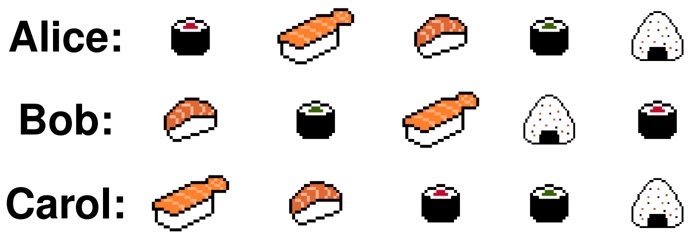
Scalable Learning of Probabilistic Circuits

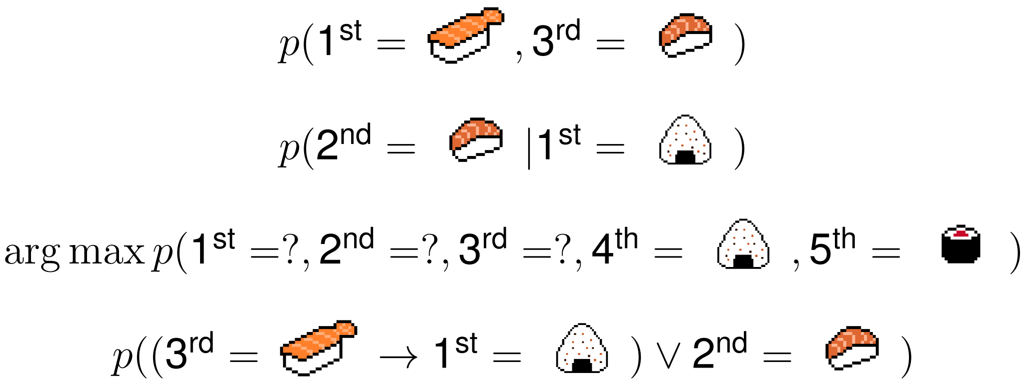
Motivation

Given a selection of sushi...

...and people's preferences...



...how can we model this as a probability distribution...



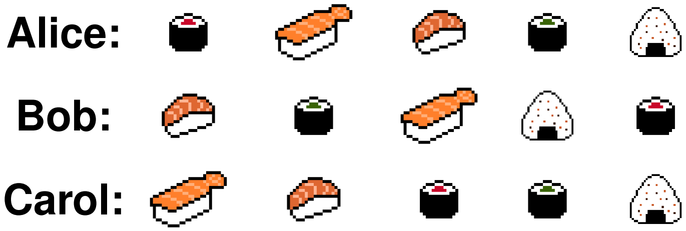
...and extract meaningful queries from it?

Kamishima [2003]

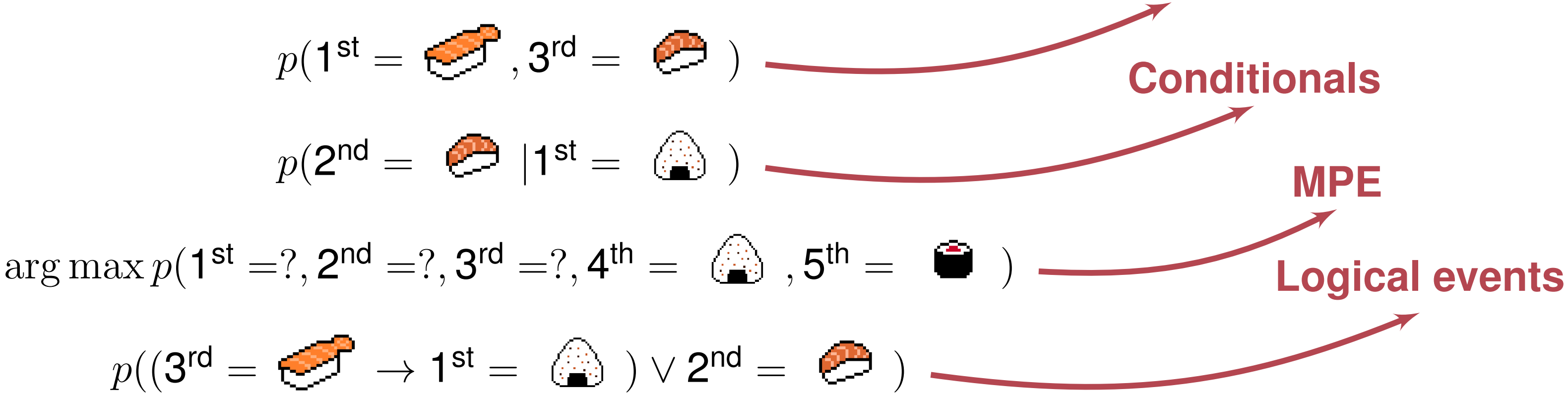
Motivation

Given a selection of sushi...

...and people's preferences...



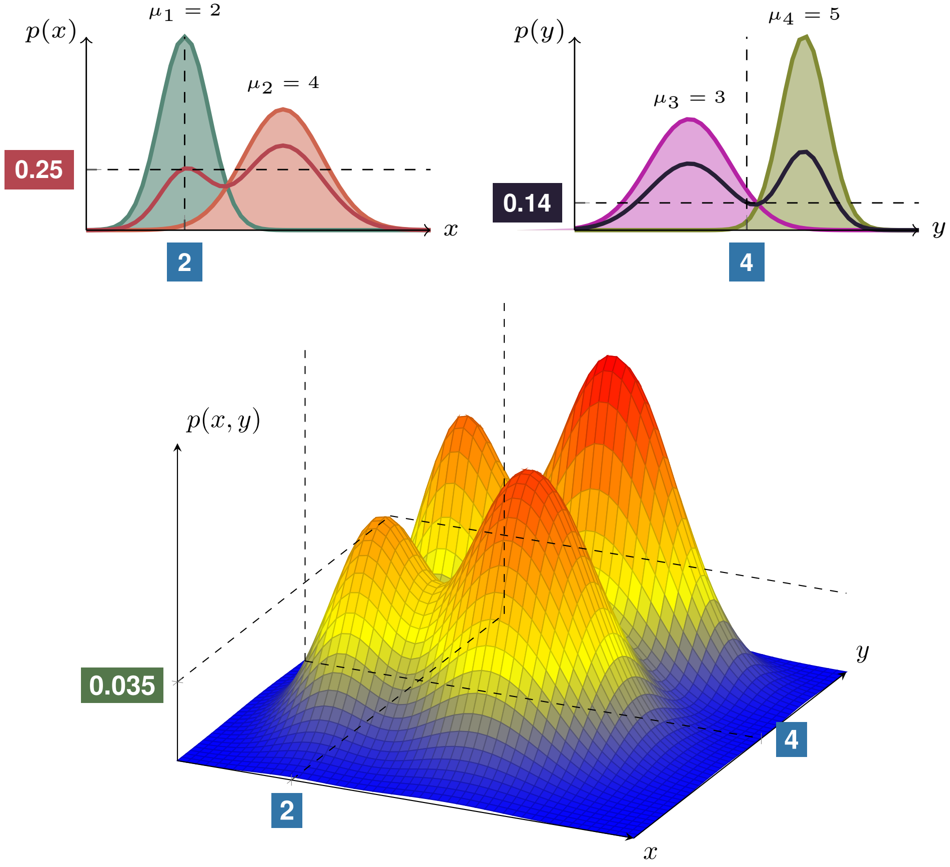
...how can we model this as a probability distribution...

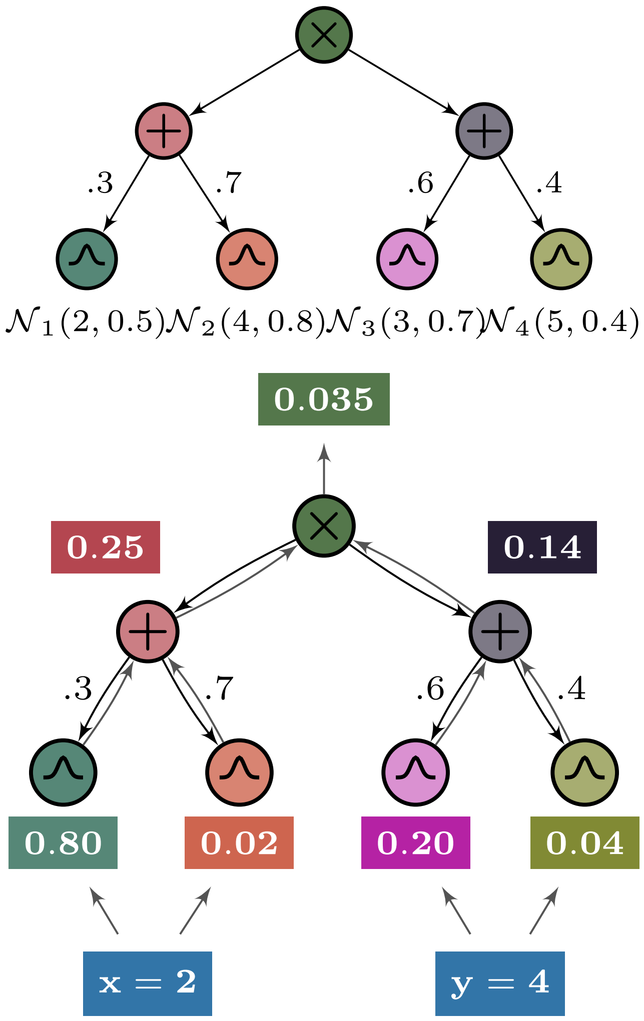


...and extract meaningful queries from it?

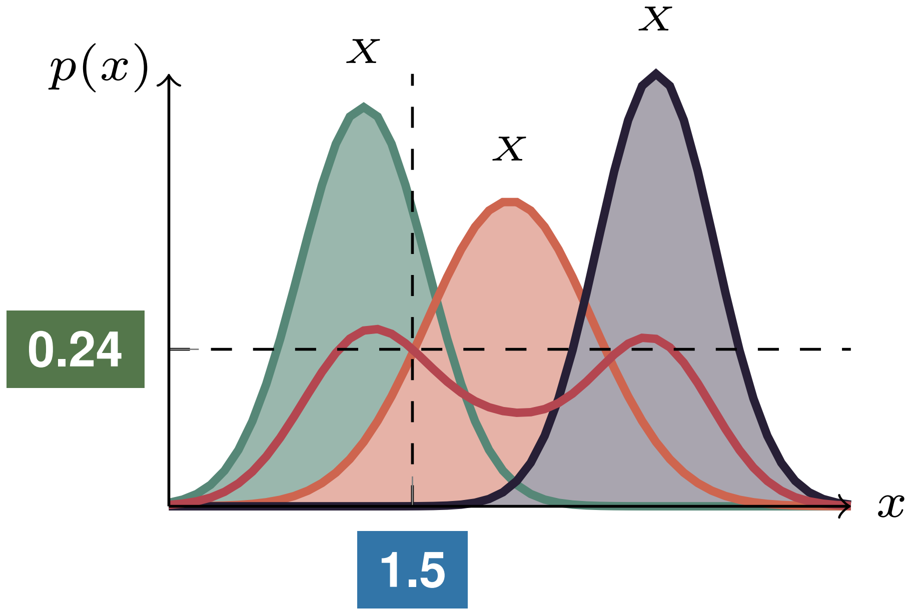
Kamishima [2003]

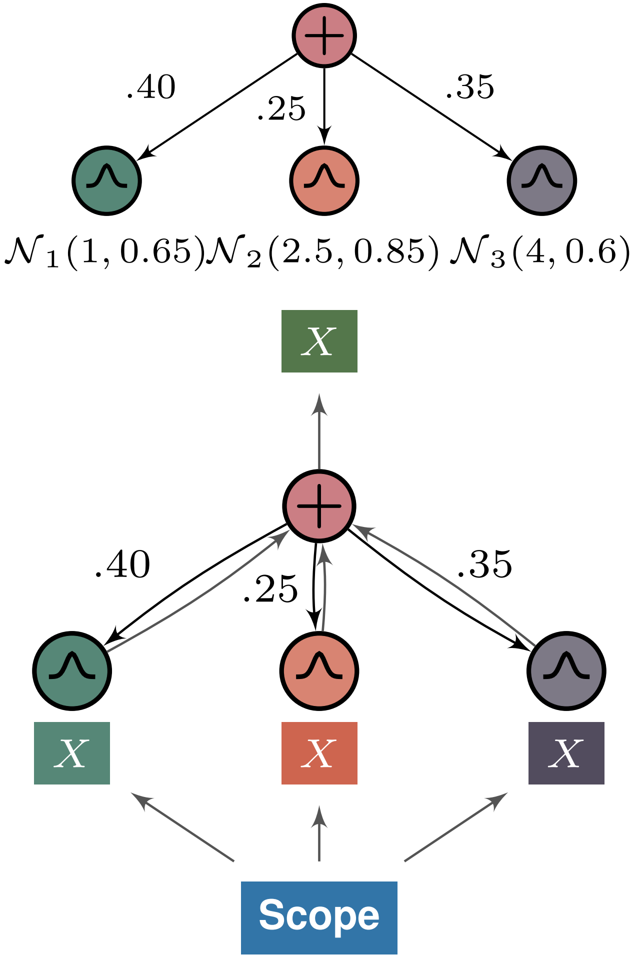
Probabilistic Circuits





Probabilistic Circuits – Smoothness

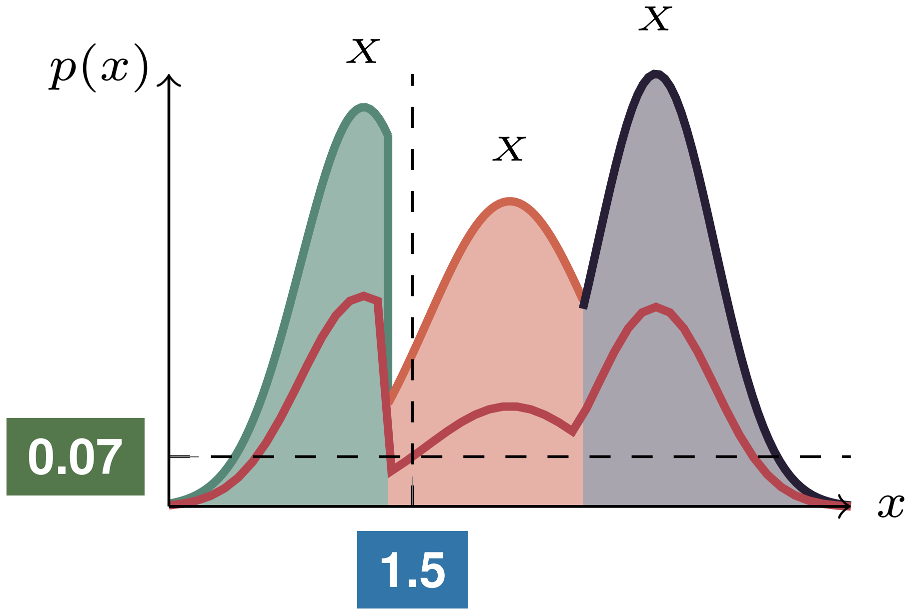


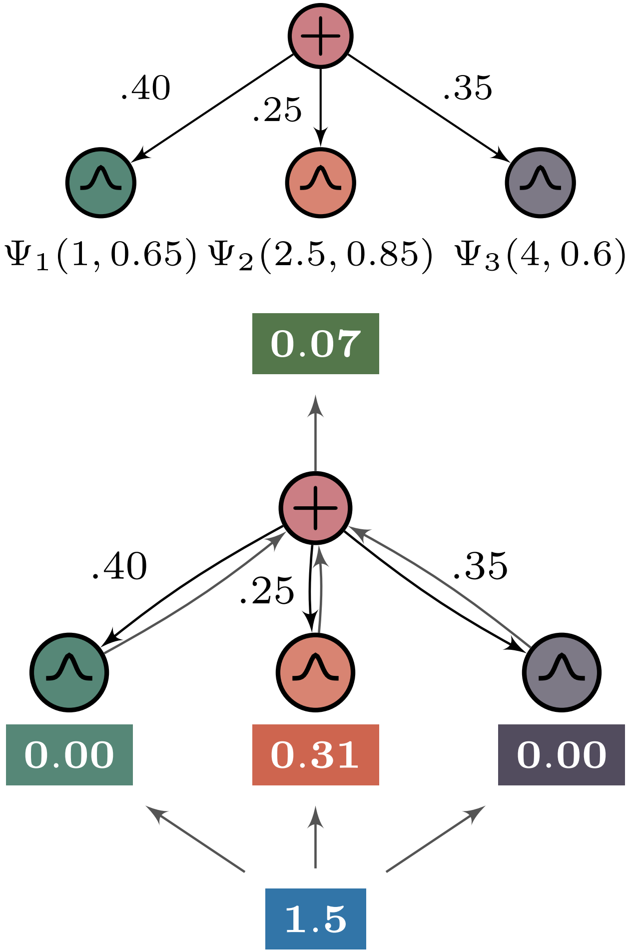


Definition 1 (Smoothness)**.**
Every sum node child mentions the _*same*_ *variables.*

Darwiche [2001a]

Probabilistic Circuits – Determinism

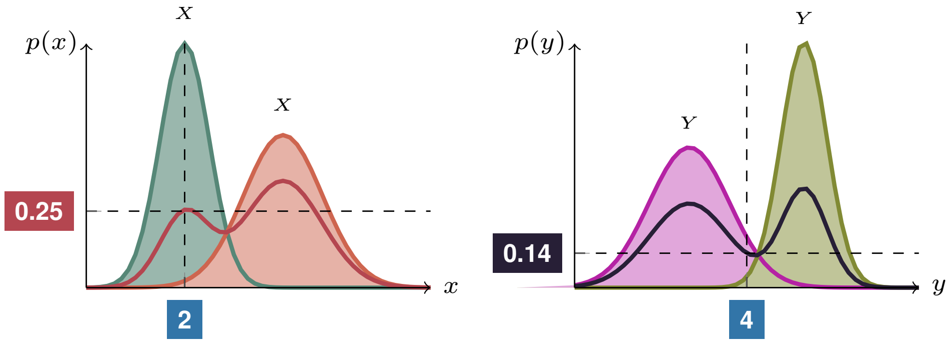


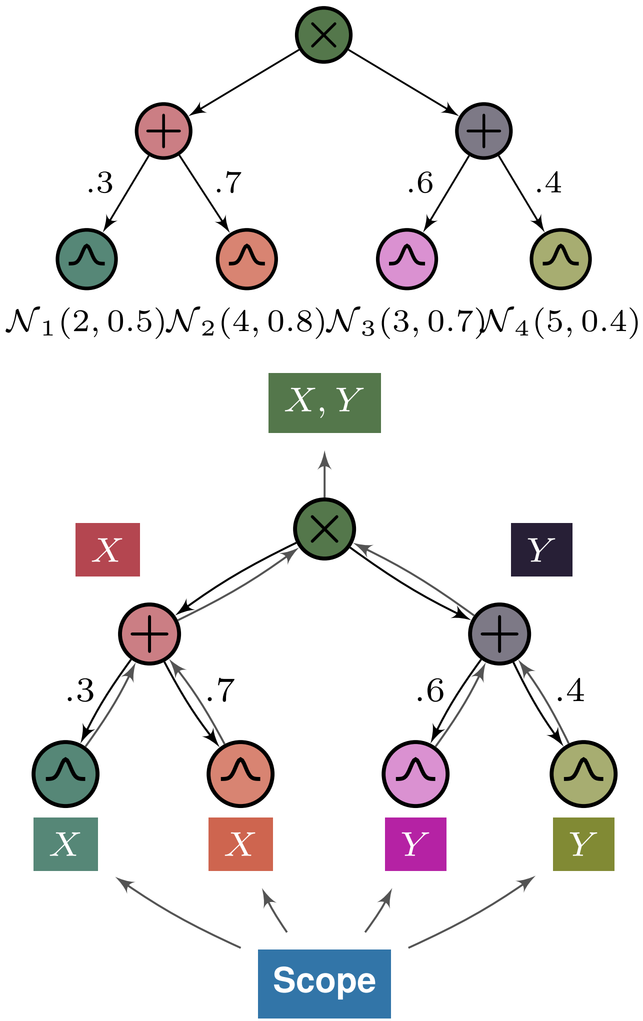


Definition 2 (Determinism)**.**
<u>At most one</u> sum node child has a positive value.

Darwiche [2001a]

Probabilistic Circuits – Decomposability

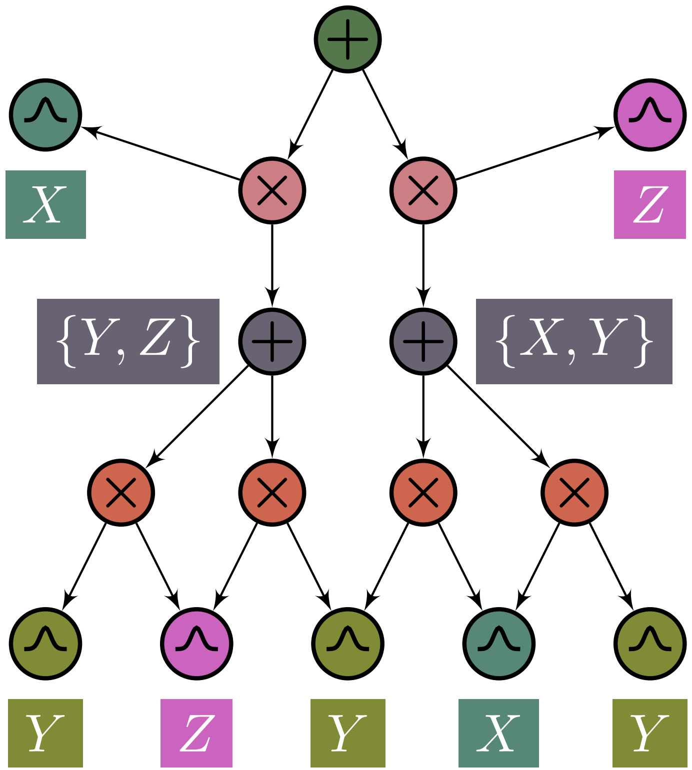


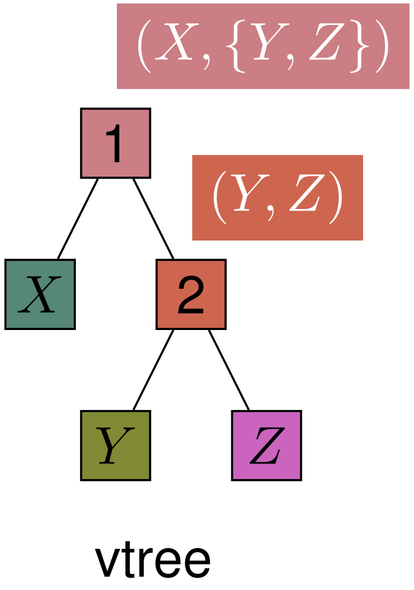


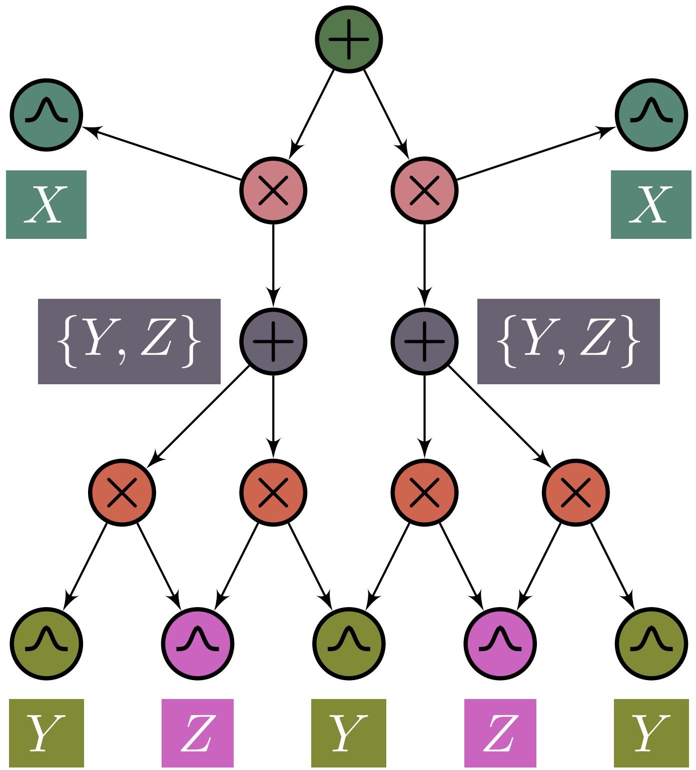
Definition 3 (Decomposability)**.**
Every product node child mentions different variables.

Darwiche [1999, 2001b]

Probabilistic Circuits – Structured Decomposability







Definition 4 (Structured decomposability). *Every product node follows a vtree decomposition.*

Pipatsrisawat and Darwiche [2008]

Probabilistic Circuits – Tractability

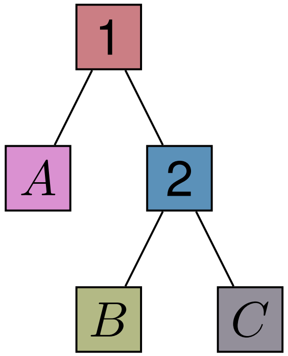
Query	+Sm?	+Dec?	+Det?	+Str Dec?
Evidence	✓	✓	✓	✓
Marginals	✗	✓	✓	✓
Conditionals	✗	✓	✓	✓
MPE	✗	✗	✓	✓
Shannon Entropy*	✗	✗	✓	✓
Rényi Entropy*	✗	✗	✓	✓
Cross Entropy*	✗	✗	✗	✓
Kullback-Leibler Div*	✗	✗	✗	✓
Rényi's Alpha Div*	✗	✗	✗	✓
Cauchy-Schwarz Div*	✗	✗	✗	✓
Logical Events	✗	✗	✗	✓
Mutual Information*	✗	✗	✗	✓

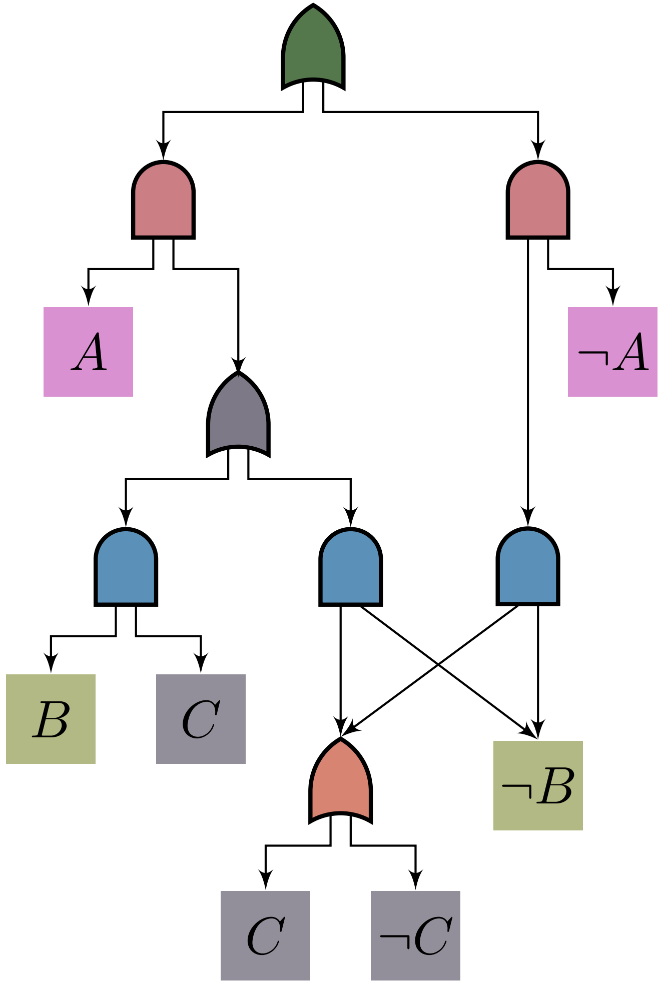
Vergari et al. [2021], Poon and Domingos [2011], Peharz et al. [2016]

Probabilistic Circuits – Logic Circuits

A	B	C	$\phi(\mathbf{x})$
0	0	0	1
1	0	0	1
0	1	0	0
1	1	0	0
0	0	1	1
1	0	1	1
0	1	1	0
1	1	1	1

$$\phi(A, B, C) = (A \vee B) \wedge (\neg B \vee C)$$



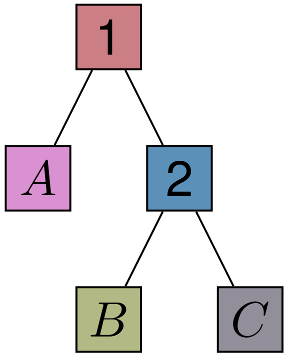


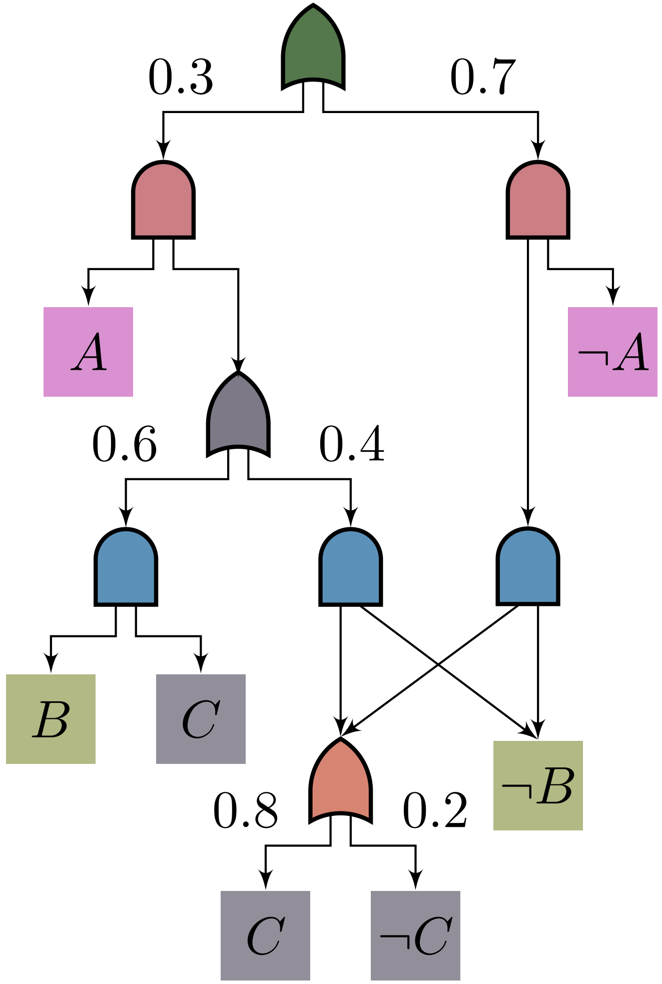
Darwiche [2011]

Probabilistic Circuits – Support

A	B	C	$\phi(\mathbf{x})$	$p(\mathbf{x})$
0	0	0	1	0.140
1	0	0	1	0.024
0	1	0	0	0.000
1	1	0	0	0.000
0	0	1	1	0.560
1	0	1	1	0.096
0	1	1	0	0.000
1	1	1	1	0.180

$$\phi(A, B, C) = (A \vee B) \wedge (\neg B \vee C)$$





Kisa et al. [2014]

Learning Probabilistic Circuits

Divide-and-Conquer Approaches (DIV)

- Usually recursive;
- Splits data by similarity and stat dep;
- Stat dep usually costly;
- Usually tree-shaped.

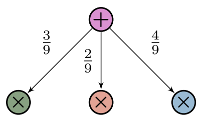
Incremental Approaches (INCR)

- Requires an initial circuit;
- Grows from local transformations;
- Local transformations preserve properties;
- Searching for candidates to transform is costly.

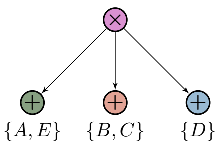
Random Approaches (RAND)

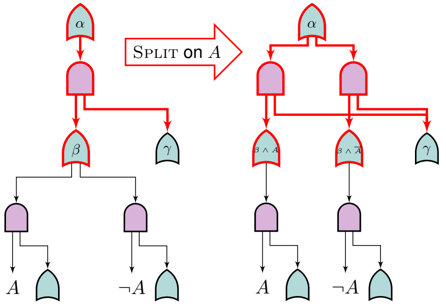
- Fast;
- Randomly generates circuits;
- Data blind and data guided approaches exist;
- Usually relies on many hyperparams;
- Worse performance.

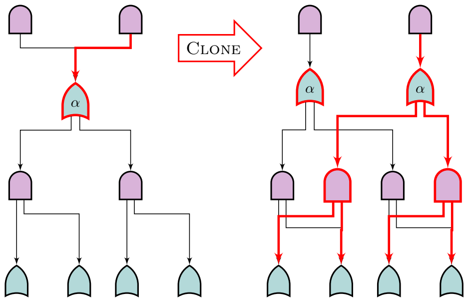
A	B	C	D	E
0	1	0	0	1
1	0	1	1	1
1	1	0	1	1
0	0	1	0	0
1	1	0	1	0
0	1	1	0	1
1	0	1	1	1
1	1	0	0	0
0	1	1	0	1

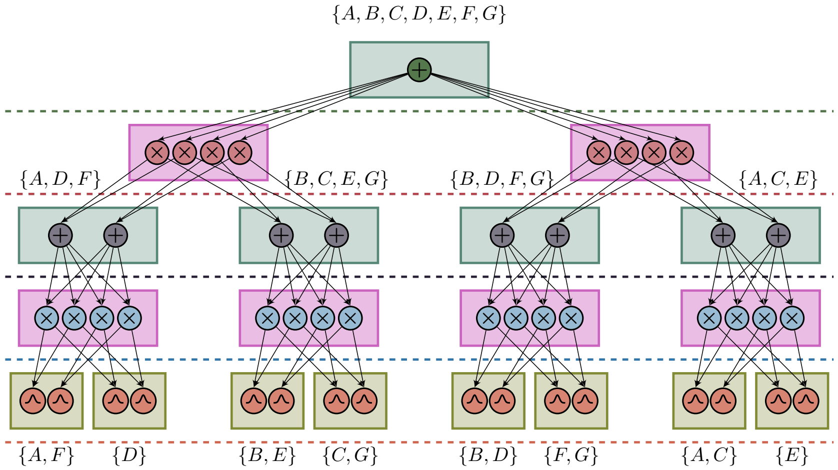


A	B	C	D	E
0	1	0	0	1
1	0	1	1	1
1	1	0	1	1
0	0	1	0	0
1	1	0	1	0
0	1	1	0	1
1	0	1	1	1
1	1	0	0	0
0	1	1	0	1









Learning Probabilistic Circuits

Divide-and-Conquer Approaches (DIV)

- Usually recursive;
- Splits data by similarity and stat dep;
- Stat dep usually costly;
- Usually tree-shaped.

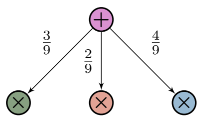
Incremental Approaches (INCR)

- Requires an initial circuit;
- Grows from local transformations;
- Local transformations preserve properties;
- Searching for candidates to transform is costly.

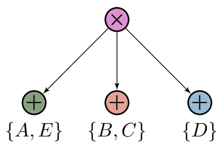
Random Approaches (RAND)

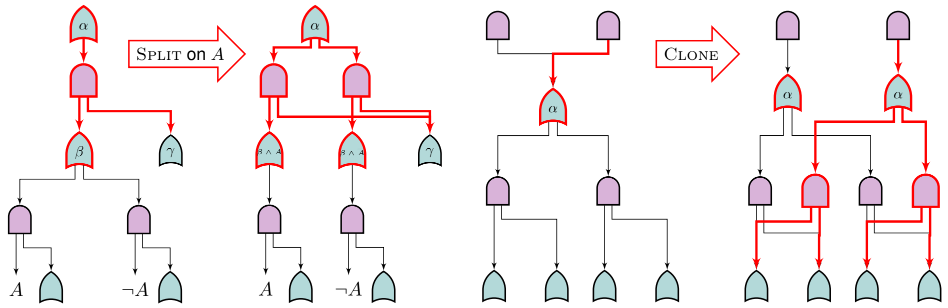
- Fast;
- Randomly generates circuits;
- Data blind and data guided approaches exist;
- Usually relies on many hyperparams;
- Worse performance.

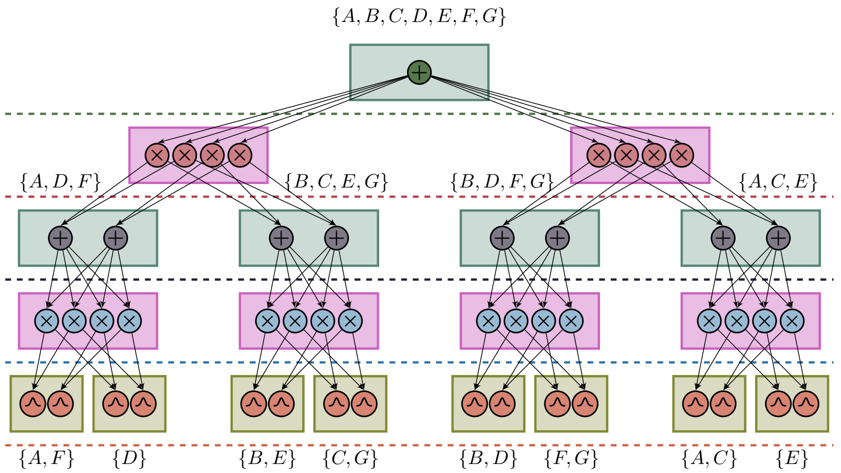
A	B	C	D	E
0	1	0	0	1
1	0	1	1	1
1	1	0	1	1
0	0	1	0	0
1	1	0	1	0
0	1	1	0	1
1	0	1	1	1
1	1	0	0	0
0	1	1	0	1



A	B	C	D	E
0	1	0	0	1
1	0	1	1	1
1	1	0	1	1
0	0	1	0	0
1	1	0	1	0
0	1	1	0	1
1	0	1	1	1
1	1	0	0	0
0	1	1	0	1







Learning Probabilistic Circuits

Divide-and-Conquer Approaches (DIV)

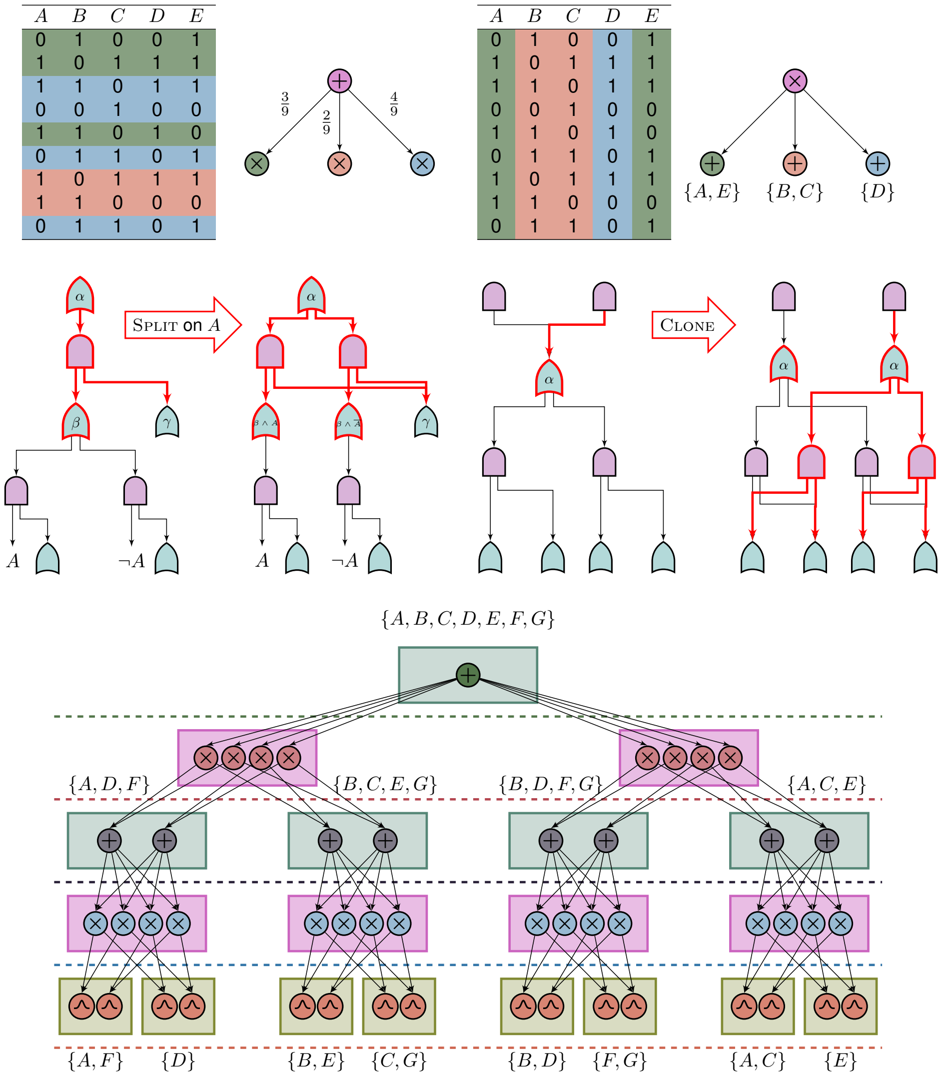
- Usually recursive;
- Splits data by similarity and stat dep;
- Stat dep usually costly;
- Usually tree-shaped.

Incremental Approaches (INCR)

- Requires an initial circuit;
- Grows from local transformations;
- Local transformations preserve properties;
- Searching for candidates to transform is costly.

Random Approaches (RAND)

- Fast;
- Randomly generates circuits;
- Data blind and data guided approaches exist;
- Usually relies on many hyperparams;
- Worse performance.



A	B	C	D	E
0	1	0	0	1
1	0	1	1	1
1	1	0	1	1
0	0	1	0	0
1	1	0	1	0
0	1	1	0	1
1	0	1	1	1
1	1	0	0	0
0	1	1	0	1

A	B	C	D	E
0	1	0	0	1
1	0	1	1	1
1	1	0	1	1
0	0	1	0	0
1	1	0	1	0
0	1	1	0	1
1	0	1	1	1
1	1	0	0	0
0	1	1	0	1

Learning Probabilistic Circuits – Where are we right now?

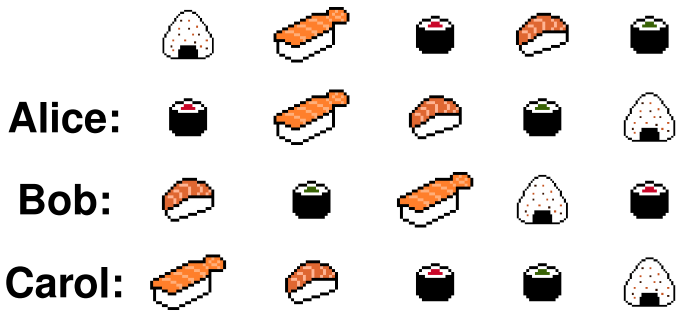
Name	Class	Time Complexity	# hyperparams	Accepts logic?	Sm?	Dec?	Det?	Str Dec?	$\{0,1\}$?	$\mathbb{N}$?	$\mathbb{R}$?	Reference
LearnSPN	DIV	$\begin{cases} \mathcal{O}(nkmc) & \text{, if sum} \\ \mathcal{O}(nm^3) & \text{, if product} \end{cases}$	≥ 2	✗	✓	✓	✗	✗	✓	✓	✓	Gens and Domingos [2013]
ID-SPN	DIV	$\begin{cases} \mathcal{O}(nkmc) & \text{, if sum} \\ \mathcal{O}(nm^3) & \text{, if product} \\ \mathcal{O}(ic(rn+m)) & \text{, if input} \end{cases}$	$\geq 2+3$	✗	✓	✓	✗	✗	✓	✓	✗	Rooshenas and Lowd [2014]
Prometheus	DIV	$\begin{cases} \mathcal{O}(nkmc) & \text{, if sum} \\ \mathcal{O}(m(\log m)^2) & \text{, if product} \end{cases}$	≥ 1	✗	✓	✓	✗	✗	✓	✓	✓	Jaini et al. [2018a]
LearnPSDD	INCR	$\begin{cases} \mathcal{O}(m^2) & \text{, top-down vtree} \\ \mathcal{O}(m^4) & \text{, bottom-up vtree} \\ \mathcal{O}(i\lvert\mathcal{C}\rvert^2) & \text{, circuit structure} \end{cases}$	1	✓	✓	✓	✓	✓	✓	✗	✗	Liang et al. [2017]
Strudel	INCR	$\begin{cases} \mathcal{O}(m^2 n) & \text{, CLT + vtree} \\ \mathcal{O}(i(\lvert\mathcal{C}\rvert n+m^2)) & \text{, circuit structure} \end{cases}$	1	✓	✓	✓	✓	✓	✓	✗	✗	Dang et al. [2020]
RAT-SPN	RAND	$\mathcal{O}(rd(s+l))$	4	✗	✓	✓	✗	✗	✓	✓	✓	Peharz et al. [2020]
XPC	RAND	$\mathcal{O}(i(t+kn)+ikm^2n)$	3	✗	✓	✓	✓	✓	✓	✗	✗	Mauro et al. [2021]
SamplePSDD	RAND	$\begin{cases} \mathcal{O}(m) & \text{, random vtree} \\ \mathcal{O}(kc\log c+\log_2^2 k) & \text{, per call} \end{cases}$	1	✓	✓	✓	✓	✓	✓	✗	✗	Geh and Mauá [2021]
LearnRP	RAND	$\begin{cases} \mathcal{O}(m^2) & \text{, top-down vtree} \\ \mathcal{O}(m^4) & \text{, bottom-up vtree} \\ \mathcal{O}(knm) & \text{, per call} \end{cases}$	0	✗	✓	✓	✗	✓	✓	✓	✓	To appear

Learning Probabilistic Circuits – Where are we right now?

Name	Class	Time Complexity	# hyperparams	Accepts logic?	Sm?	Dec?	Det?	Str Dec?	$\{0,1\}$?	$\mathbb{N}$?	$\mathbb{R}$?	Reference
LearnSPN	DIV	$\mathcal{O}(nkmc)$, if sum $\mathcal{O}(nm^3)$, if product	≥ 2	✗	✓	✓	✗	✗	✓	✓	✓	Gens and Domingos [2013]
ID-SPN	DIV	$\mathcal{O}(nkmc)$, if sum $\mathcal{O}(nm^3)$, if product $\mathcal{O}(ic(rn+m))$, if input	$\geq 2+3$	✗	✓	✓	✗	✗	✓	✓	✗	Rooshenas and Lowd [2014]
Prometheus	DIV	$\mathcal{O}(nkmc)$, if sum $\mathcal{O}(m(\log m)^2)$, if product	≥ 1	✗	✓	✓	✗	✗	✓	✓	✓	Jaini et al. [2018a]
LearnPSDD	INCR	$\mathcal{O}(m^2)$, top-down vtree $\mathcal{O}(m^4)$, bottom-up vtree $\mathcal{O}(i\|\mathcal{C}\|^2)$, circuit structure	1	✓	✓	✓	✓	✓	✓	✗	✗	Liang et al. [2017]
Strudel	INCR	$\mathcal{O}(m^2 n)$, CLT + vtree $\mathcal{O}(i(\|\mathcal{C}\|n+m^2))$, circuit structure	1	✓	✓	✓	✓	✓	✓	✗	✗	Dang et al. [2020]
RAT-SPN	RAND	$\mathcal{O}(rd(s+l))$	4	✗	✓	✓	✗	✗	✓	✓	✓	Peharz et al. [2020]
XPC	RAND	$\mathcal{O}(i(t+kn)+ikm^2n)$	3	✗	✓	✓	✓	✓	✓	✗	✗	Mauro et al. [2021]
⇒ SamplePSDD	RAND	$\mathcal{O}(m)$, random vtree $\mathcal{O}(kc\log c+\log_2^2 k)$, per call	1	✓	✓	✓	✓	✓	✓	✗	✗	Geh and Mauá [2021]
LearnRP	RAND	$\mathcal{O}(m^2)$, top-down vtree $\mathcal{O}(m^4)$, bottom-up vtree $\mathcal{O}(knm)$, per call	0	✗	✓	✓	✗	✓	✓	✓	✓	To appear

A Logical Perspective

Motivation



If we assume

- n sushi types,
- k sized rankings with $k \leq n$,
- X_{ij} binary variables; i is sushi type, j is position in ranking;

then the total number of possible assignments of the $n \cdot k$ variables is 2^{nk}...

...but many of these are zero probability assignments!

If we can embed total ranking constraints...

...we go down to k! total assignments!

Takeaway: models which exploit domain knowledge are much more efficient!

Example:

$n = 3,\ k = 3$

X_{11}	X_{12}	X_{13}	X_{21}	$\cdots$	X_{33}	$p(\mathbf{x}) > 0$
0	0	0	0	0	0	0
1	0	0	0	0	0	0
0	1	0	0	0	0	0
1	1	0	0	0	0	0
0	0	1	0	0	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
0	1	1	1	1	1	0
1	1	1	1	1	1	0

Assignments: $2^{3 \cdot 3} = 512$

Positive assignments: $3! = 6$

Choi et al. [2015]

Motivation

Existing approaches:

LearnPSDD (Liang et al. [2017]):

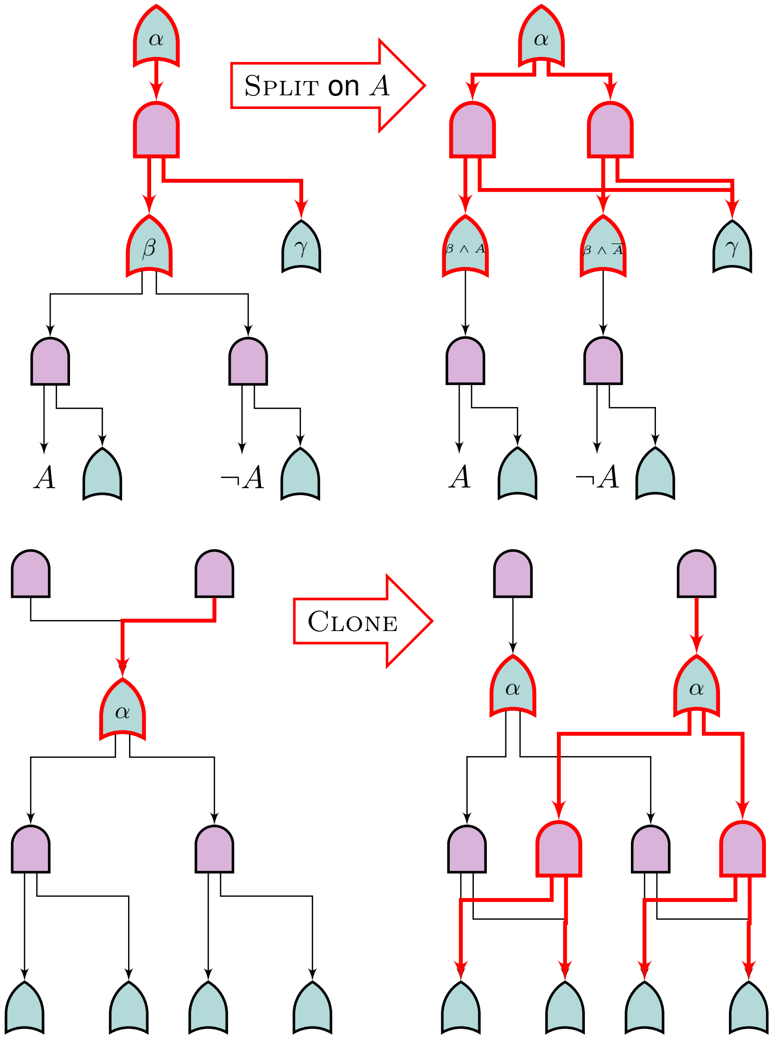
- ☒ Requires initial logic circuit encoding the support...
- ☒ Scales poorly to complex formulae and/or high dimension...
- ☒ Costly whole circuit evaluation at every iteration...
- ☑ Very good performance!

Strudel (Dang et al. [2020]):

- ☑ Constructs an initial structure (from a CLT)!
- ☒ But does not encode constraints...
- ☑ Scales to high dimension!
- ☒ As long as the circuit doesn't get too big...

SamplePSDD (Geh and Mauá [2021]):

- ☑ Scales to high dimension and complex formulae!
- ☑ Constructs a structure consistent with constraints!
- ☒ But does so by relaxing the formula...
- ☒ Performance varies on set bounds and vtree structure...

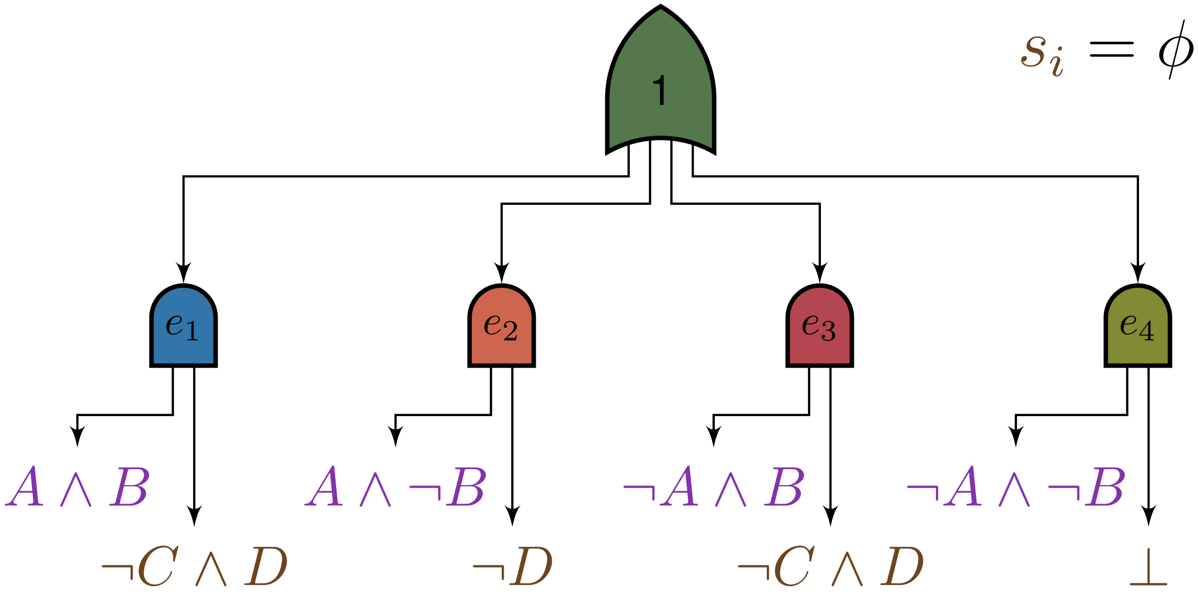


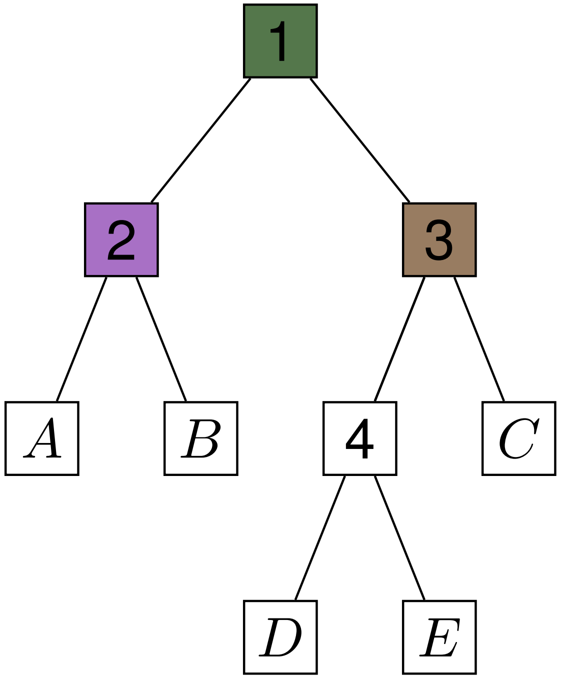
SamplePSDD

Common assumption: p_i are conjunctions of literals.

$$\phi(A, B, C, D) = (A \wedge \neg B \wedge \neg D) \vee (B \wedge \neg C \wedge D)$$

$$s_i = \phi|_{p_i}$$



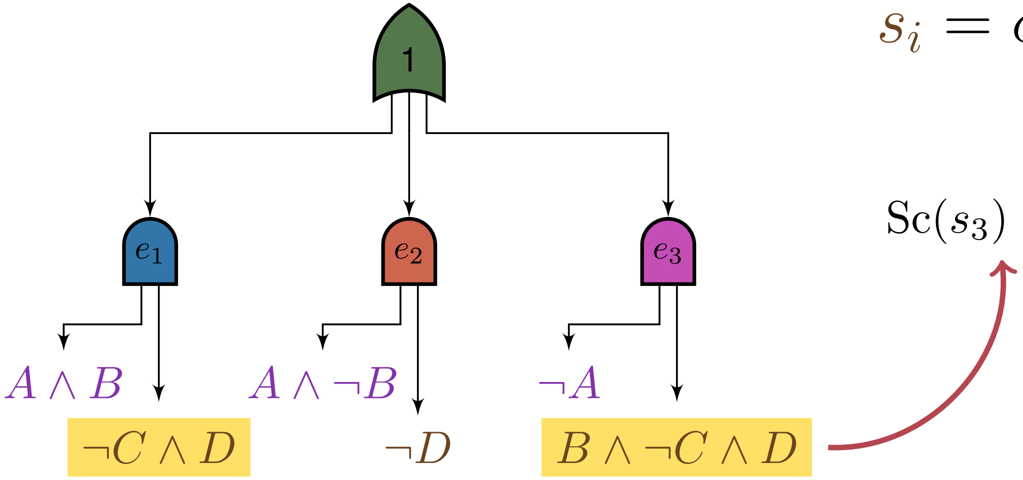


Problem: size of circuit is exponential in the size of p_i's scope.

SamplePSDD

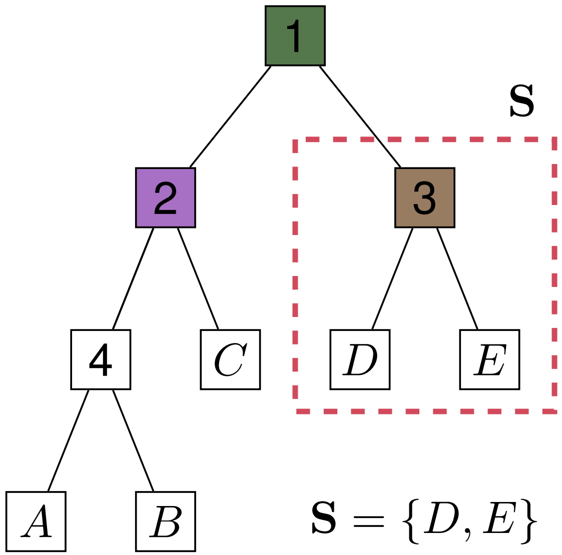
Solution: randomly sample a bounded number (k) of p_i

$$\phi(A,B,C,D) = (A \wedge \neg B \wedge \neg D) \vee (B \wedge \neg C \wedge D)$$



$$s_i = \phi|_{p_i}$$

$\mathrm{Sc}(s_3) \not\subseteq \mathbf{S}$



$\mathbf{S} = \{D, E\}$

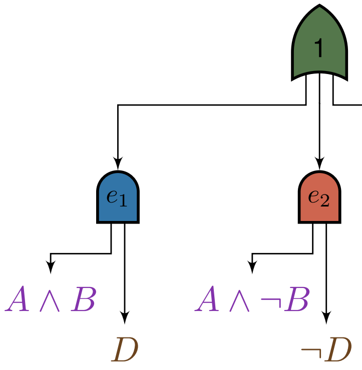
But: this violates structured decomposability:

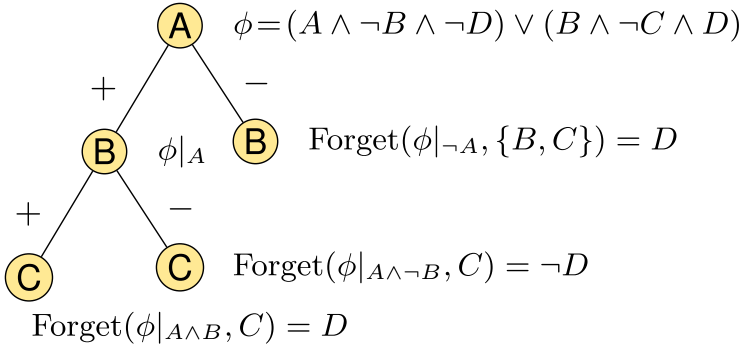
$\neg C \wedge D$ contains C, and $C \notin \mathbf{S}$

$\neg B \wedge \neg C \wedge D$ contains B and C, and $B, C \notin \mathbf{S}$

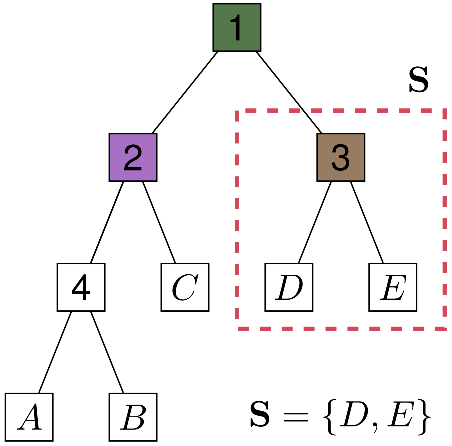
SamplePSDD

New solution: relax logical constraints ϕ



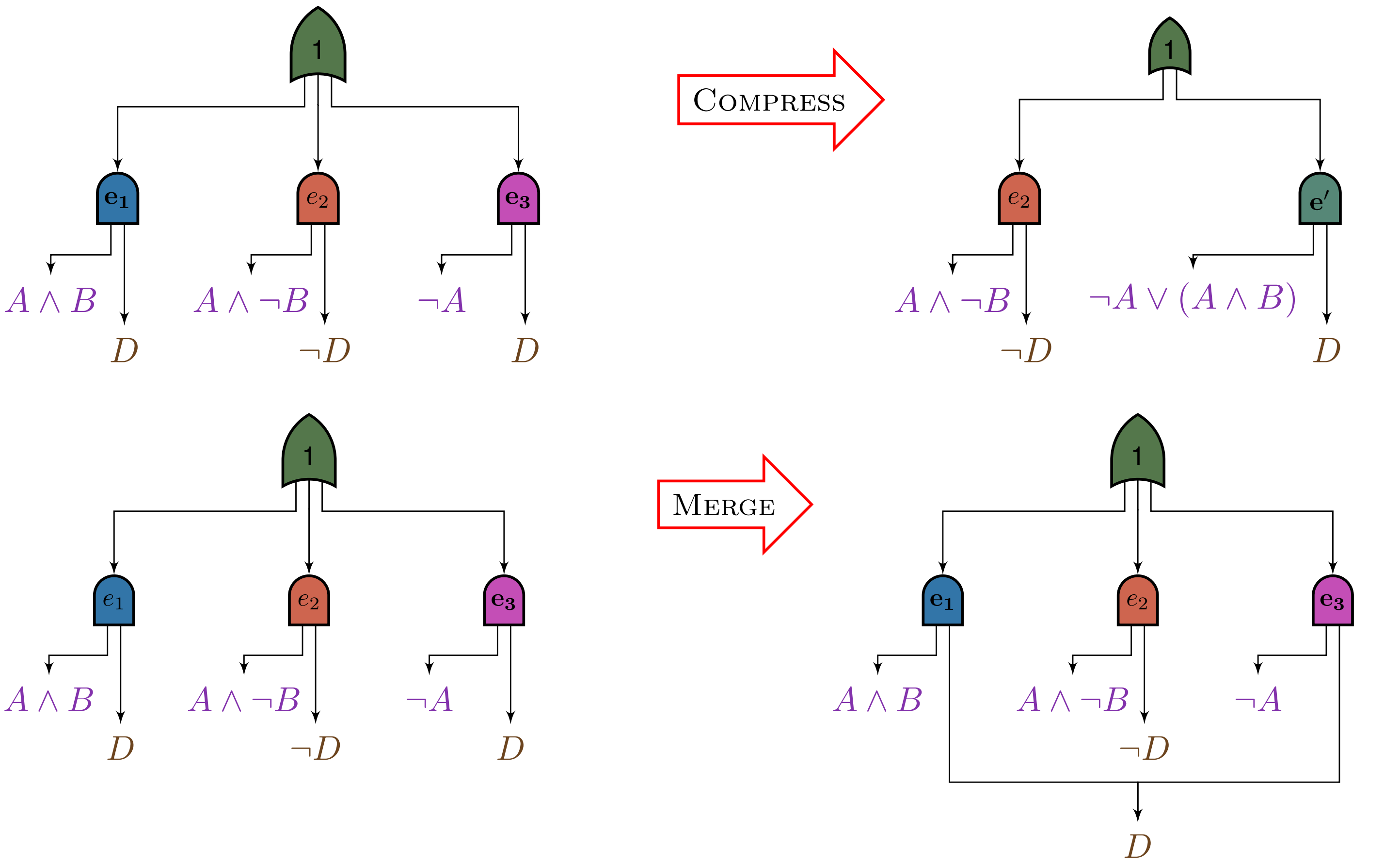


Now all s_i respect **S**



SamplePSDD

Apply **local transformations** for variety and size reduction



Darwiche [2011]

Experiments

Evaluation: we sample 30 PSDDs and use 5 ensemble strategies:

- ● **Likelihood weighting (LLW)**,
- ■ **Uniform weights**,
- ◆ **Expectation-Maximization (EM)**,
- ▲ **Stacking**,
- ▼ **Bayesian Model Combination (BMC)**;

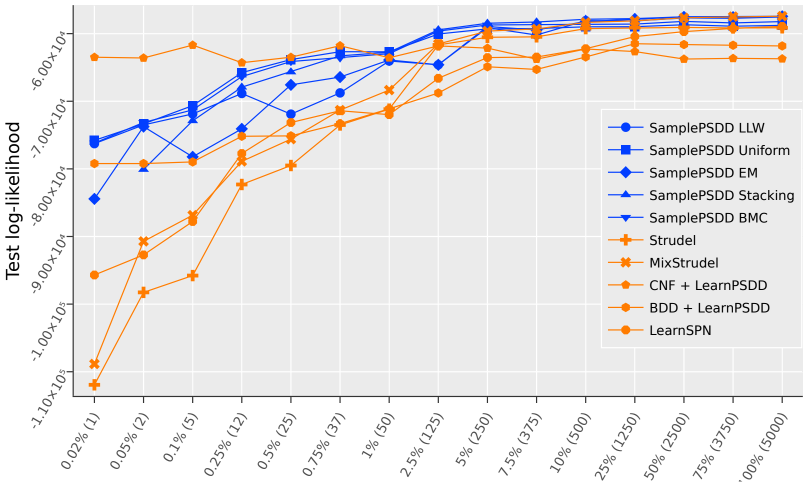
comparing against STRUDEL, LEARNPSDD and LEARNSPN.

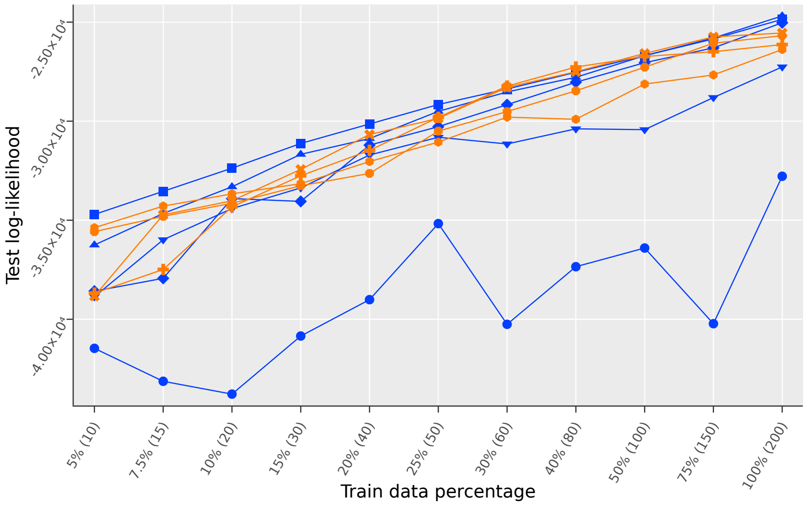
Datasets: we evaluate with 5 data + knowledge as logic constraints:

	Dataset	#vars	#train	ϕ's size
⇒	LED	14	5000	23
⇒	LED + IMAGES	157	700	39899
	SUSHI RANKING	100	3500	17413
	SUSHI TOP 5	10	3500	37
	DOTA 2 GAMES	227	92650	1308

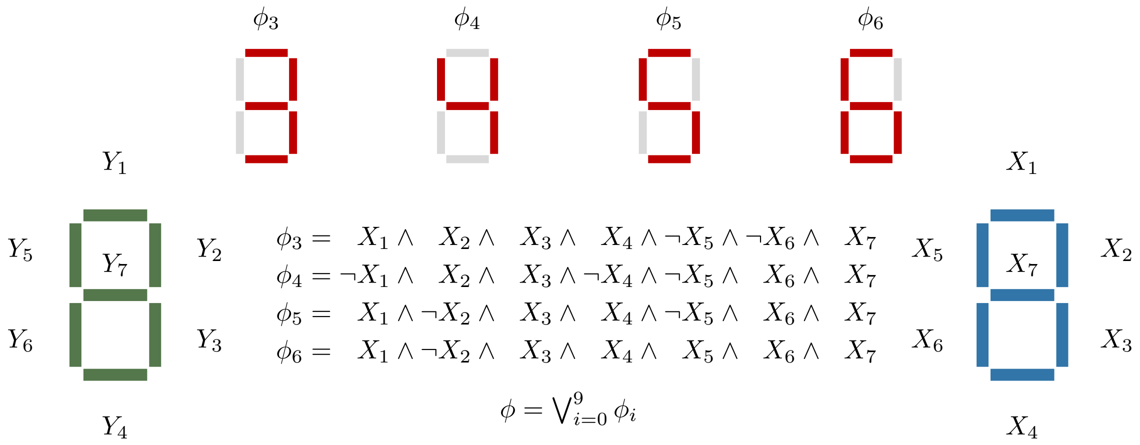
Our approach fares **better** with **fewer** data, yet remains **competitive** under **lots of data**.

Mattei et al. [2020], Kamishima [2003], Shen et al. [2017],
Choi et al. [2015], Gens and Domingos [2013], Dang et al. [2020]



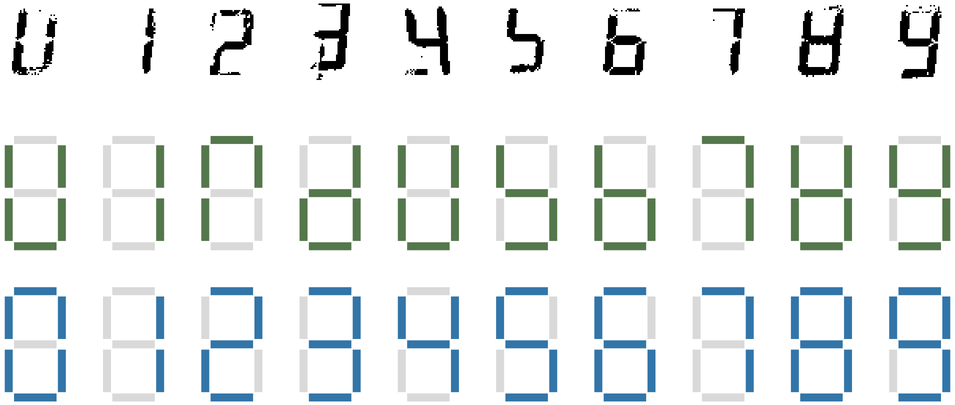


Experiments – LED



$$\phi_3 = X_1 \wedge X_2 \wedge X_3 \wedge X_4 \wedge \neg X_5 \wedge \neg X_6 \wedge X_7$$
$$\phi_4 = \neg X_1 \wedge X_2 \wedge X_3 \wedge \neg X_4 \wedge \neg X_5 \wedge X_6 \wedge X_7$$
$$\phi_5 = X_1 \wedge \neg X_2 \wedge X_3 \wedge X_4 \wedge \neg X_5 \wedge X_6 \wedge X_7$$
$$\phi_6 = X_1 \wedge \neg X_2 \wedge X_3 \wedge X_4 \wedge X_5 \wedge X_6 \wedge X_7$$

$$\phi = \bigvee_{i=0}^{9} \phi_i$$



Mattei et al. [2020]

Experiments

Evaluation: we sample 30 PSDDs and use 5 ensemble strategies:

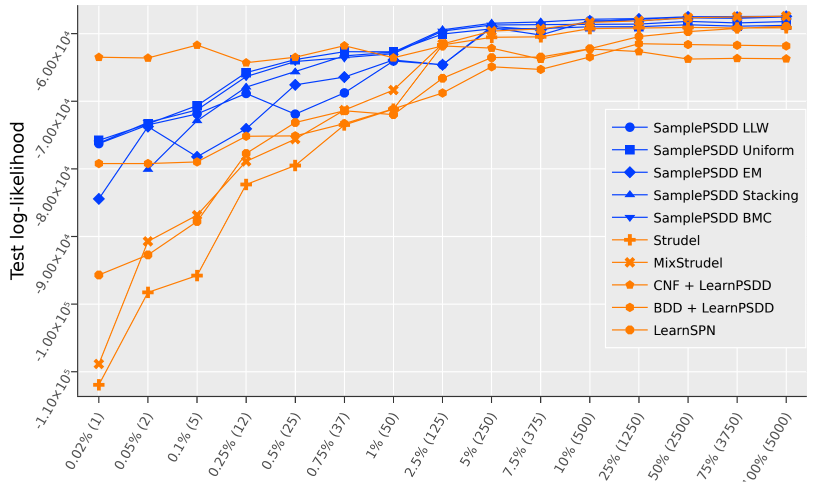
- ● **Likelihood weighting (LLW)**,
- ■ **Uniform weights**,
- ◆ **Expectation-Maximization (EM)**,
- ▲ **Stacking**,
- ▼ **Bayesian Model Combination (BMC)**;

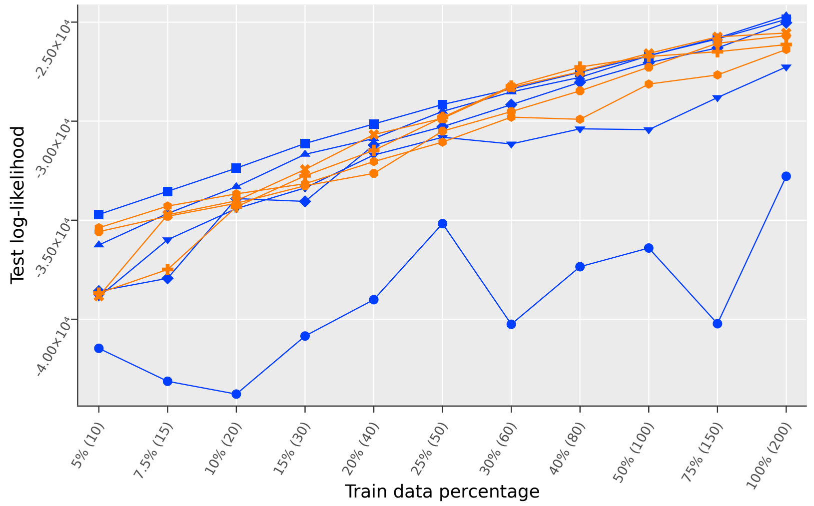
comparing against STRUDEL, LEARNPSDD and LEARNSPN.

Datasets: we evaluate with 5 data + knowledge as logic constraints:

	Dataset	#vars	#train	ϕ's size
⇒	LED	14	5000	23
⇒	LED + IMAGES	157	700	39899
	SUSHI RANKING	100	3500	17413
	SUSHI TOP 5	10	3500	37
	DOTA 2 GAMES	227	92650	1308

Our approach fares **better** with **fewer** data, yet remains **competitive** under **lots of data**.





Mattei et al. [2020], Kamishima [2003], Shen et al. [2017],
Choi et al. [2015], Gens and Domingos [2013], Dang et al. [2020]

Experiments

Evaluation: we sample 30 PSDDs and use 5 ensemble strategies:

- ● **Likelihood weighting (LLW)**,
- ■ **Uniform weights**,
- ◆ **Expectation-Maximization (EM)**,
- ▲ **Stacking**,
- ▼ **Bayesian Model Combination (BMC)**;

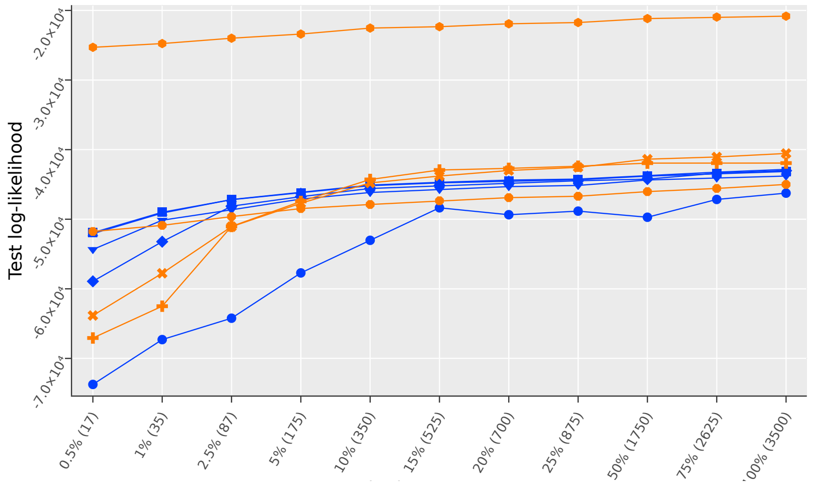
comparing against **STRUDEL**, **LEARNPSDD** and **LEARNSPN**.

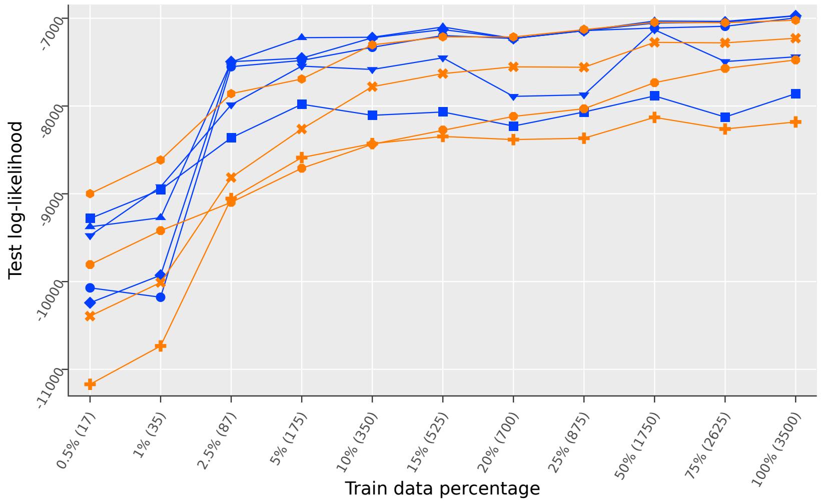
Datasets: we evaluate with 5 data + knowledge as logic constraints:

Dataset	#vars	#train	ϕ's size
LED	14	5000	23
LED + IMAGES	157	700	39899
⇒ SUSHI RANKING	100	3500	17413
⇒ SUSHI TOP 5	10	3500	37
DOTA 2 GAMES	227	92650	1308

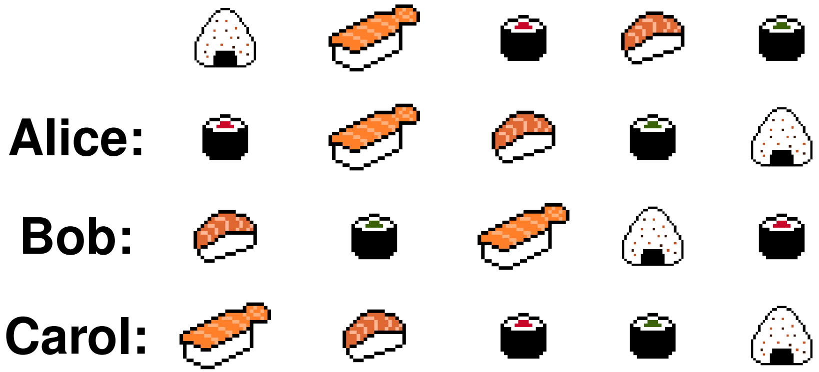
Our approach fares **better** with **fewer** data, yet remains **competitive** under **lots of data**.

Mattei et al. [2020], Kamishima [2003], Shen et al. [2017], Choi et al. [2015], Gens and Domingos [2013], Dang et al. [2020]





Experiments – SUSHI RANKING



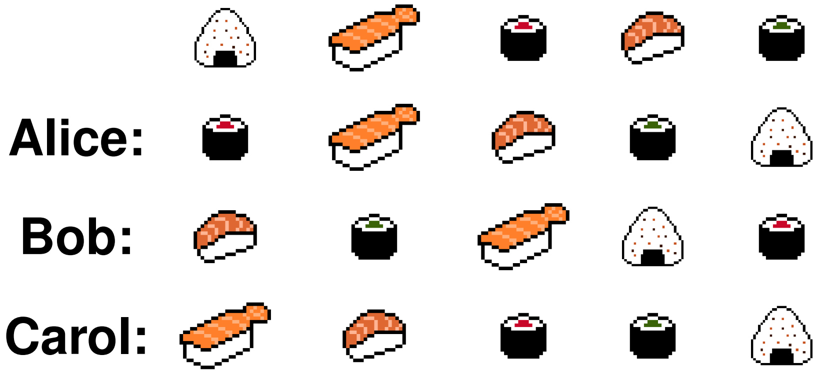
n sushi types and k rank positions

$$\alpha = \underbrace{\begin{aligned} &(\ X_{i1} \wedge \neg X_{i2} \wedge \cdots \wedge \neg X_{ik}) \\ \vee&(\neg X_{i1} \wedge \ X_{i2} \wedge \cdots \wedge \neg X_{ik}) \\ &\vdots \\ \vee&(\neg X_{i1} \wedge \neg X_{i2} \wedge \cdots \wedge \ X_{ik}) \end{aligned}}_{\text{Rank position}} \qquad \beta = \underbrace{\begin{aligned} &(\ X_{1j} \wedge \neg X_{2j} \wedge \cdots \wedge \neg X_{nj}) \\ \vee&(\neg X_{1j} \wedge \ X_{2j} \wedge \cdots \wedge \neg X_{nj}) \\ &\vdots \\ \vee&(\neg X_{1j} \wedge \neg X_{2j} \wedge \cdots \wedge \ X_{nj}) \end{aligned}}_{\text{Type uniqueness}}$$

$$\phi = \alpha \wedge \beta$$

Kamishima [2003], Shen et al. [2017]

Experiments – Sushi Top 5



n sushi types and k rank positions

Top k out of n sushi $\equiv$ n-choose-k model
n-choose-k model $\equiv$ cardinality $\text{Exactly}(k, n)$

$$\phi = \text{Exactly}(k, n) = \left(\sum_X X = k\right)$$

Kamishima [2003], Choi et al. [2015]

Experiments

Evaluation: we sample 30 PSDDs and use 5 ensemble strategies:

- ● **Likelihood weighting (LLW)**,
- ■ **Uniform weights**,
- ◆ **Expectation-Maximization (EM)**,
- ▲ **Stacking**,
- ▼ **Bayesian Model Combination (BMC)**;

comparing against **Strudel**, **LearnPSDD** and **LearnSPN**.

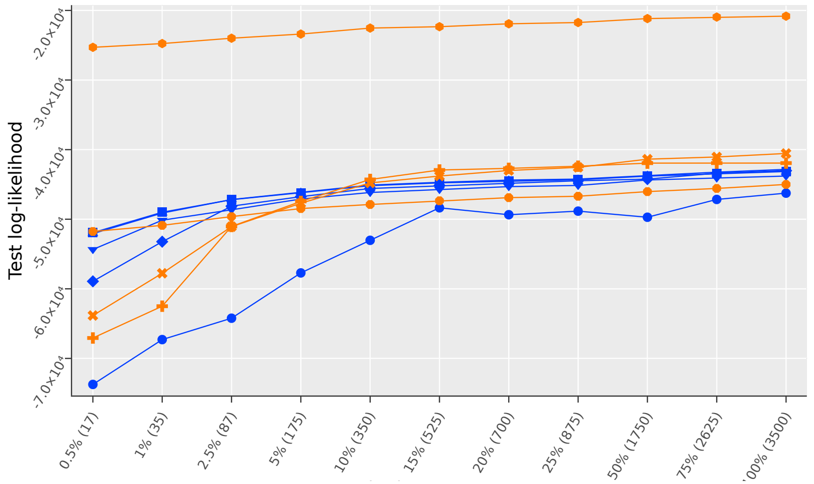
Datasets: we evaluate with 5 data + knowledge as logic constraints:

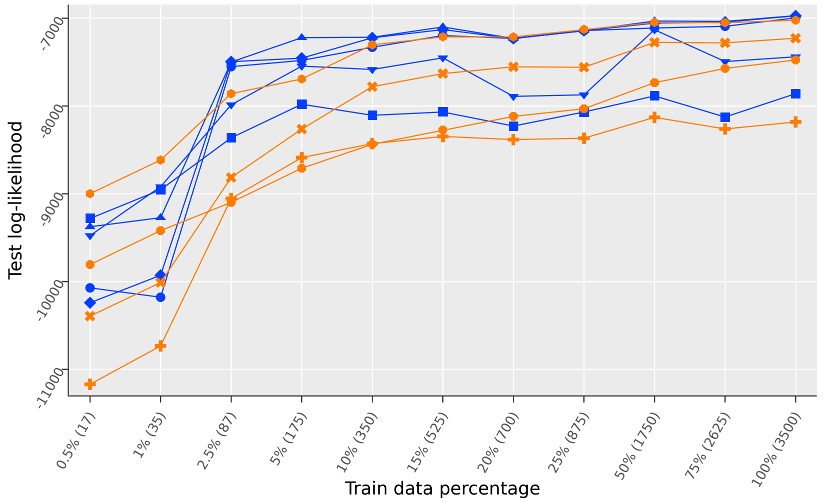
	Dataset	#vars	#train	ϕ's size
	LED	14	5000	23
	LED + Images	157	700	39899
⇒	Sushi Ranking	100	3500	17413
⇒	Sushi Top 5	10	3500	37
	Dota 2 Games	227	92650	1308

Our approach fares **better** with **fewer** data, yet

remains **competitive** under **lots of data**.

Mattei et al. [2020], Kamishima [2003], Shen et al. [2017],
Choi et al. [2015], Gens and Domingos [2013], Dang et al. [2020]





Experiments

Evaluation: we sample 30 PSDDs and use 5 ensemble strategies:

- ● **Likelihood weighting (LLW)**,
- ■ **Uniform weights**,
- ◆ **Expectation-Maximization (EM)**,
- ▲ **Stacking**,
- ▼ **Bayesian Model Combination (BMC)**;

comparing against **Strudel**, **LearnPSDD** and **LearnSPN**.

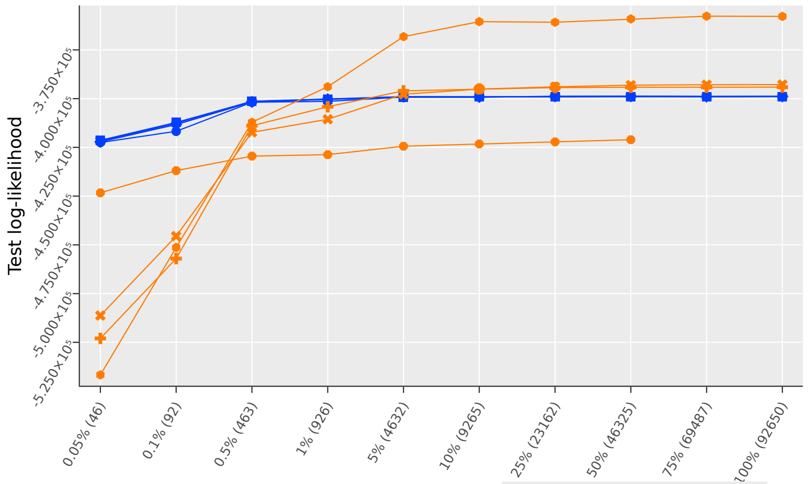
Datasets: we evaluate with 5 data + knowledge as logic constraints:

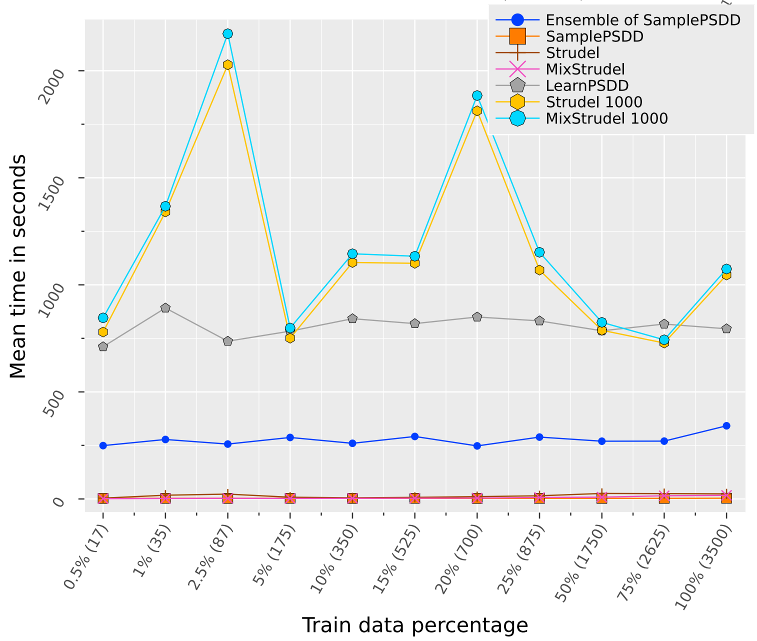
	Dataset	#vars	#train	ϕ's size
	LED	14	5000	23
	LED + Images	157	700	39899
	Sushi Ranking	100	3500	17413
	Sushi Top 5	10	3500	37
$\Rightarrow$	Dota 2 Games	227	92650	1308

Our approach fares **better** with **fewer** data, yet

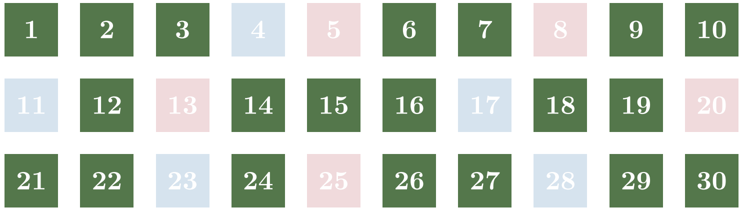
remains **competitive** under **lots of data**.

Mattei et al. [2020], Kamishima [2003], Shen et al. [2017],
Choi et al. [2015], Gens and Domingos [2013], Dang et al. [2020]





Experiments – DOTA 2 GAMES



X_1	X_2	X_3	X_4	X_5
4	11	17	23	28

n characters, k for each team

Y_5	Y_4	Y_3	Y_2	Y_1
25	20	13	8	5

$$\underbrace{\alpha = \text{Exactly}(k, n)}_{\text{Intractable as CNF}}$$

$$\underbrace{\beta = \text{Exactly}(k, n)}_{\text{Intractable as CNF}}$$

$$\underbrace{\gamma = X_i \neq Y_j, \forall X_i, Y_j}_{\text{Intractable as BDD}}$$

$$\phi = \alpha \wedge \beta \wedge \gamma$$

Tridgell [2016], Sinz [2005], Nishino et al. [2016], Eén and Sörensson [2006]

Experiments

Evaluation: we sample 30 PSDDs and use 5 ensemble strategies:

- ● **Likelihood weighting (LLW)**,
- ■ **Uniform weights**,
- ◆ **Expectation-Maximization (EM)**,
- ▲ **Stacking**,
- ▼ **Bayesian Model Combination (BMC)**;

comparing against **Strudel**, **LearnPSDD** and **LearnSPN**.

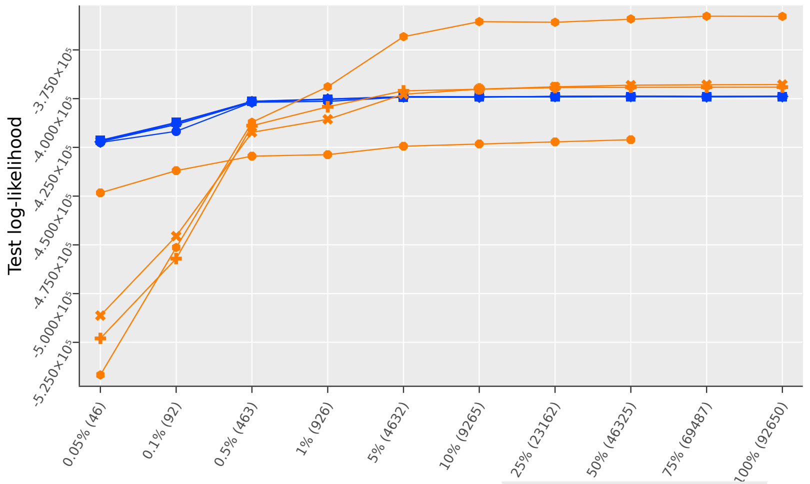
Datasets: we evaluate with 5 data + knowledge as logic constraints:

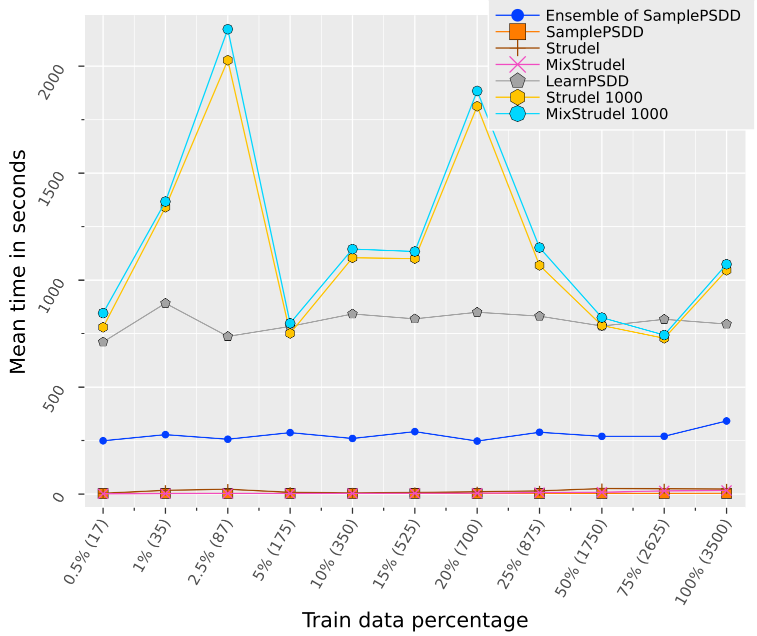
Dataset	#vars	#train	ϕ's size
LED	14	5000	23
LED + Images	157	700	39899
Sushi Ranking	100	3500	17413
Sushi Top 5	10	3500	37
$\Rightarrow$ Dota 2 Games	227	92650	1308

Our approach fares **better** with **fewer** data, yet

remains **competitive** under **lots of data**.

Mattei et al. [2020], Kamishima [2003], Shen et al. [2017], Choi et al. [2015], Gens and Domingos [2013], Dang et al. [2020]

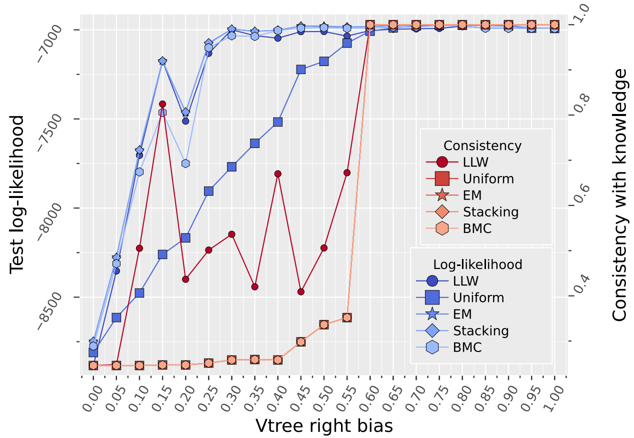


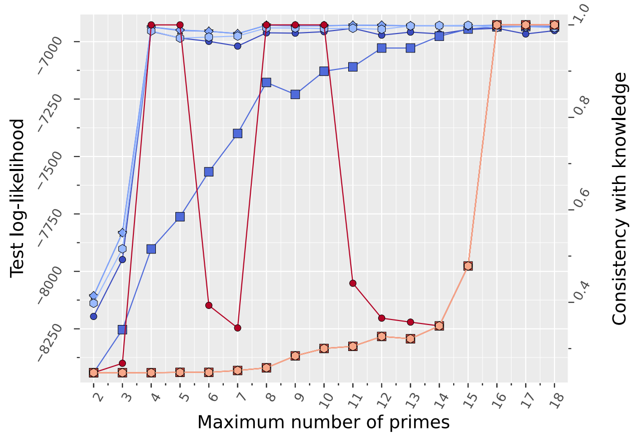


SamplePSDD – Experiments

What is the impact of higher k's and right-leaning vtrees

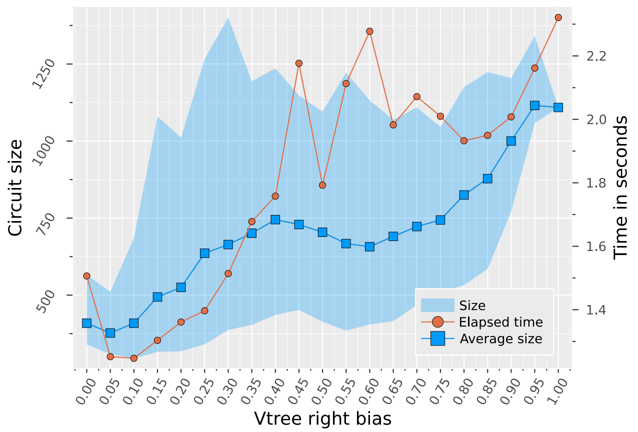
in log-likelihood and consistency?

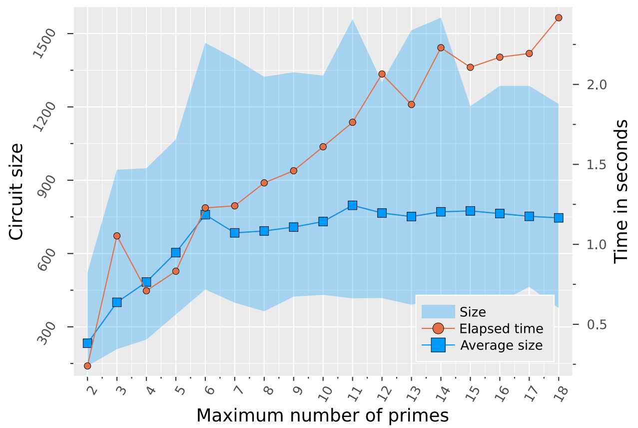




Samples perform **better** with higher k's and right-leaning vtrees...

...but at a **cost** to complexity.





Learning Probabilistic Circuits – Where are we right now?

Name	Class	Time Complexity	# hyperparams	Accepts logic?	Sm?	Dec?	Det?	Str Dec?	$\{0,1\}$?	$\mathbb{N}$?	$\mathbb{R}$?	Reference
LearnSPN	DIV	$\begin{cases} \mathcal{O}(nkmc) & \text{, if sum} \\ \mathcal{O}(nm^3) & \text{, if product} \end{cases}$	≥ 2	✗	✓	✓	✗	✗	✓	✓	✓	Gens and Domingos [2013]
ID-SPN	DIV	$\begin{cases} \mathcal{O}(nkmc) & \text{, if sum} \\ \mathcal{O}(nm^3) & \text{, if product} \\ \mathcal{O}(ic(rn+m)) & \text{, if input} \end{cases}$	$\geq 2+3$	✗	✓	✓	✗	✗	✓	✓	✗	Rooshenas and Lowd [2014]
Prometheus	DIV	$\begin{cases} \mathcal{O}(nkmc) & \text{, if sum} \\ \mathcal{O}(m(\log m)^2) & \text{, if product} \end{cases}$	≥ 1	✗	✓	✓	✗	✗	✓	✓	✓	Jaini et al. [2018a]
LearnPSDD	INCR	$\begin{cases} \mathcal{O}(m^2) & \text{, top-down vtree} \\ \mathcal{O}(m^4) & \text{, bottom-up vtree} \\ \mathcal{O}(i\lvert\mathcal{C}\rvert^2) & \text{, circuit structure} \end{cases}$	1	✓	✓	✓	✓	✓	✓	✗	✗	Liang et al. [2017]
Strudel	INCR	$\begin{cases} \mathcal{O}(m^2 n) & \text{, CLT + vtree} \\ \mathcal{O}(i(\lvert\mathcal{C}\rvert n + m^2)) & \text{, circuit structure} \end{cases}$	1	✓	✓	✓	✓	✓	✓	✗	✗	Dang et al. [2020]
RAT-SPN	RAND	$\mathcal{O}(rd(s+l))$	4	✗	✓	✓	✗	✗	✓	✓	✓	Peharz et al. [2020]
XPC	RAND	$\mathcal{O}(i(t+kn)+ikm^2 n)$	3	✗	✓	✓	✓	✓	✓	✗	✗	Mauro et al. [2021]
SamplePSDD	RAND	$\begin{cases} \mathcal{O}(m) & \text{, random vtree} \\ \mathcal{O}(kc\log c + \log_2^2 k) & \text{, per call} \end{cases}$	1	✓	✓	✓	✓	✓	✓	✗	✗	Geh and Mauá [2021]
⇒ LearnRP	RAND	$\begin{cases} \mathcal{O}(m^2) & \text{, top-down vtree} \\ \mathcal{O}(m^4) & \text{, bottom-up vtree} \\ \mathcal{O}(knm) & \text{, per call} \end{cases}$	0	✗	✓	✓	✗	✓	✓	✓	✓	To appear

A Data Perspective

Motivation

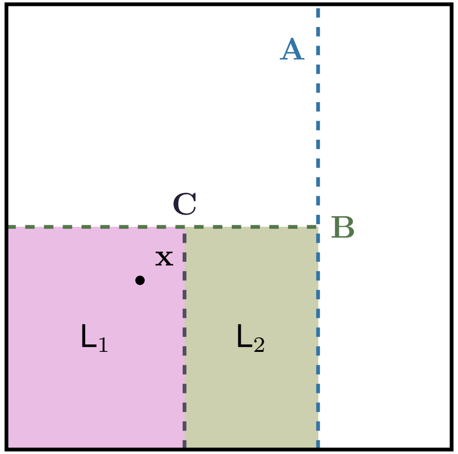
Density Estimation Trees...

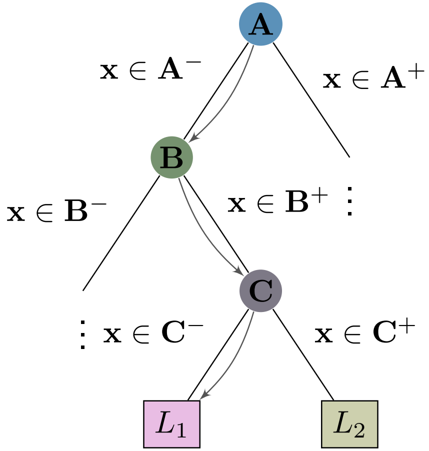
- ☑ ...are fast;
- ☑ ...are interpretable;
- ☑ ...are (somewhat) explainable;
- ☑ ...have extensive literature coverage;
- ☒ ...are not so expressive;
- ☒ ...only accept marginalization queries;
- ☒ ...are not so accurate;

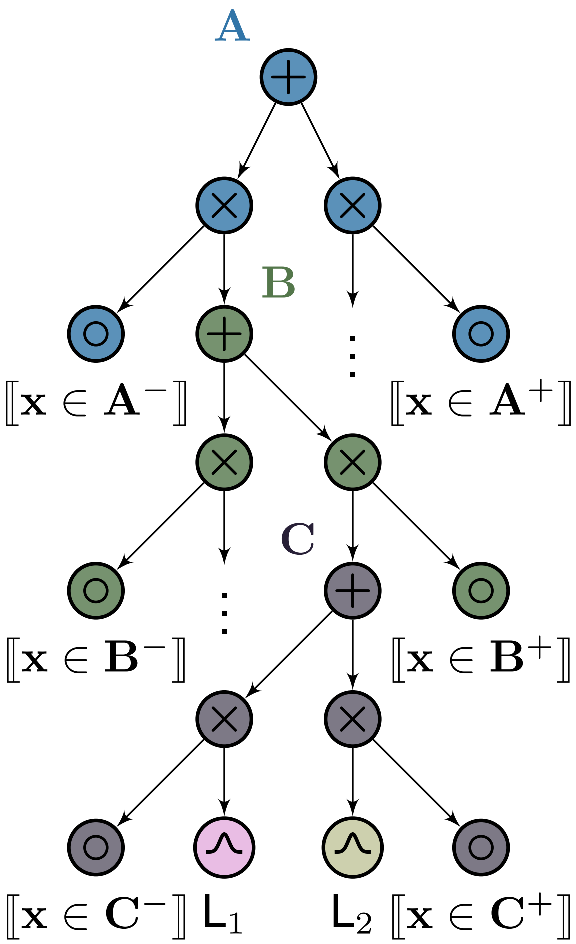
...but are subsumed by circuits!

Learn DETs $\subseteq$ Learn PCs?

Can we take advantage of known learning procedures in DETs and transplant them to more general circuits?

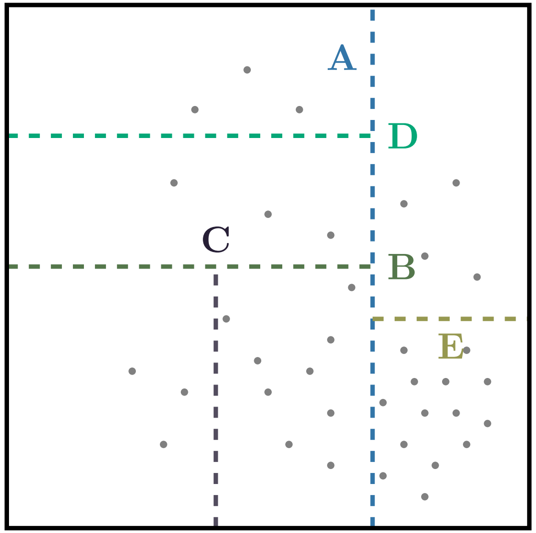




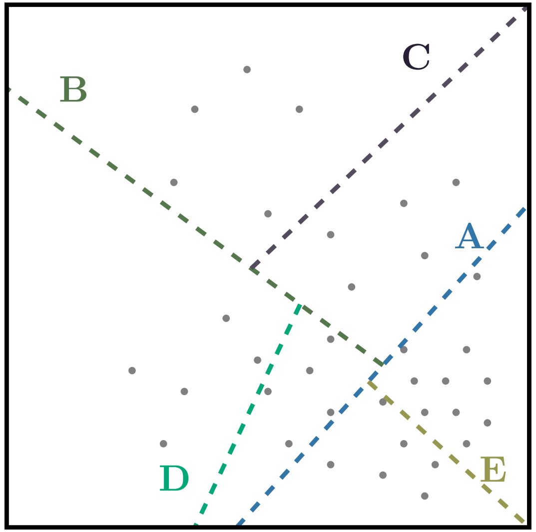


Correia et al. [2020], Ram and Gray [2011]

Random Projections



Axis-aligned projections

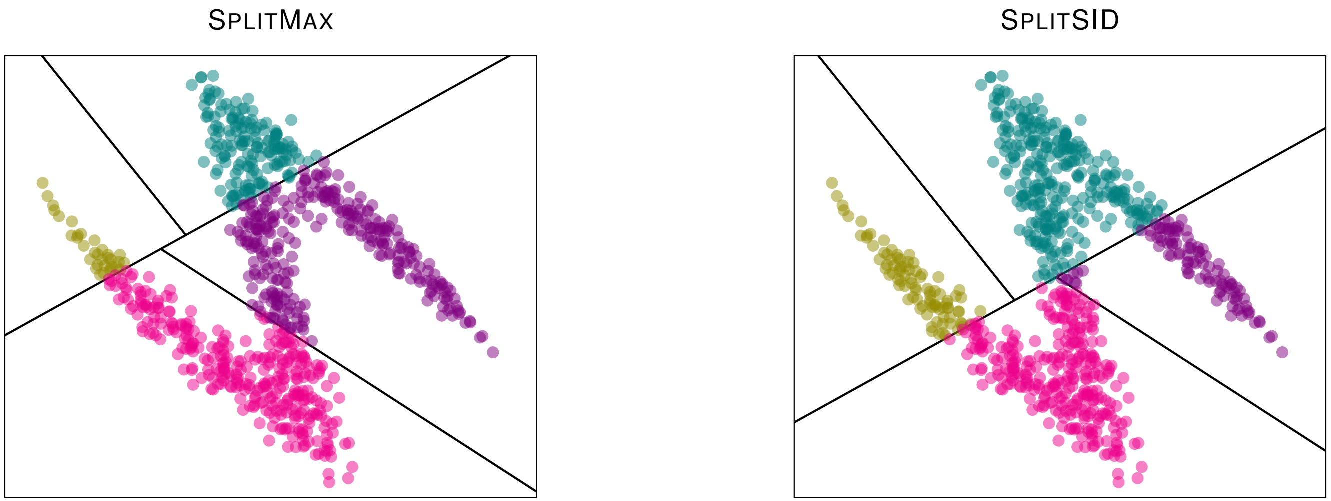


Random projections

If the data has *intrinsic dimension* d, then with constant probability the part of the data at level d or higher of the tree has average diameter less than half of the data.

Freund et al. [2008], Dasgupta and Freund [2008]

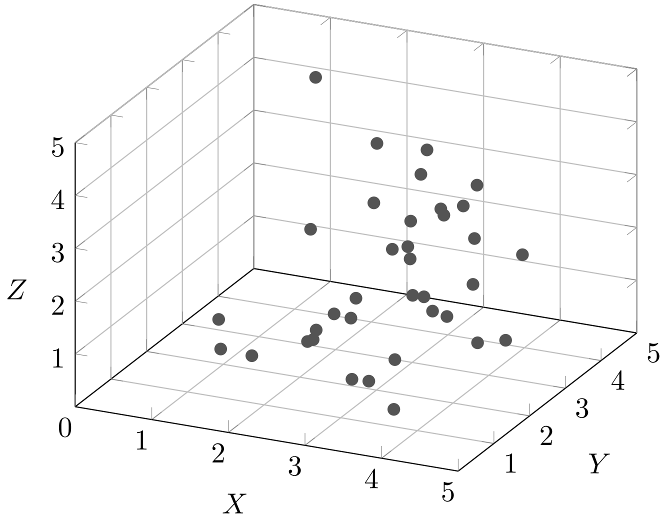
Random Projections

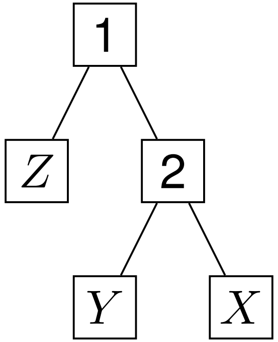


If the data has *intrinsic dimension* d, then with constant probability the part of the data at level d or higher of the tree has average diameter less than half of the data.

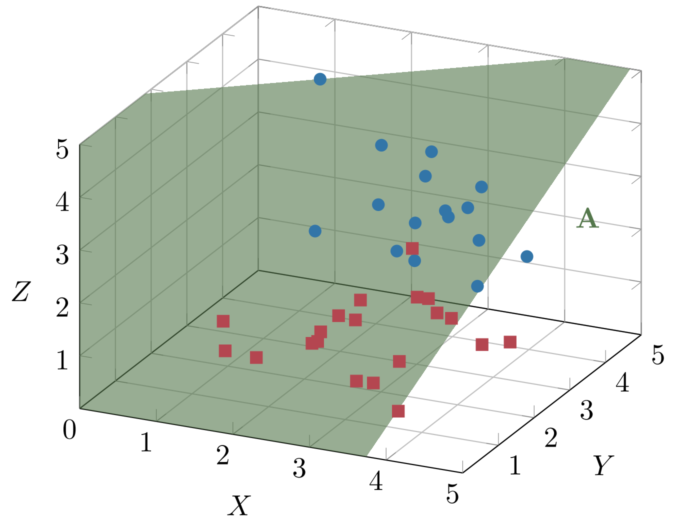
Freund et al. [2008], Dasgupta and Freund [2008]

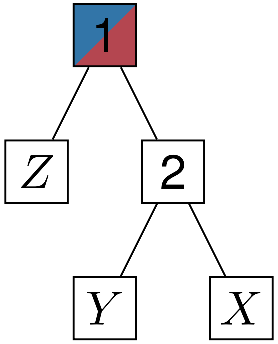
LearnRP



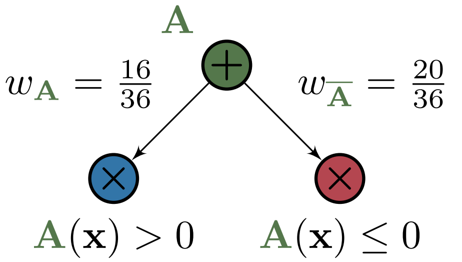


LearnRP



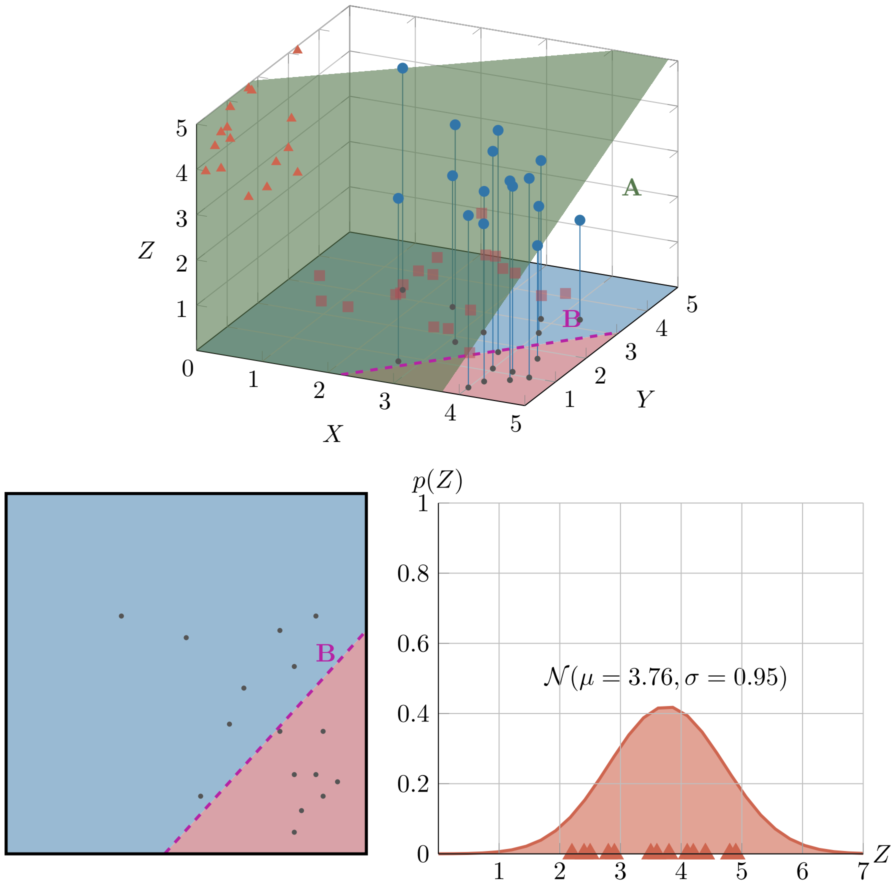


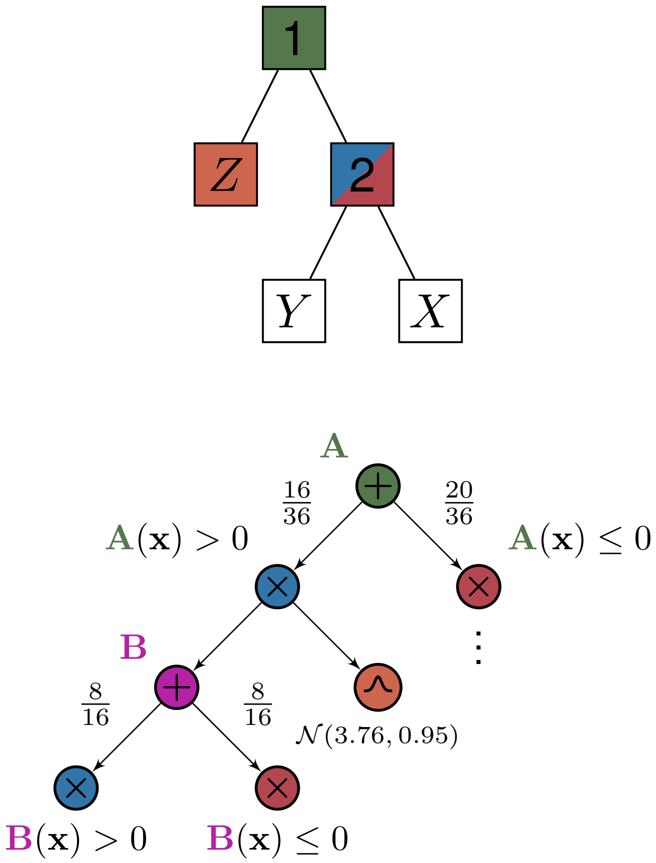
$$\mathbf{A}(x, y, z) = \begin{bmatrix} x & y & z \end{bmatrix} \cdot \underbrace{\begin{bmatrix} -0.31 \\ -0.40 \\ 0.85 \end{bmatrix}}_{a} + \underbrace{1}_{\theta}$$



$w_{\mathbf{A}}$: probability of $\mathbf{A}(\mathbf{x}) > 0$

LearnRP





$$\mathbf{B}(x, y) = \begin{bmatrix} x & y \end{bmatrix} \cdot \underbrace{\begin{bmatrix} 1.10 \\ -1.00 \end{bmatrix}}_{b} - \underbrace{2.43}_{\gamma}$$

Parameter Optimization

Expectation-Maximization (EM)

- Full EM (dataset $\mathbf{D}$)

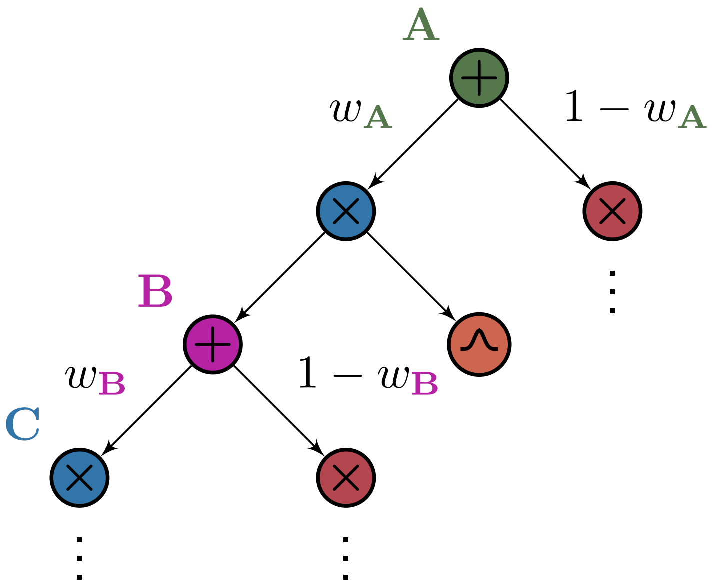
$$w_{\mathbf{B}} \propto w_{\mathbf{B}} \cdot \sum_{\mathbf{x} \in \mathbf{D}} \frac{1}{p_{\mathbf{A}}(\mathbf{x})} \cdot \frac{\partial p_{\mathbf{A}}(\mathbf{x})}{\partial p_{\mathbf{B}}(\mathbf{x})} \cdot p_{\mathbf{C}}(\mathbf{x})$$

- Minibatch EM (batch $\mathbf{M} \subset \mathbf{D}$)

$$w_{\mathbf{B}} \propto w_{\mathbf{B}} \cdot \sum_{\mathbf{x} \in \mathbf{M}} \frac{1}{p_{\mathbf{A}}(\mathbf{x})} \cdot \frac{\partial p_{\mathbf{A}}(\mathbf{x})}{\partial p_{\mathbf{B}}(\mathbf{x})} \cdot p_{\mathbf{C}}(\mathbf{x})$$

LEARNRP-100: LEARNRP + 100 itrs of minibatch

LEARNRP-F: LEARNRP-100 + 30 itrs of full



Dempster et al. [1977], Liu and Van den Broeck [2021]

LearnRP – Datasets

Dataset	Vars	Train	Test	Domain	Dataset	Vars	Train	Test	Domain
ACCIDENTS	111	12758	2551	$\{0,1\}$	NLTCS	16	16181	3236	$\{0,1\}$
AD	1556	2461	491	$\{0,1\}$	PLANTS	69	17412	3482	$\{0,1\}$
AUDIO	100	15000	3000	$\{0,1\}$	PUMSB-STAR	163	12262	2452	$\{0,1\}$
BBC	1058	1670	330	$\{0,1\}$	EACHMOVIE	500	4524	591	$\{0,1\}$
NETFLIX	100	15000	3000	$\{0,1\}$	RETAIL	135	22041	4408	$\{0,1\}$
BOOK	500	8700	1739	$\{0,1\}$	ABALONE	8	3760	417	$\mathbb{R}$
20-NEWSGRP	910	11293	3764	$\{0,1\}$	CA	22	7373	819	$\mathbb{R}$
REUTERS-52	889	6532	1540	$\{0,1\}$	QUAKE	4	1961	217	$\mathbb{R}$
WEBKB	839	2803	838	$\{0,1\}$	SENSORLESS	48	52659	5850	$\mathbb{R}$
DNA	180	1600	1186	$\{0,1\}$	BANKNOTE	4	1235	137	$\mathbb{R}$
JESTER	100	9000	4116	$\{0,1\}$	FLOWSIZE	3	1358674	150963	$\mathbb{R}$
KDD	65	180092	34955	$\{0,1\}$	KINEMATICS	8	7373	819	$\mathbb{R}$
KOSAREK	190	33375	6675	$\{0,1\}$	IRIS	4	90	10	$\mathbb{R}$
MSNBC	17	291326	58265	$\{0,1\}$	OLDFAITH	2	245	27	$\mathbb{R}$
MSWEB	294	29441	5000	$\{0,1\}$	CHEMDIABET	3	131	14	$\mathbb{R}$

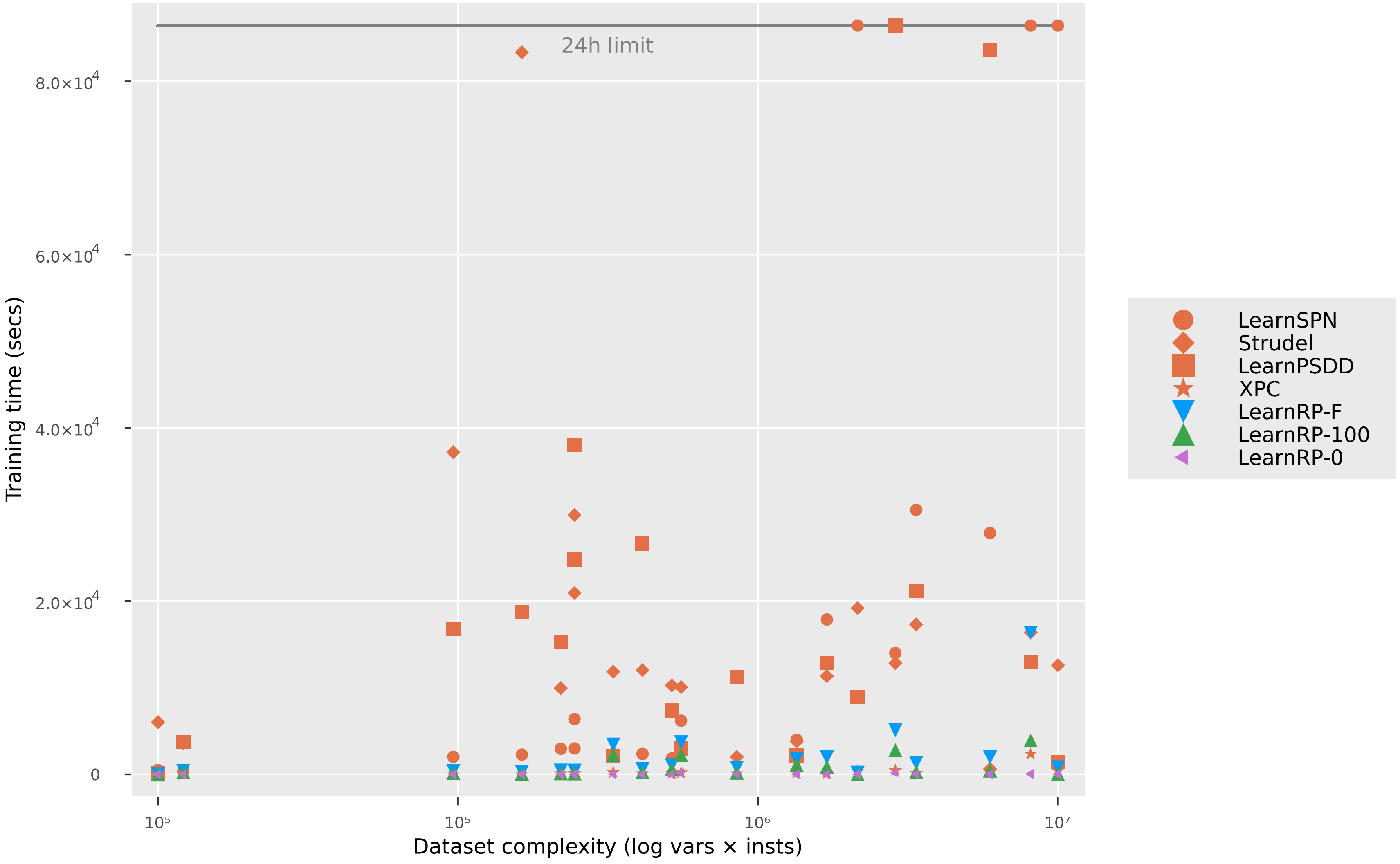
Lowd and Davis [2010], Van Haaren and Davis [2012], Dua and Graff [2017], Güvenir and Uysal [2000]

Experiments

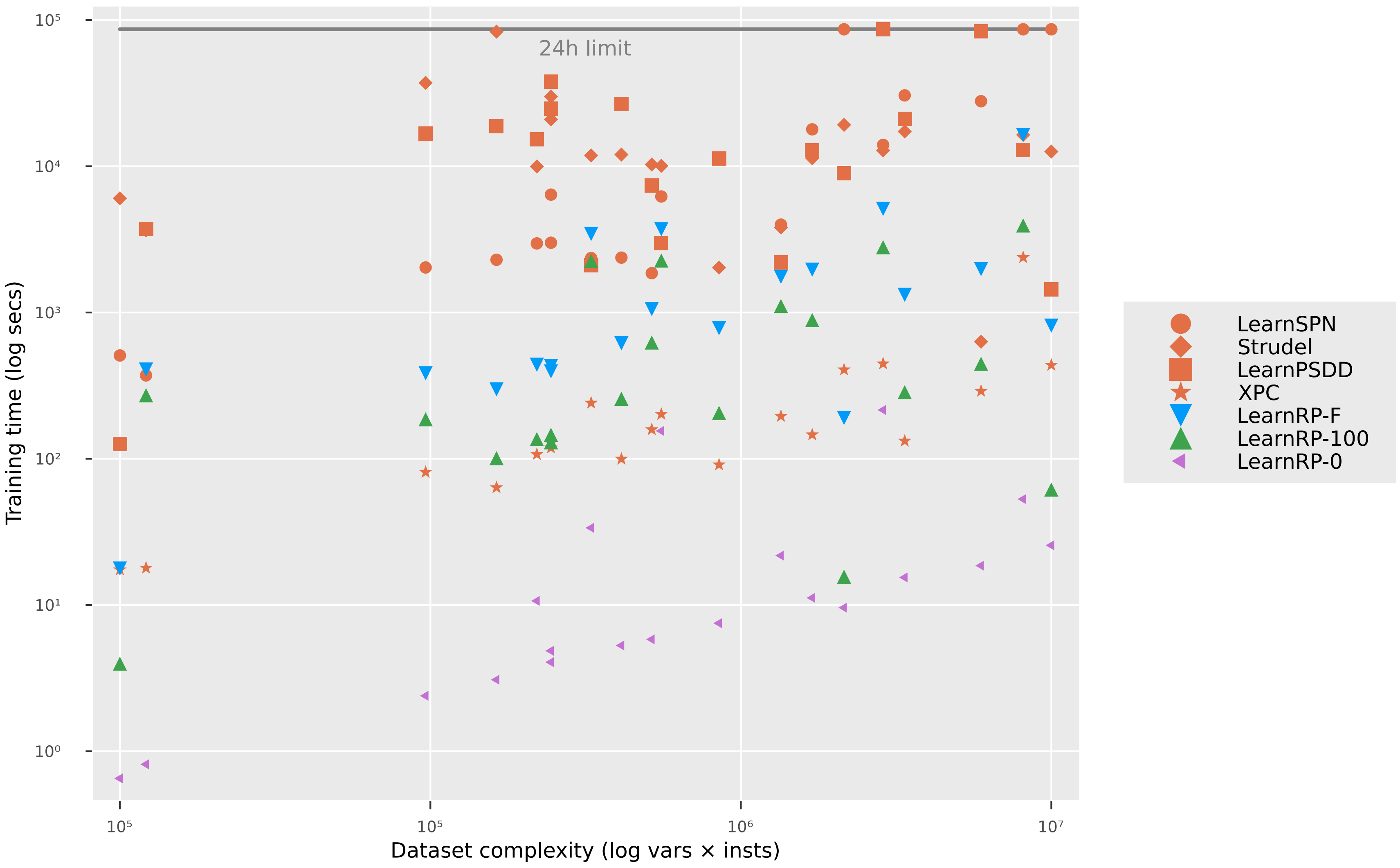
<table>
<tr><th>Dataset</th><th>LEARNSPN</th><th>STRUDEL</th><th>LEARNPSDD</th><th>XPC</th><th>PROMETHEUS</th><th>LEARNRP-F</th><th>LEARNRP-100</th></tr>
<tr><td>ACCIDENTS</td><td>-30.03</td><td>|-28.73|</td><td>-30.16</td><td>-31.02</td><td>-27.91</td><td><u>-28.66</u></td><td>-28.81</td></tr>
<tr><td>AD</td><td>-19.73</td><td><u>-16.38</u></td><td>-31.78</td><td>-15.50</td><td>-23.96</td><td>|-19.26|</td><td>-19.99</td></tr>
<tr><td>AUDIO</td><td>-40.50</td><td>-41.50</td><td><u>-39.94</u></td><td>-40.91</td><td>-39.80</td><td>|-40.27|</td><td>-40.30</td></tr>
<tr><td>BBC</td><td>|-250.68|</td><td>-254.41</td><td>-253.19</td><td>-248.34</td><td><u>-248.50</u></td><td>-254.15</td><td>-251.57</td></tr>
<tr><td>NETFLIX</td><td>|-57.02|</td><td>-58.69</td><td>-55.71</td><td>-57.58</td><td><u>-56.47</u></td><td>|-57.02|</td><td>-57.03</td></tr>
<tr><td>BOOK</td><td>-35.88</td><td>-34.99</td><td>-34.97</td><td>-34.75</td><td>|-34.40|</td><td><u>-33.56</u></td><td>-33.41</td></tr>
<tr><td>20-NEWSGRP</td><td>-155.92</td><td>-154.47</td><td>-155.97</td><td>|-153.75|</td><td>-154.17</td><td><u>-152.63</u></td><td>-152.34</td></tr>
<tr><td>REUTERS-52</td><td>|-85.06|</td><td>-86.22</td><td>-89.61</td><td><u>-84.70</u></td><td>-84.59</td><td>-85.69</td><td>-85.76</td></tr>
<tr><td>WEBKB</td><td>-158.20</td><td>-155.33</td><td>-161.09</td><td>|-153.67|</td><td>-155.21</td><td><u>-153.52</u></td><td>-151.80</td></tr>
<tr><td>DNA</td><td>-82.52</td><td>-86.22</td><td>-88.01</td><td>-86.61</td><td>-84.45</td><td><u>-83.57</u></td><td>|-83.62|</td></tr>
<tr><td>JESTER</td><td>-75.98</td><td>-55.03</td><td>-51.29</td><td>-53.43</td><td><u>-52.80</u></td><td>-52.92</td><td>|-52.86|</td></tr>
<tr><td>KDD</td><td>-2.18</td><td>|-2.13|</td><td>-2.11</td><td>-2.15</td><td><u>-2.12</u></td><td>-2.14</td><td>-2.14</td></tr>
<tr><td>KOSAREK</td><td>-10.98</td><td>-10.68</td><td>-10.52</td><td>-10.77</td><td><u>-10.59</u></td><td>|-10.62|</td><td>-10.66</td></tr>
<tr><td>MSNBC</td><td><u>-6.11</u></td><td>-6.04</td><td>-6.04</td><td>|-6.18|</td><td>-6.04</td><td>-6.33</td><td>-6.35</td></tr>
<tr><td>MSWEB</td><td>-10.25</td><td>-9.71</td><td>|-9.89|</td><td>-9.93</td><td><u>-9.86</u></td><td>-9.90</td><td>-9.93</td></tr>
<tr><td>NLTCS</td><td>-6.11</td><td>-6.06</td><td>-5.99</td><td>|-6.05|</td><td><u>-6.01</u></td><td>-6.22</td><td>-6.27</td></tr>
<tr><td>PLANTS</td><td><u>-12.97</u></td><td>|-12.98|</td><td>-13.02</td><td>-14.19</td><td>-12.81</td><td>-13.77</td><td>-13.81</td></tr>
<tr><td>PUMSB-STAR</td><td>|-24.78|</td><td><u>-24.12</u></td><td>-26.12</td><td>-26.06</td><td>-22.75</td><td>-26.12</td><td>-26.33</td></tr>
<tr><td>EACHMOVIE</td><td>-52.48</td><td>-53.67</td><td>-58.01</td><td>-54.82</td><td>|-51.49|</td><td><u>-51.41</u></td><td>-50.95</td></tr>
<tr><td>RETAIL</td><td>-11.04</td><td><u>-10.81</u></td><td>-10.72</td><td>-10.94</td><td>-10.87</td><td>|-10.84|</td><td>-10.86</td></tr>
<tr><td>Avg. Rank</td><td>4.83 ± 1.89</td><td>4.30± 1.92</td><td>|4.03± 2.57|</td><td>4.62 ± 1.88</td><td>2.50 ± 1.43</td><td><u>3.62 ± 1.47</u></td><td>4.10 ± 1.98</td></tr>
<tr><td>Pos. (mean)</td><td>7th</td><td>5th</td><td>|3rd|</td><td>6th</td><td>1st</td><td><u>2nd</u></td><td>4th</td></tr>
</table>

Gens and Domingos [2013], Dang et al. [2020], Liang et al. [2017], Mauro et al. [2021], Jaini et al. [2018a]

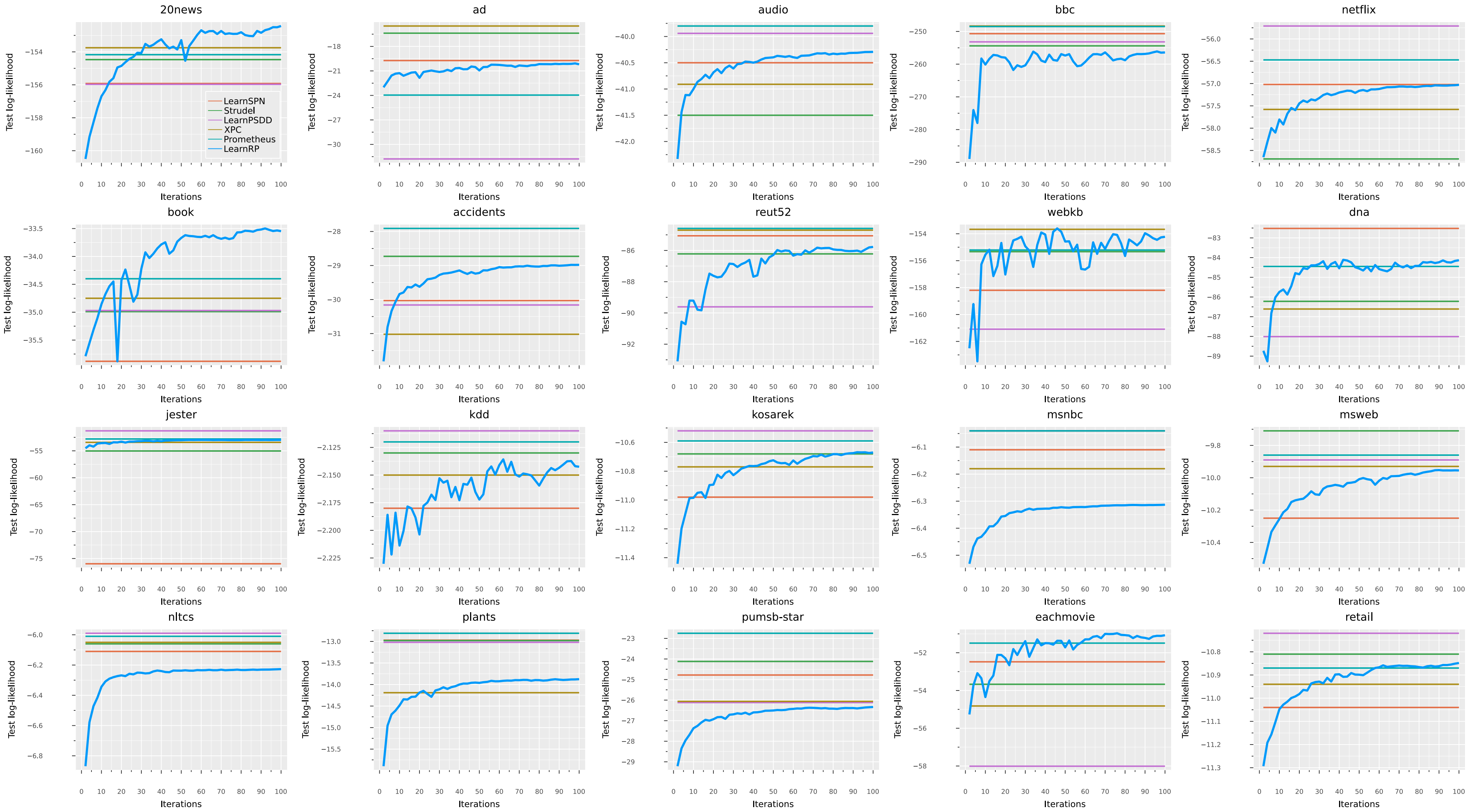
Experiments



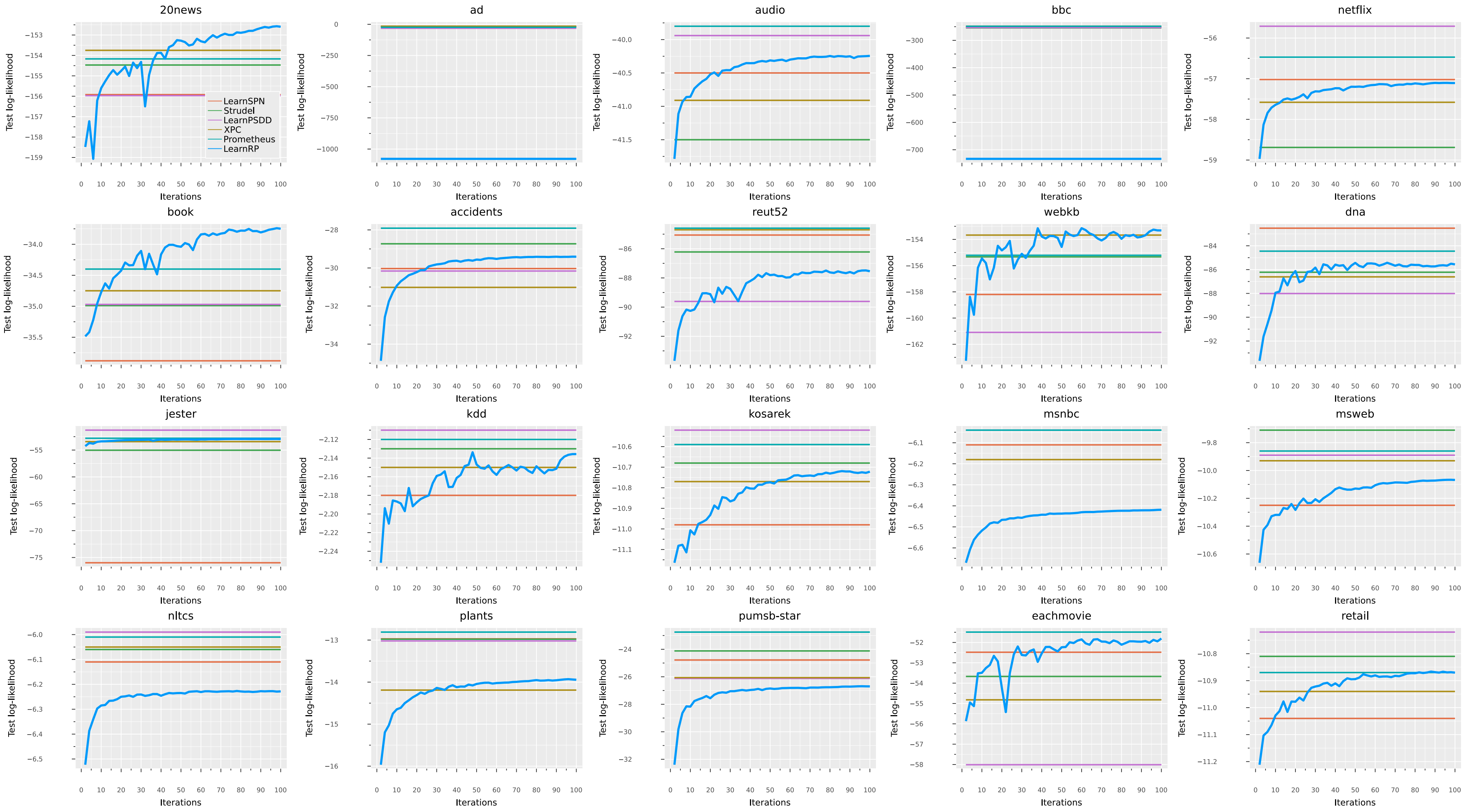
Experiments



LearnRP – Learning Curves



LEARNRP – Random Initializations



Experiments

Dataset	Vars	SRBMs	oSLRAU	GBMMs	iGMMs	GMMs	PROMETHEUS	iSPTs	LEARNRP	Size
ABALONE	8	-2.28	-0.94	|-1.17|	—	-4.65	**-0.85**	—	-3.58	317
BANKNOTE	4	|-2.76|	**-1.39**	-4.64	—	-4.32	-1.96	—	-4.27	79
CA	22	-4.95	21.19	3.42	—	-7.33	**27.82**	—	|9.48|	2675
KINEMATICS	8	**-5.55**	-11.13	-11.20	—	-11.15	|-11.12|	—	-10.16	319
QUAKE	4	-2.38	**-1.21**	-3.76	—	-4.09	-1.50	—	|-1.63|	79
SENSORLESS	48	-26.91	60.72	8.56	—	-34.14	**62.03**	—	|17.52|	12650
CHEMDIABET	3	—	—	—	|-3.02|	-18.49	**-2.59**	-2.88	-19.06	47
FLOWSIZE	3	-0.79	15.32	|5.72|	—	2.27	**18.03**	—	2.83	49
OLDFAITH	2	—	—	—	|-1.73|	-4.18	**-1.48**	-1.70	-4.26	19
IRIS	4	—	—	—	-3.94	-2.26	**-1.06**	-3.74	|-3.14|	79

Jaini et al. [2018a], Salakhutdinov and Hinton [2009], Rasmussen [2000], Jaini et al. [2018b], Hsu et al. [2017], Trapp et al. [2016], Dua and Graff [2017], Güvenir and Uysal [2000]

In conclusion

Contributions

Literature review

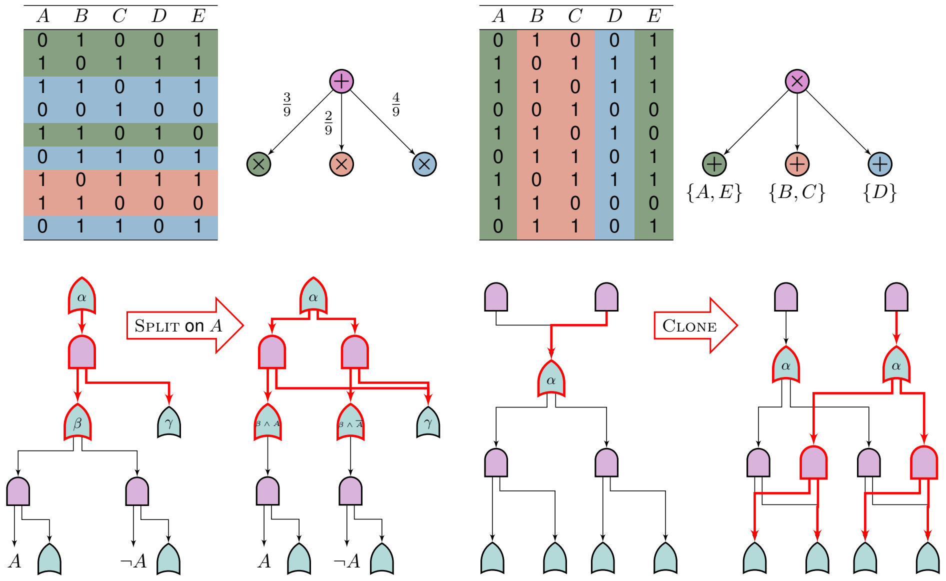
- Systematic review of literature;
- Taxonomy of popular algorithms;
- Complexity analysis;
- Pros and cons.

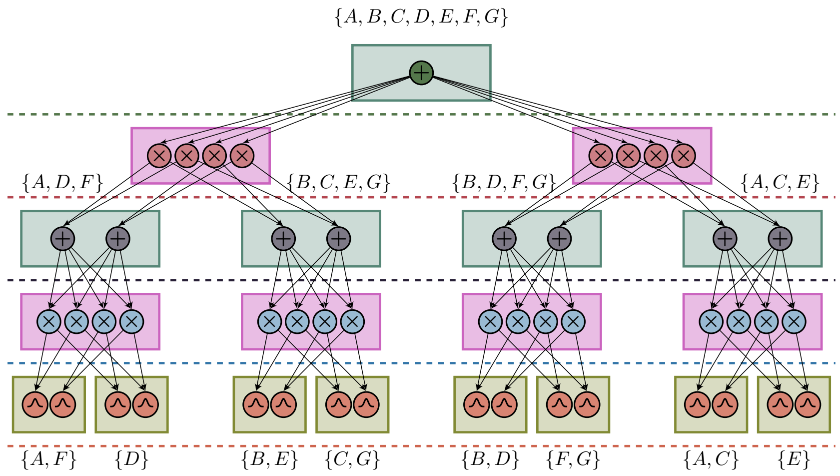
SamplePSDD

- Consistent with a relaxation of a formula;
- Relaxation as a function of vtree and sampling;
- Compromise between tractability and consistency;
- Ensembles mitigate relaxation.

LearnRP

- Simple strategy;
- Inspiration from known DET literature;
- Orders of magnitude faster;
- Competitive performance.





Contributions

Literature review

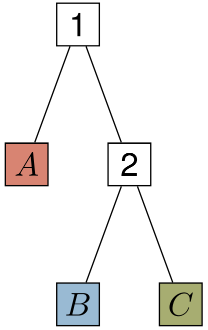
- Systematic review of literature;
- Taxonomy of popular algorithms;
- Complexity analysis;
- Pros and cons.

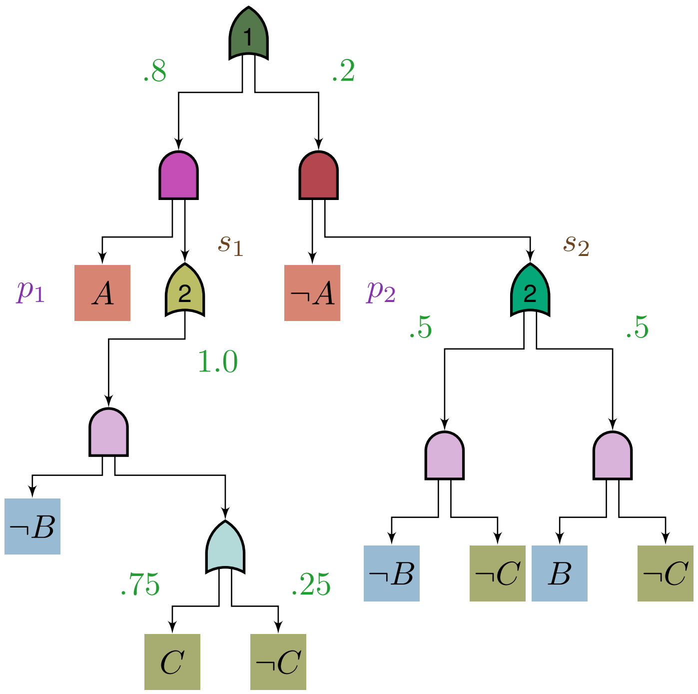
A	B	C	$p(\mathbf{x})$
0	0	0	0.1
0	1	0	0.1
1	0	0	0.2
1	0	1	0.6

SamplePSDD

- Consistent with a relaxation of a formula;
- Relaxation as a function of vtree and sampling;
- Compromise between tractability and consistency;
- Ensembles mitigate relaxation.

$$\phi(A, B, C) = (A \to \neg B) \land (C \to A)$$





LearnRP

- Simple strategy;
- Inspiration from known DET literature;
- Orders of magnitude faster;
- Competitive performance.

Contributions

Literature review

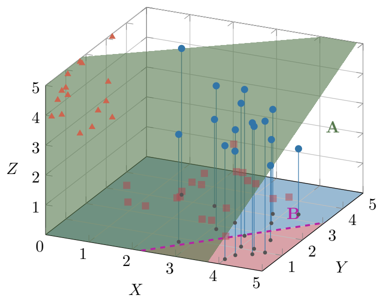
- Systematic review of literature;
- Taxonomy of popular algorithms;
- Complexity analysis;
- Pros and cons.

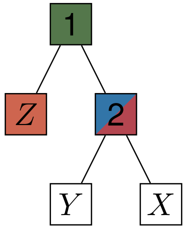
SamplePSDD

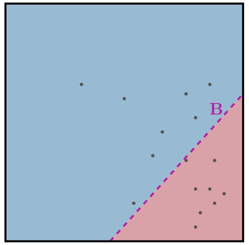
- Consistent with a relaxation of a formula;
- Relaxation as a function of vtree and sampling;
- Compromise between tractability and consistency;
- Ensembles mitigate relaxation.

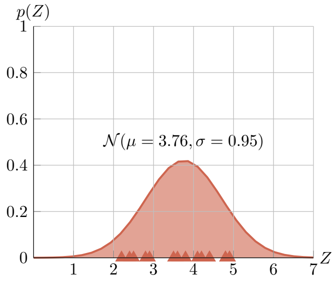
LearnRP

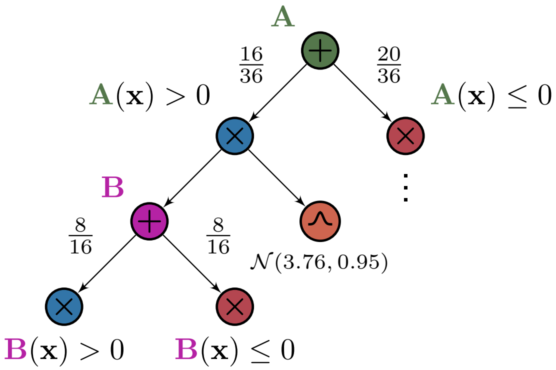
- Simple strategy;
- Inspiration from known DET literature;
- Orders of magnitude faster;
- Competitive performance.











$$\mathbf{B}(x, y) = \begin{bmatrix} x & y \end{bmatrix} \cdot \begin{bmatrix} 1.10 \\ -1.00 \end{bmatrix} - 2.43$$