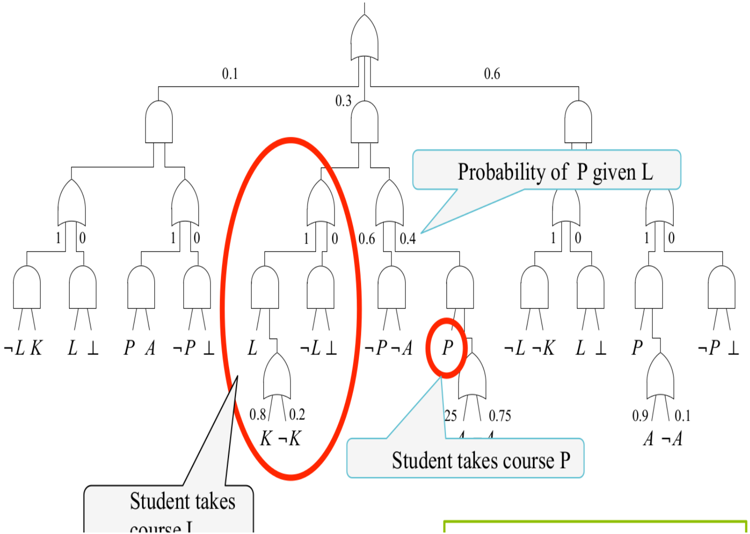
(a) Vtree (b) SDD

- **Context:** Boolean formula that captures all variable instantiations under which the decision diagram will branch to a node.
  - $(p_1, s_1), \ldots, (p_k, s_k)$: Elements on some path from SDD root, r, to node n (p_k or s_k).
  - $p_1 \wedge \ldots \wedge p_k$: sub-context for n and is feasible iff $s_i \neq \bot$. (To reach node n, all the primes on a path to n must be satisfied.)
  - Context: Disjunction (OR) of all sub-contexts and feasible iff some sub-context is feasible.
  - E.g.: Node normalized for $v = 5$ have contexts $\neg L \wedge K, L, \neg L \wedge \neg K$
  - Properties:
    - A node is implied by its context and underlying SDD.
    - Contexts are mutually exclusive and exhaustive for nodes normalized for the same vtree node.
    - Sub-contexts of a node are mutually exclusive.
    - A context/sub-context is feasible if it has a strictly positive probability.

Relationship between local and global distributions

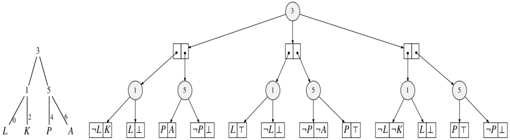
- Global iterpretation to node (local) distribution: γ_n = feasible context/sub-context, then $\mathrm{Pr}_n(.) = \mathrm{Pr}_r(.|\gamma_n)$.
- Global interpretation to parameters:
  - n = terminal node, then $\theta = \mathrm{Pr}_r(X|\gamma_n)$
  - n = decision node, then $\theta_i = \mathrm{Pr}_r(p_i|\gamma_n)$
  - Intuitively, θ_i is the probability of prime p_i given that the decision of node n has been implied.
- PSDD as decision diagram: From top-down perspective, a decision node represents a choice between its primes (sentences). Generalizes decision trees/BDDs, which only branch to a single variable.

Interpretation of PSDD Parameters



Source: Adnan Darwiche, Representation learning workshop, Simons Institute

PSDD Independence



(a) Vtree (b) SDD

- BayesNet: Independences conditioned on variables. E.g.: A and L are ind. given P. K and P are ind. given AL.
- PSDD: Indpendences conditioned on Boolean sentences (contexts):
  - **Independence I:** Probability of a prime is ind. of *a* sub-context once the context is known. (Can replace γ_n with β_n)

$$\Pr_r(p_i|\gamma_n, \beta_n) = \Pr_r(p_i|\gamma_n) = \Pr_r(p_i|\beta_n) = \theta_i$$

  - **Independence II:** For vtree node v, variables inside v are ind. of those outside v given γ_v.
    E.g.: for $v = 5$, using context L, variables PA and LK are ind. given L

Reasoning with PSDDs

- Calculate $\Pr_r(\mathbf{e})$ and $\Pr_r(X|\mathbf{e})$. Runtime: Linear in PSDD size.
- $\Pr_n(\mathbf{e}_v) = \sum_{i=1}^{k} \Pr_{p_i}(\mathbf{e}_l).\Pr_{s_i}(\mathbf{e}_r).\theta_i$ (bottom-up)
- $\Pr_r(X, e_{\bar{v}}) = \sum_{i=1}^{k} \Pr_{n_i}(X).\Pr_r(\gamma_{n_i}, \mathbf{e}_{\bar{v}})$
  $\Pr_r(X, e_{\bar{v}}) = \Pr_r(X, e)$ if $\neg X$ is not satisfied by **e**, otherwise $\Pr_r(X, e) = 0$.
- To compute $\Pr_r(\gamma_n, \mathbf{e}_{\bar{v}})$:
  probability of a sub-context = multiplication of parameters.
  context probability = summation of sub-context probabilities (since they are mutually exclusive).