Probabilistic and Logistic Circuits:

A New Synthesis of Logic and Machine Learning

Guy Van den Broeck

UCLA

HRL/ACTIONS @ KR

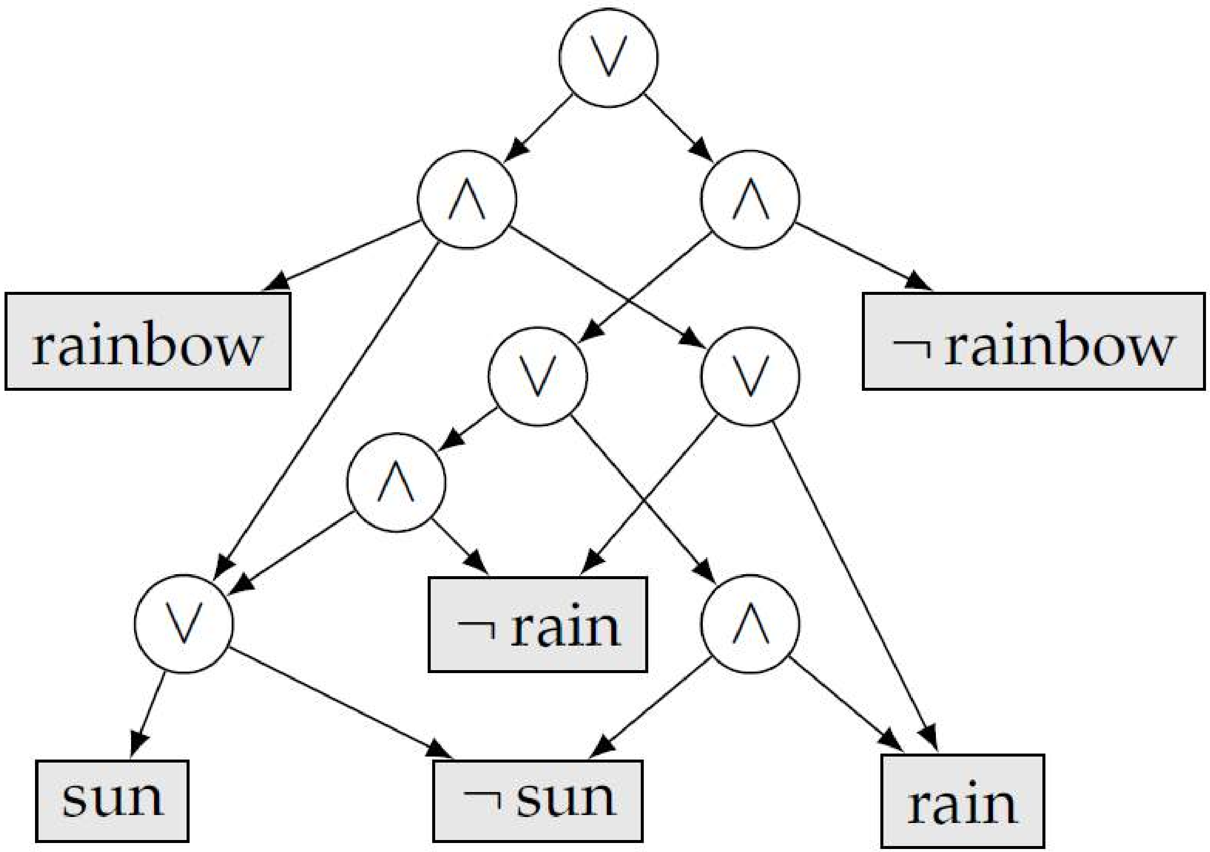
Oct 28, 2018

StarAI Research Lab UCLA

Foundation:
Logical Circuit Languages

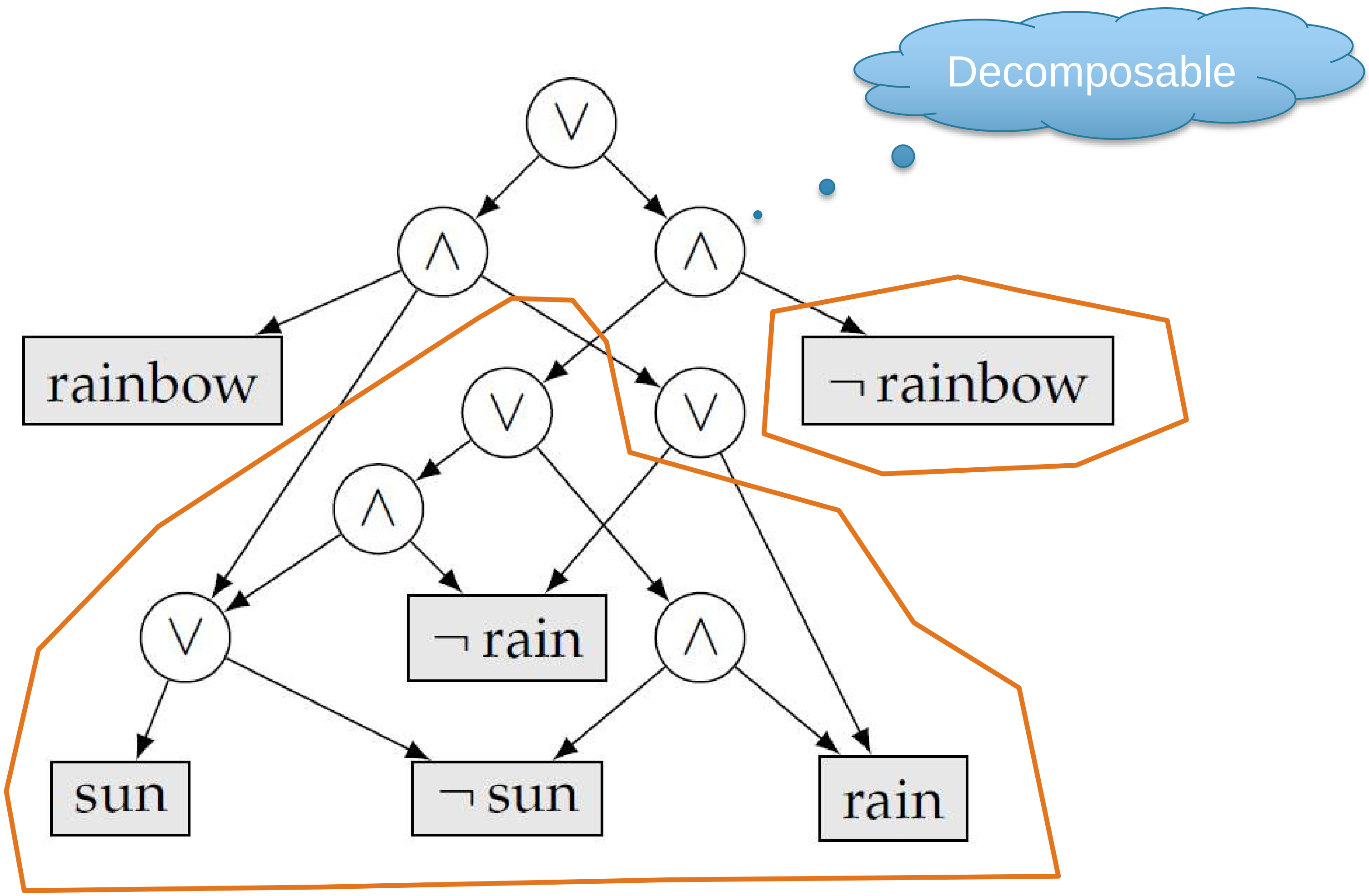
Negation Normal Form Circuits

Δ = (sun ∧ rain ⇒ rainbow)



[Darwiche 2002]

Decomposable Circuits



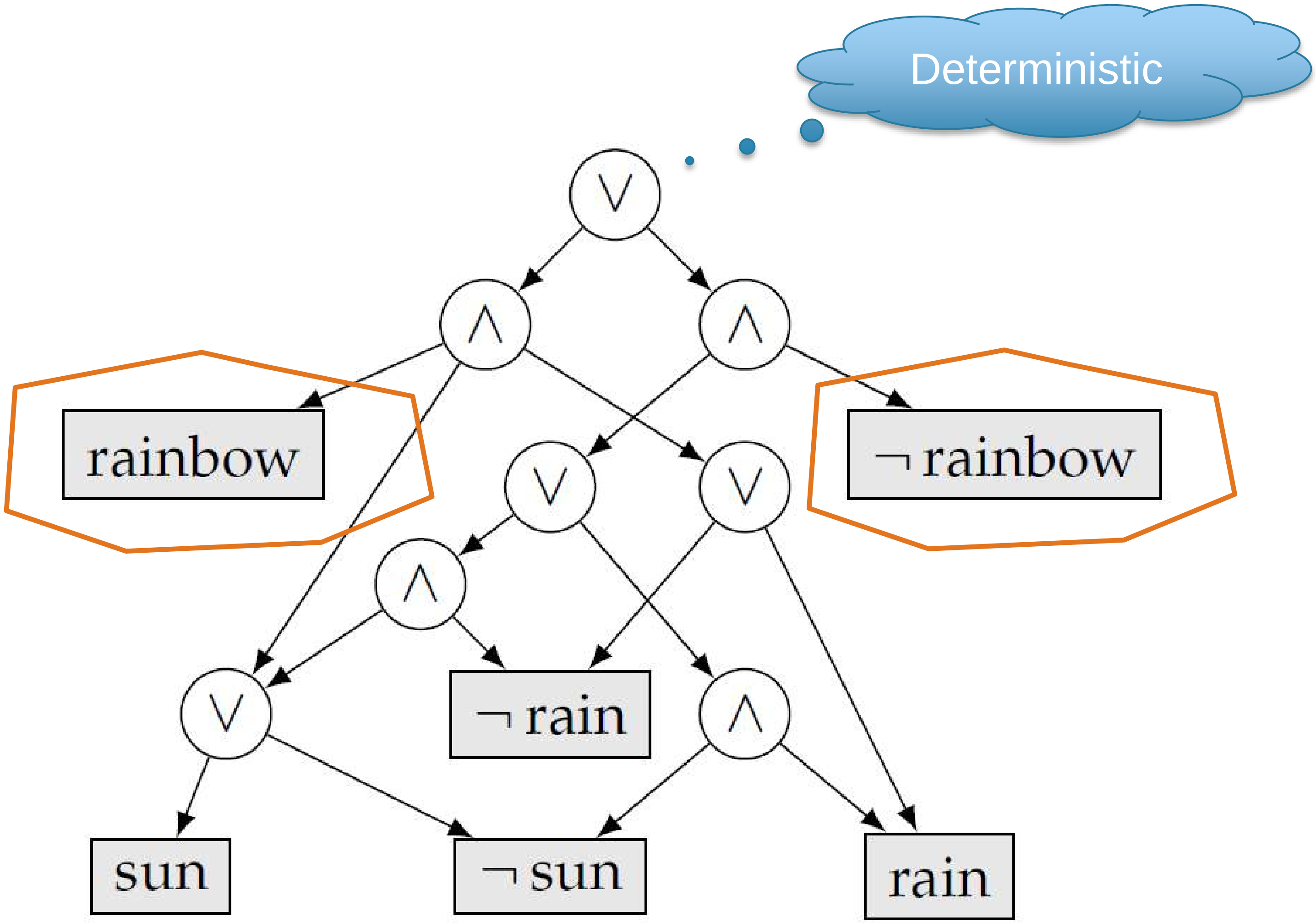
[Darwiche 2002]

Tractable for Logical Inference

- Is there a solution? (SAT) ✓
  - SAT($\alpha \vee \beta$) iff SAT(α) or SAT(β) (*always*)
  - SAT($\alpha \wedge \beta$) iff SAT(α) and SAT(β) (*decomposable*)
- How many solutions are there? (#SAT)

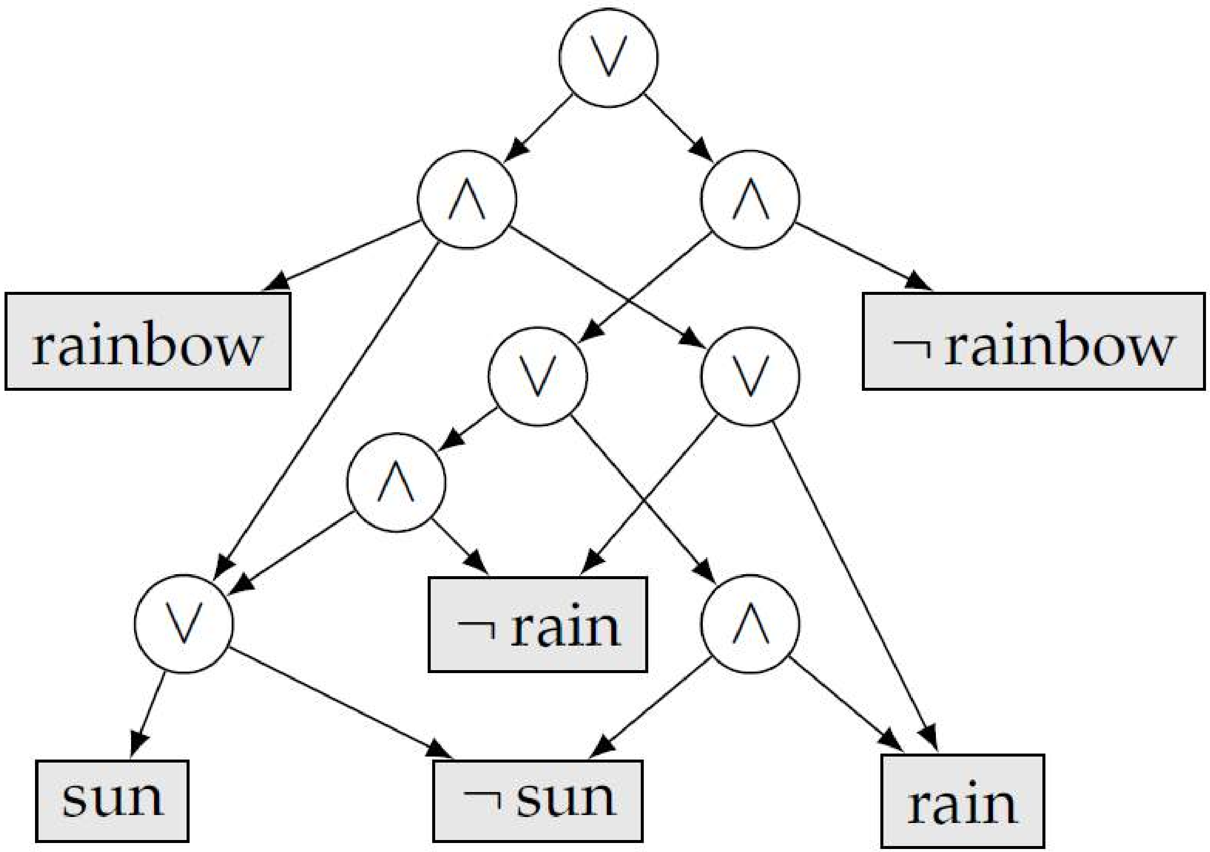
- Complexity linear in circuit size ☺

Deterministic Circuits

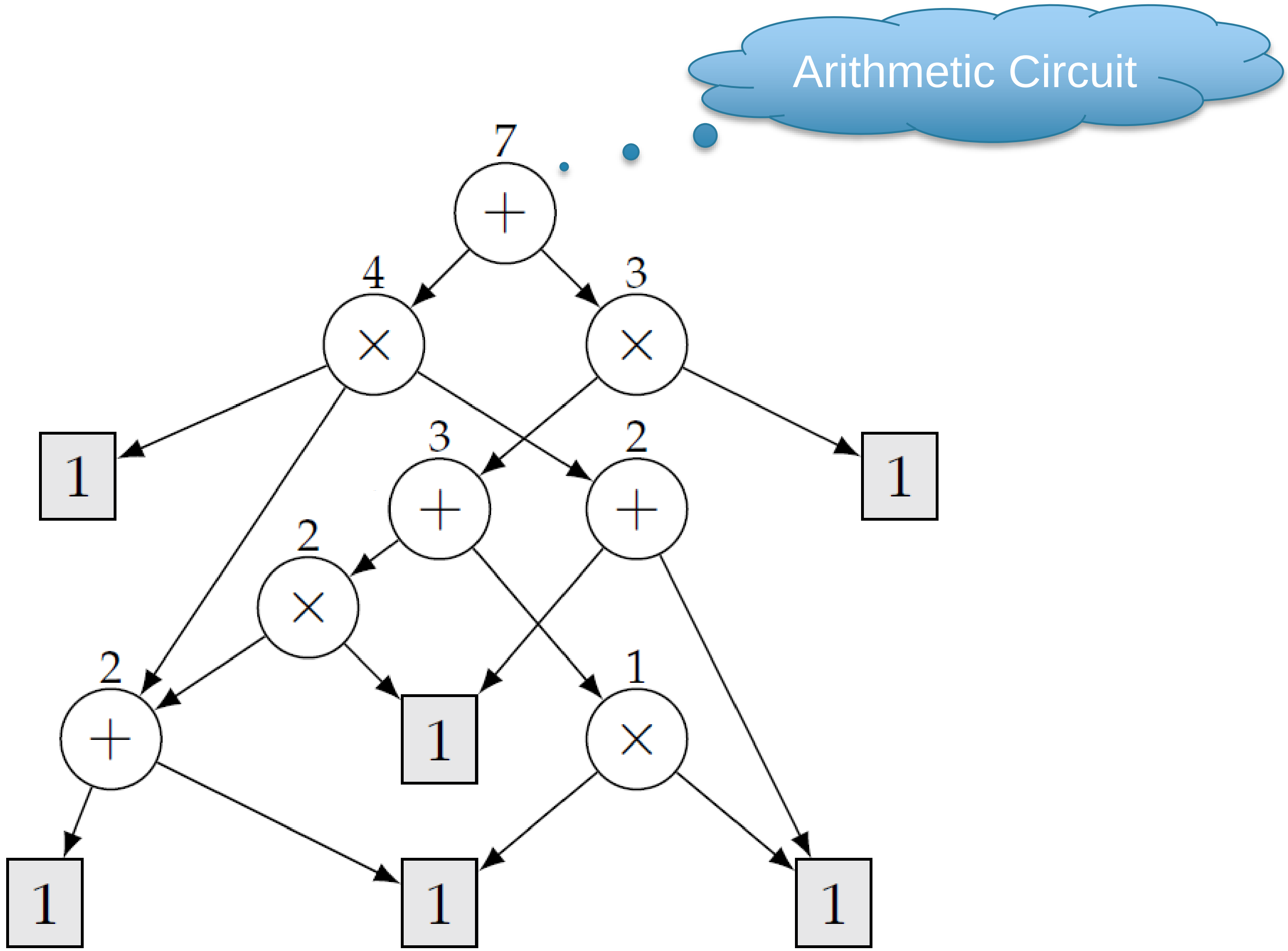


[Darwiche 2002]

How many solutions are there? (#SAT)



How many solutions are there? (#SAT)

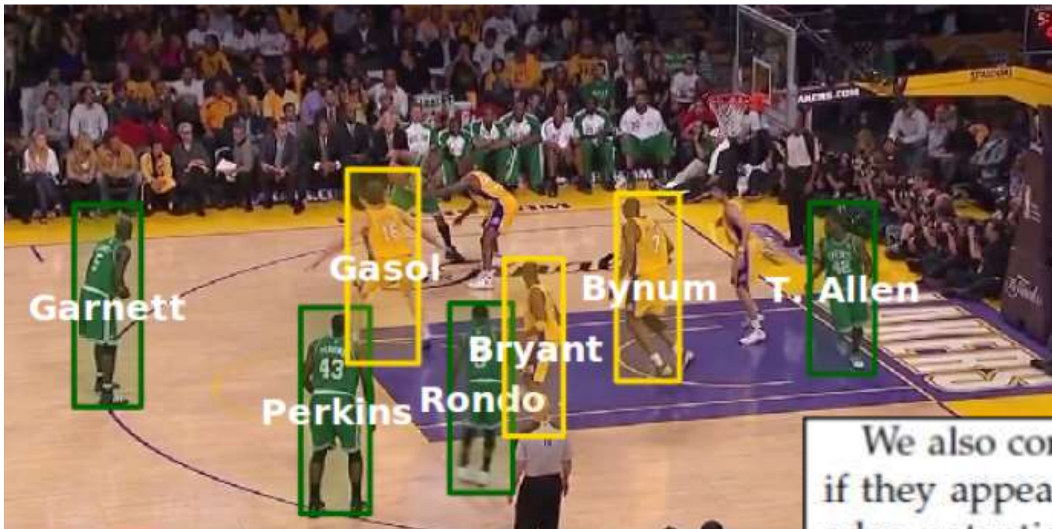


Tractable for Logical Inference

- Is there a solution? (SAT) ✓
- How many solutions are there? (#SAT) ✓
- Stricter languages (e.g., BDD, SDD):
  - Equivalence checking ✓
  - Conjoin/disjoint/negate circuits ✓
- Complexity linear in circuit size ☺
- Compilation into circuit language by either
  - ↓ exhaustive SAT solver
  - ↑ conjoin/disjoin/negate

Learning with
Logical Constraints

Motivation: Video



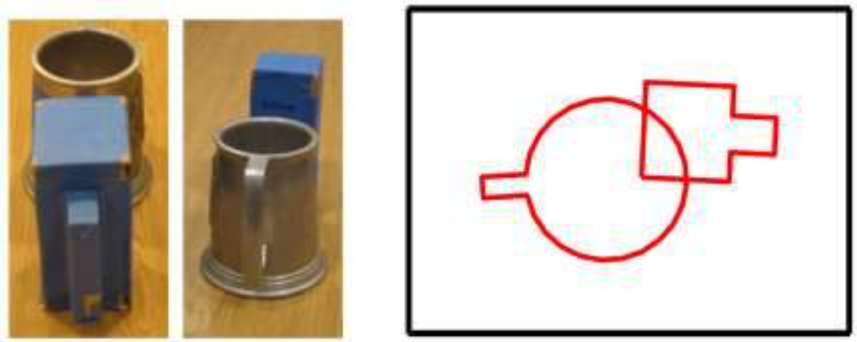
We also connect all pairs of identity nodes $y_{t,i}$ and $y_{t,j}$ if they appear in the same time t. We then introduce an edge potential that enforces mutual exclusion:

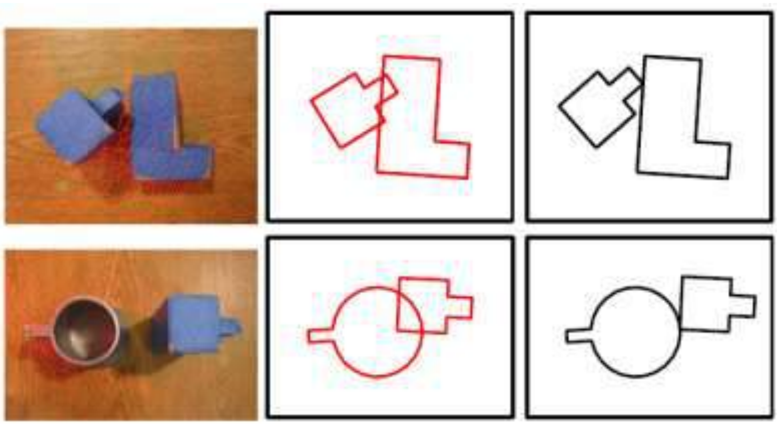
$$\psi_{\text{mutex}}(y_{t,i}, y_{t,j}) = \begin{cases} 1 & \text{if } y_{t,i} \neq y_{t,j} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

This potential specifies the constraint that a player can be appear only *once* in a frame. For example, if the i-th detection $y_{t,i}$ has been assign to Bryant, $y_{t,j}$ cannot have the same identity because Bryant is impossible to appear twice in a frame.

[Lu, W. L., Ting, J. A., Little, J. J., & Murphy, K. P. (2013). Learning to track and identify players from broadcast sports videos.]

Motivation: Robotics





> The method developed in this paper can be used in a broad variety of semantic mapping and object manipulation tasks, providing an efficient and effective way to incorporate collision constraints into a recursive state estimator, obtaining optimal or near-optimal solutions.

[Wong, L. L., Kaelbling, L. P., & Lozano-Perez, T., Collision-free state estimation. ICRA 2012]

Motivation: Language

- Non-local dependencies:

  At least one verb in each sentence

- Sentence compression

  If a modifier is kept, its subject is also kept

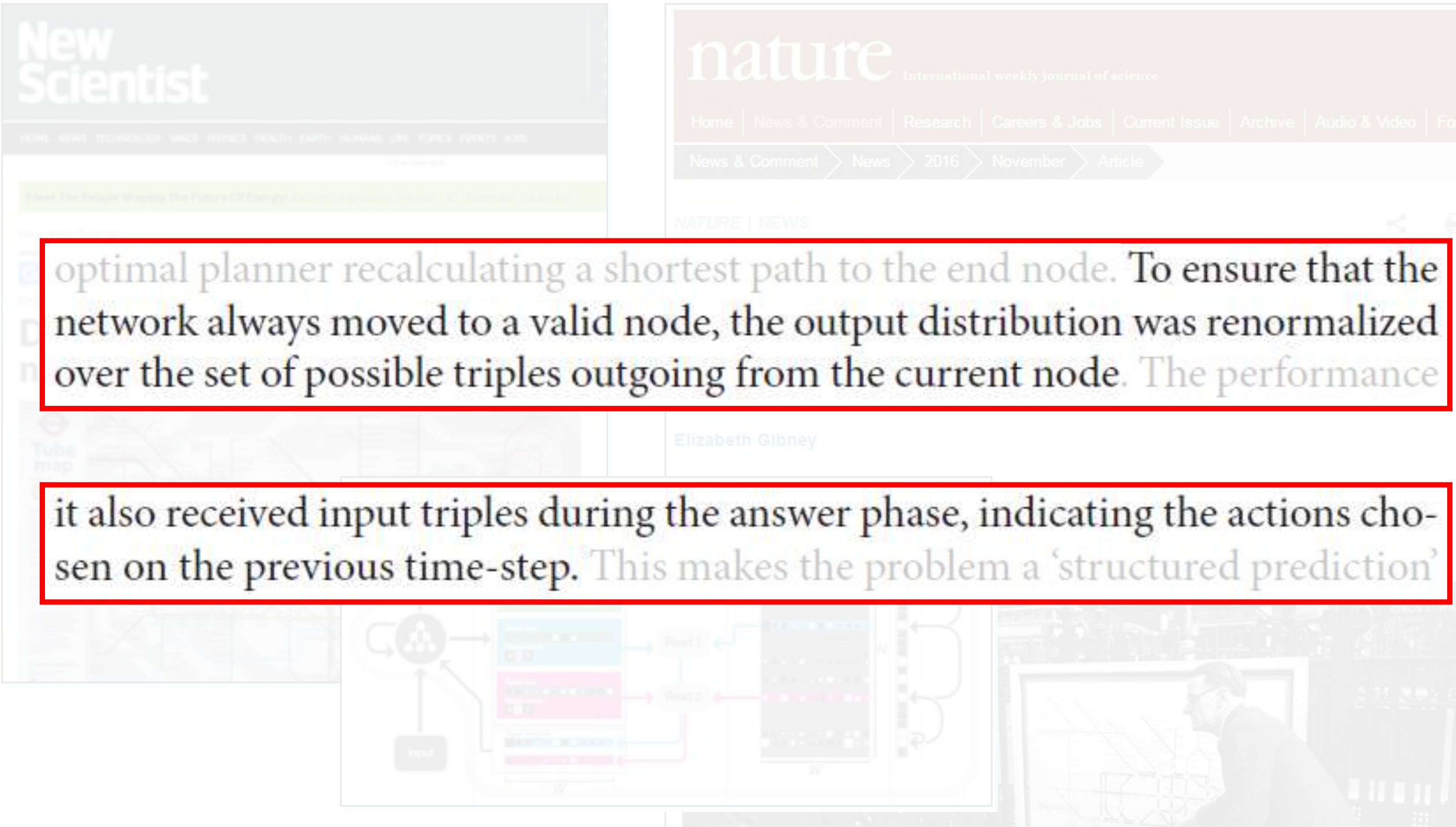
- Information extraction
- Semantic role labeling

… and many more!

Citations	
Start	The citation must start with author or editor.
AppearsOnce	Each field must be a consecutive list of words, and can appear at most once in a citation.
Punctuation	State transitions must occur on punctuation marks.
BookJournal	The words *proc*, *journal*, *proceedings*, *ACM* are *JOURNAL* or *BOOKTITLE*.
...	...
TechReport	The words *tech*, *technical* are *TECH_REPORT*.
Title	Quotations can appear only in titles.
Location	The words *CA*, *Australia*, *NY* are *LOCATION*.

[Chang, M., Ratinov, L., & Roth, D. (2008). Constraints as prior knowledge],…, [Chang, M. W., Ratinov, L., & Roth, D. (2012). Structured learning with constrained conditional models.], [https://en.wikipedia.org/wiki/Constrained_conditional_model]

Motivation: Deep Learning



optimal planner recalculating a shortest path to the end node. To ensure that the network always moved to a valid node, the output distribution was renormalized over the set of possible triples outgoing from the current node. The performance

it also received input triples during the answer phase, indicating the actions chosen on the previous time-step. This makes the problem a 'structured prediction'

[Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwińska, A., et al.. (2016). Hybrid computing using a neural network with dynamic external memory. *Nature*, *538*(7626), 471-476.]

Running Example

Courses:

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

Constraints

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

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

Structured Space

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 (**P**) or Logic (**L**).
- Probability is a prerequisite for AI (**A**).
- The prerequisites for KR (**K**) 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

Boolean Constraints

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

$$P \vee L$$
$$A \Rightarrow P$$
$$K \Rightarrow (P \vee L)$$

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

Learning in Structured Spaces

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

Data + **Constraints**

(Background Knowledge)
(Physics)

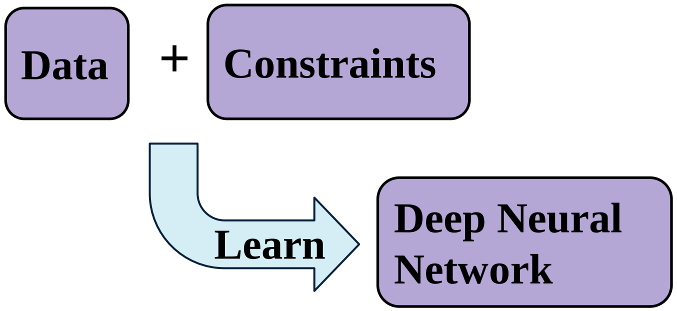
$$P \vee L$$
$$A \Rightarrow P$$
$$K \Rightarrow (P \vee L)$$

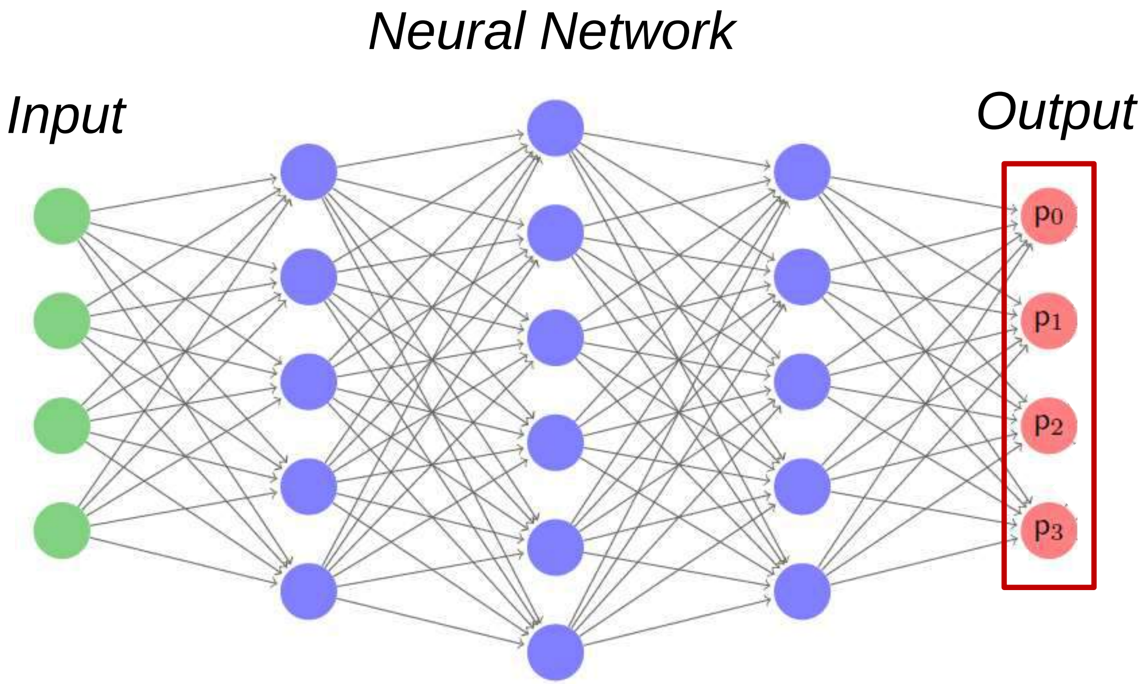
Learn → **ML Model**

Today's machine learning tools don't take knowledge as input! ☹

Deep Learning with Logical Constraints

Deep Learning with Logical Knowledge





Output is probability vector **p**, not Boolean logic!

Semantic Loss

Q: How close is output **p** *to satisfying constraint?*

Answer: Semantic loss function $L(\alpha,\mathbf{p})$

- Axioms, for example:
  - If **p** is Boolean then $L(\mathbf{p},\mathbf{p}) = 0$
  - If α implies β then $L(\alpha,\mathbf{p}) \geq L(\beta,\mathbf{p})$ (*α more strict*)
- Properties:
  - If α is equivalent to β then $L(\alpha,\mathbf{p}) = L(\beta,\mathbf{p})$
  - If **p** is Boolean and satisfies α then $L(\alpha,\mathbf{p}) = 0$

SEMANTIC Loss!

Semantic Loss: Definition

<u>Theorem</u>: Axioms imply unique semantic loss:

$$\mathrm{L}^{\mathrm{s}}(\alpha, \mathsf{p}) \propto -\log \sum_{\mathbf{x} \models \alpha} \prod_{i : \mathbf{x} \models X_i} \mathsf{p}_i \prod_{i : \mathbf{x} \models \neg X_i} (1 - \mathsf{p}_i)$$

Probability of getting **x** after flipping coins with prob. **p**

Probability of satisfying α after flipping coins with prob. **p**

Example: Exactly-One

- Data must have some label

  We agree this must be one of the 10 digits:

- Exactly-one constraint

  → For 3 classes: $\begin{cases} x_1 \lor x_2 \lor x_3 \\ \neg x_1 \lor \neg x_2 \\ \neg x_2 \lor \neg x_3 \\ \neg x_1 \lor \neg x_3 \end{cases}$

- Semantic loss:

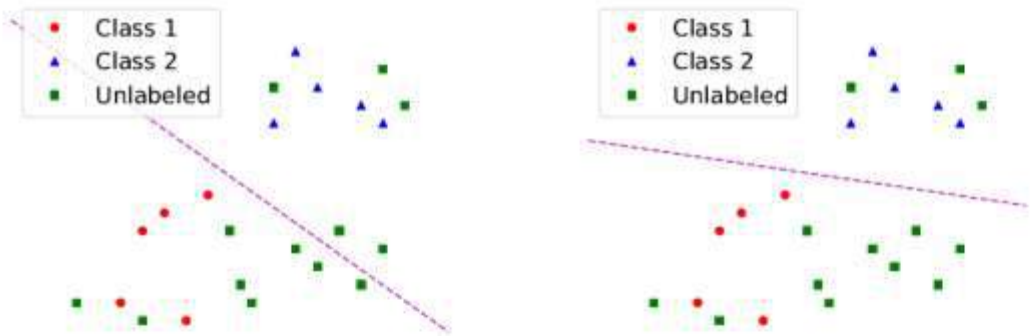
$$L^s(\text{exactly-one}, \mathrm{p}) \propto -\log \sum_{i=1}^{n} \mathrm{p}_i \underbrace{\prod_{j=1, j \neq i}^{n} (1 - \mathrm{p}_j)}_{\text{Only } x_i = 1 \text{ after flipping coins}}$$

Only $x_i = 1$ after flipping coins

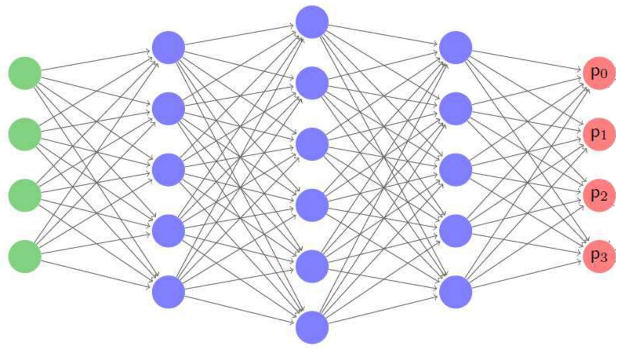
Exactly one true x after flipping coins

Semi-Supervised Learning

- Intuition: Unlabeled data must have some label



- Minimize exactly-one semantic loss on unlabeled data



Train with
$existing\ loss + w \cdot semantic\ loss$

MNIST Experiment

Accuracy % with # of used labels	100	1000	ALL
AtlasRBF (Pitelis et al., 2014)	91.9 (± 0.95)	96.32 (± 0.12)	98.69
Deep Generative (Kingma et al., 2014)	96.67(± 0.14)	97.60(± 0.02)	99.04
Virtual Adversarial (Miyato et al., 2016)	97.67	98.64	99.36
Ladder Net (Rasmus et al., 2015)	**98.94** (±0.37)	**99.16** (±0.08)	99.43 (± 0.02)
Baseline: MLP, Gaussian Noise	78.46 (±1.94)	94.26 (±0.31)	99.34 (±0.08)
Baseline: Self-Training	72.55 (±4.21)	87.43 (±3.07)	
MLP with Semantic Loss	98.38 (±0.51)	98.78 (±0.17)	99.36 (±0.02)

Competitive with state of the art in semi-supervised deep learning

FASHION Experiment

(a) Confidently Correct (b) Unconfidently Correct (c) Unconfidently Incorrect (d) Confidently Incorrect

Accuracy % with # of used labels	100	500	1000	ALL
Ladder Net (Rasmus et al., 2015)	81.46 (±0.64)	85.18 (±0.27)	86.48 (± 0.15)	90.46
Baseline: MLP, Gaussian Noise	69.45 (±2.03)	78.12 (±1.41)	80.94 (±0.84)	89.87
MLP with Semantic Loss	**86.74** (±0.71)	**89.49** (±0.24)	89.67 (±0.09)	89.81

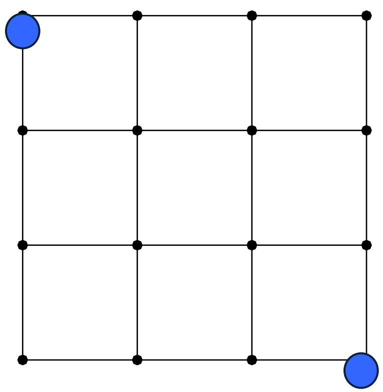
Outperforms Ladder Nets!

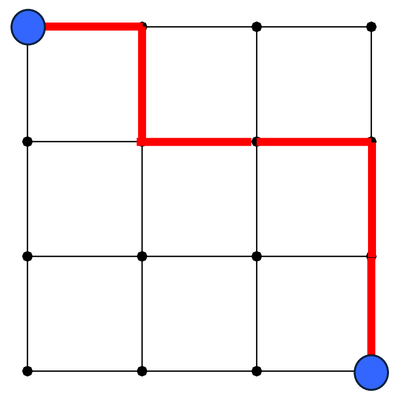
Same conclusion on CIFAR10

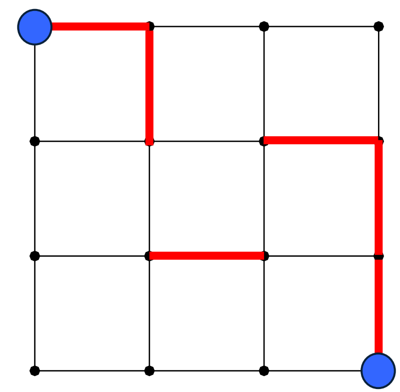
Accuracy % with # of used labels	4000	ALL
CNN Baseline in Ladder Net	76.67 (± 0.61)	90.73
Ladder Net (Rasmus et al., 2015)	79.60 (±0.47)	
Baseline: CNN, Whitening, Cropping	77.13	90.96
CNN with Semantic Loss	**81.79**	90.92

What about real constraints?
Paths

cf. Nature paper







Good variable assignment (represents route)

184

Bad variable assignment (does not represent route)

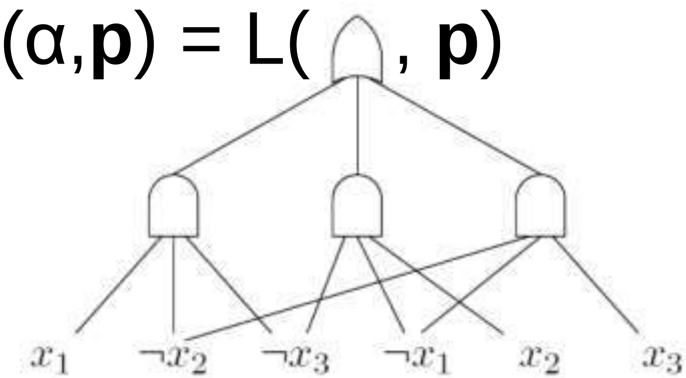
16,777,032

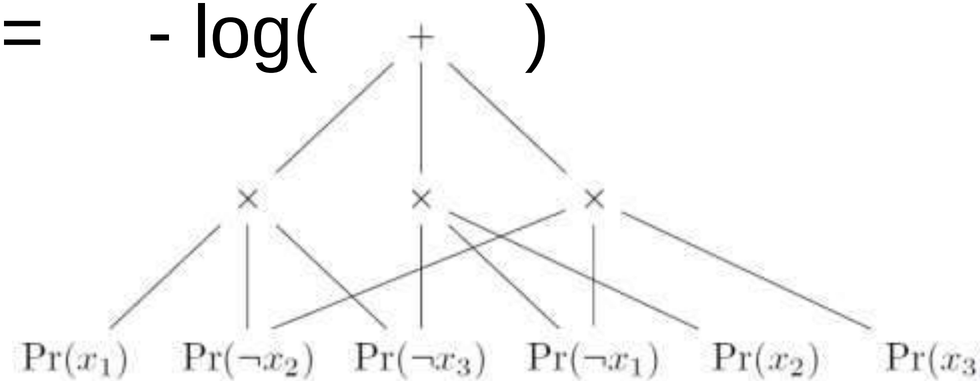
Unstructured probability space: 184+16,777,032 = 2^{24}

Space easily encoded in logical constraints ☺ [Nishino et al.]

How to Compute Semantic Loss?

- In general: #P-hard ☹
- With a logical circuit for α: Linear!
- Example: exactly-one constraint:

L(α,**p**) = L(

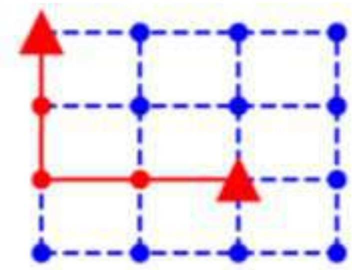
, **p**) = - log(

)

- *Why?* Decomposability and determinism!

Predict Shortest Paths

Add semantic loss
for path constraint



Test accuracy %	Coherent	Incoherent	Constraint
5-layer MLP	5.62	**85.91**	6.99
Semantic loss	**28.51**	83.14	**69.89**

Is prediction the shortest path? **This is the real task!**

Are individual edge predictions correct?

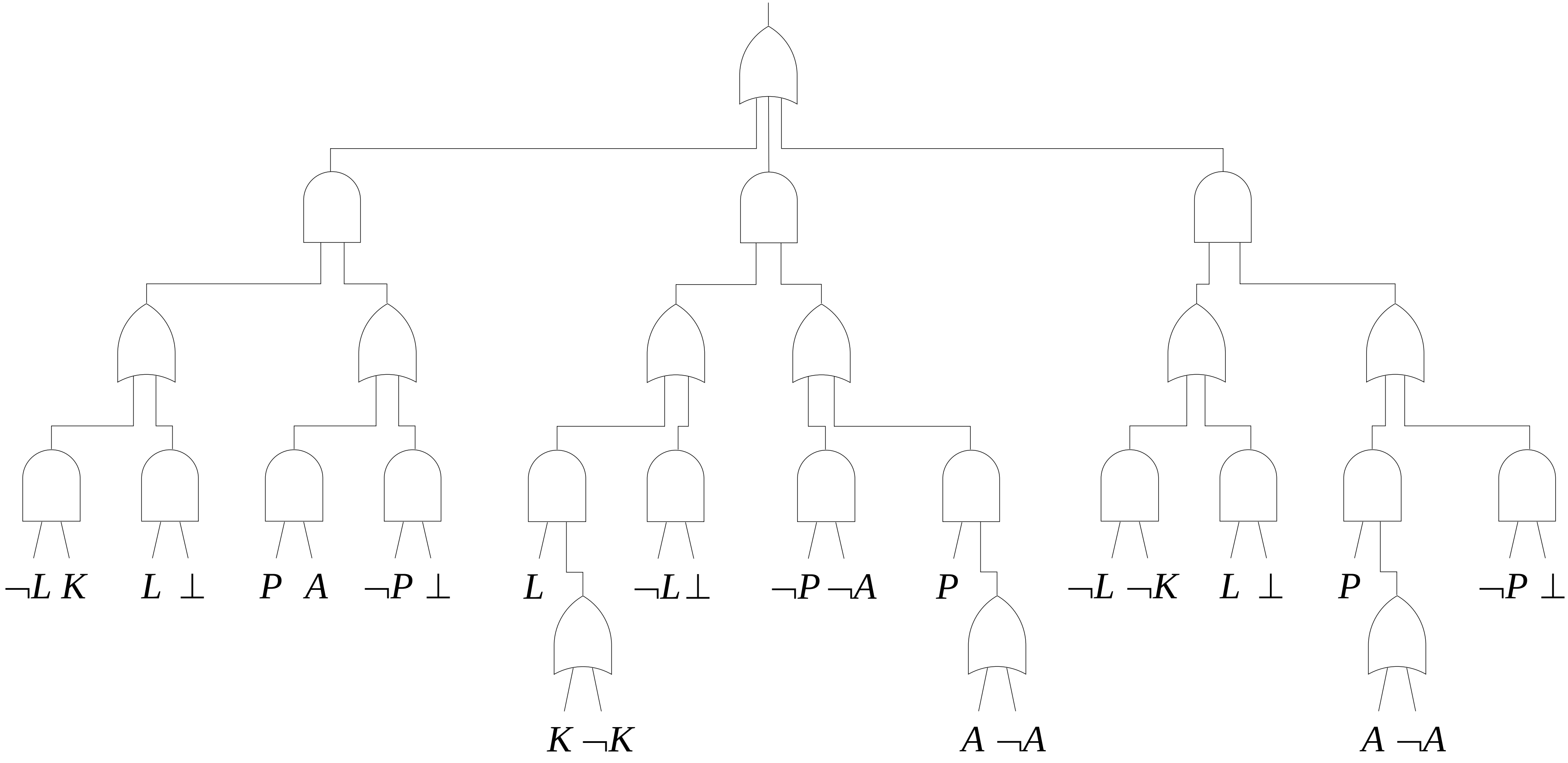
Is output a path?

(same conclusion for predicting sushi preferences, see paper)

Probabilistic Circuits

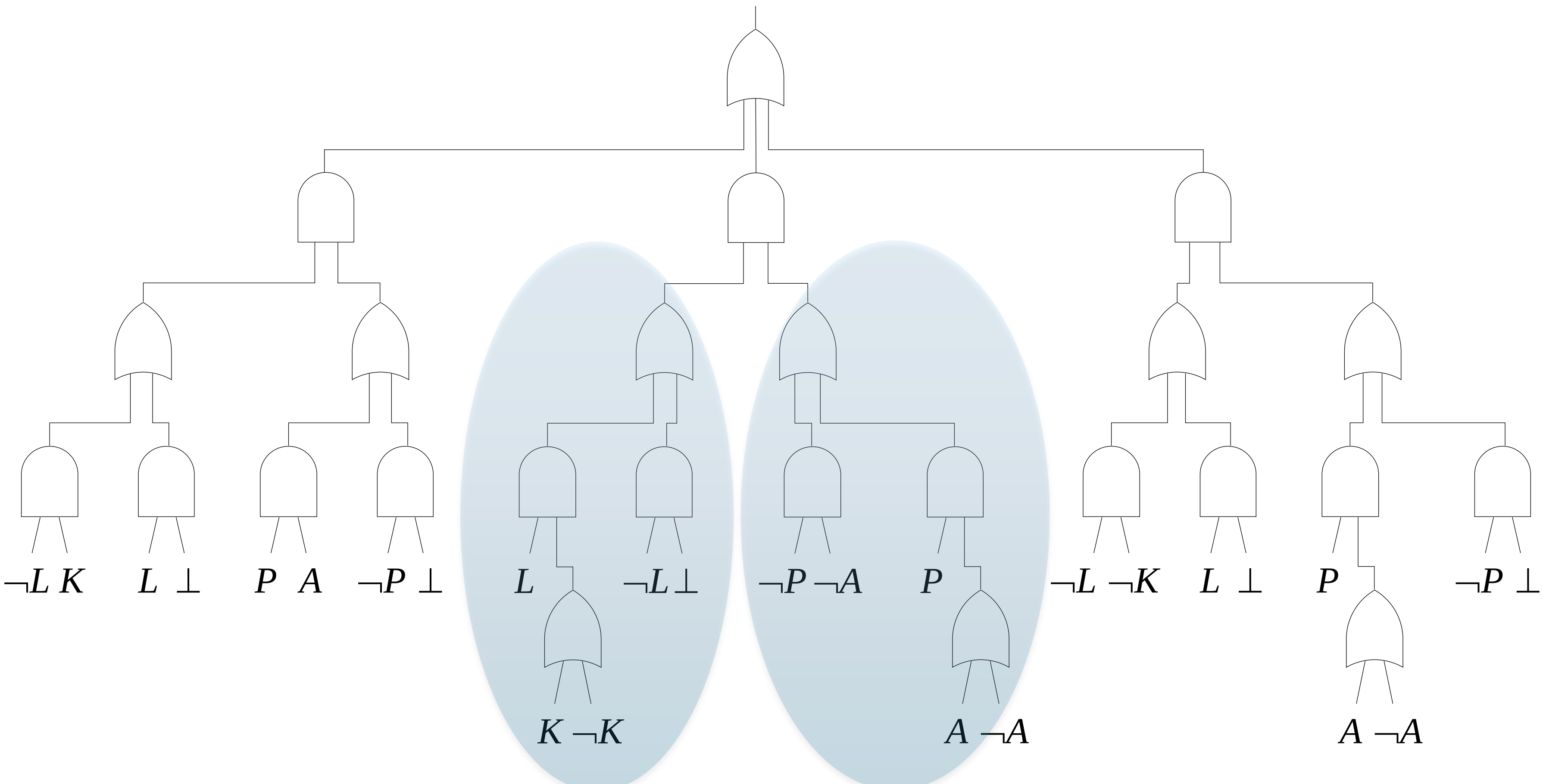
Logical Circuits

$P \vee L$
$A \Rightarrow P$
$K \Rightarrow (P \vee L)$



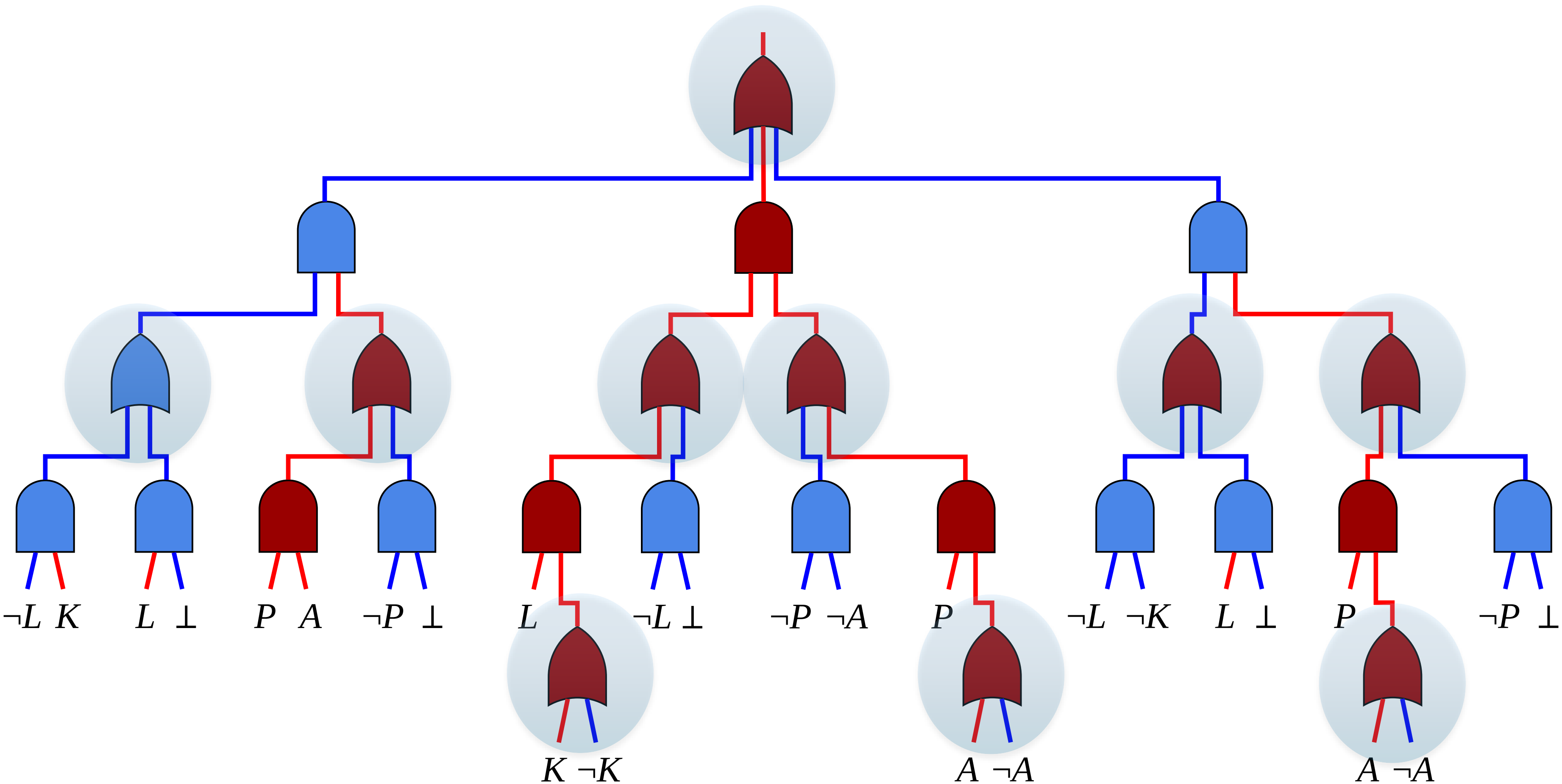
Can we represent a **distribution** over the solutions to the constraint?

Recall: Decomposability



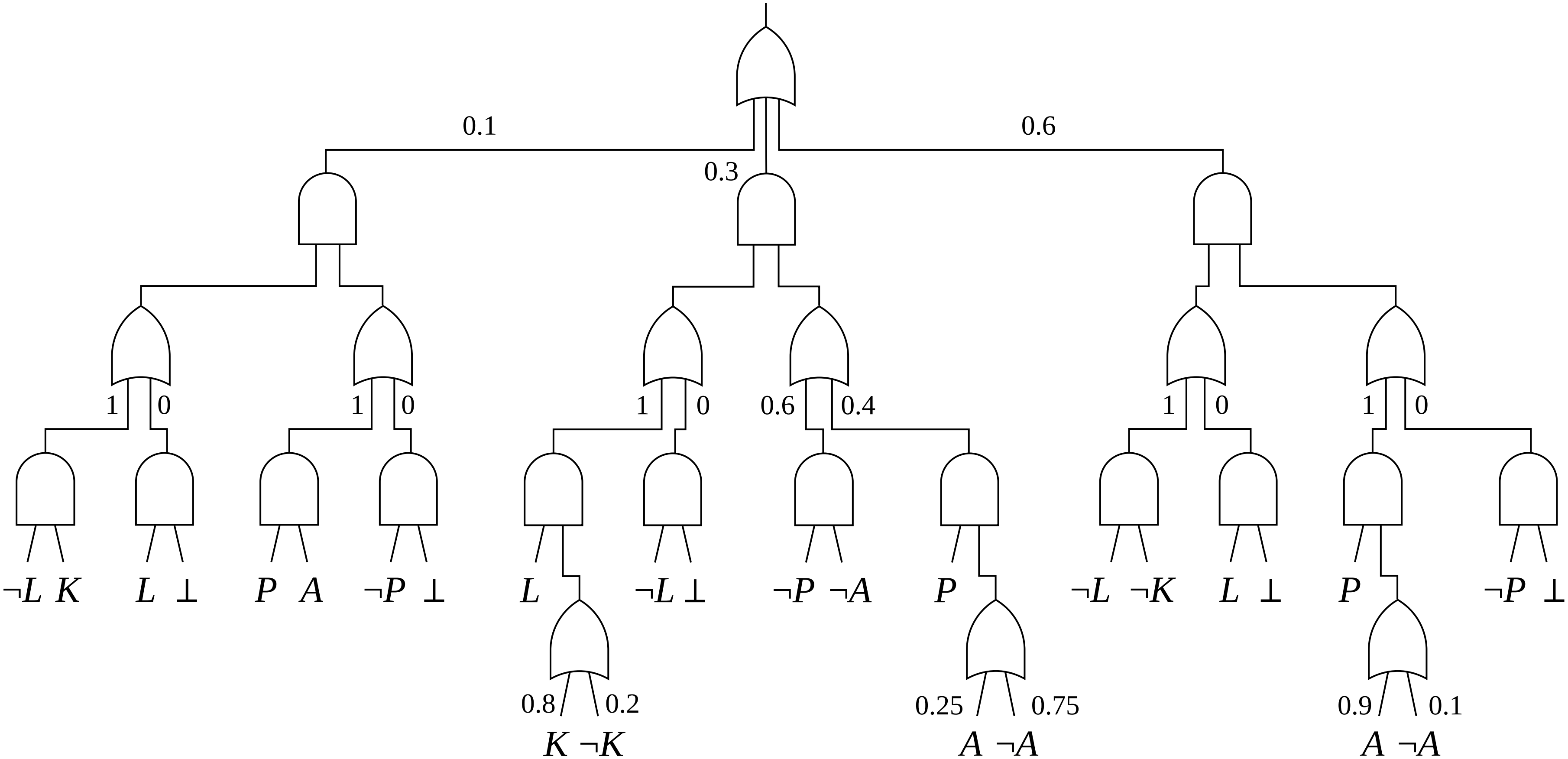
AND gates have disjoint input circuits

Recall: Determinism



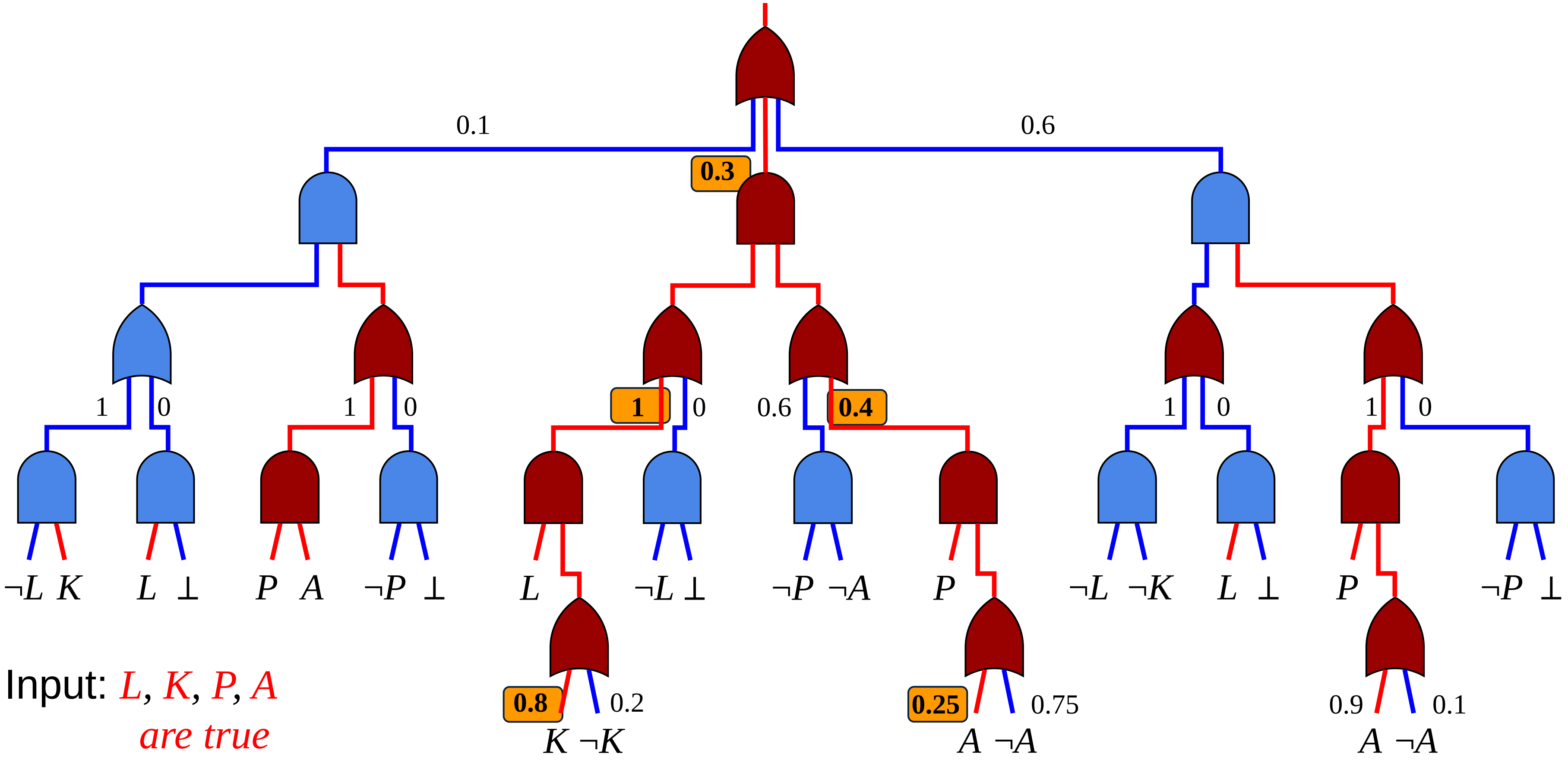
Input: L, K, P, A are true and ¬L, ¬K, ¬P, ¬A are false
Property: OR gates have at most one true input wire

PSDD: Probabilistic SDD



Syntax: assign a normalized probability to each OR gate input

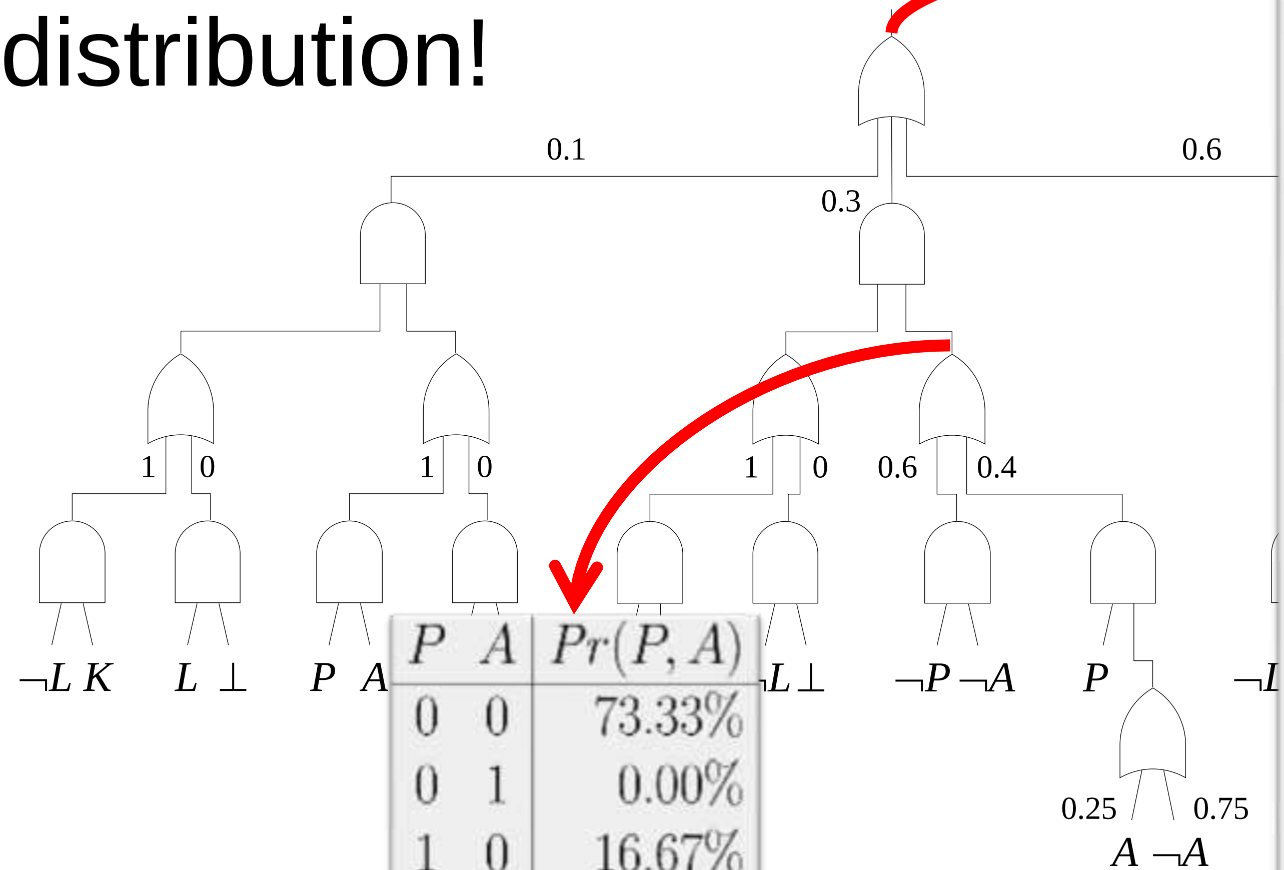
PSDD: Probabilistic SDD



Input: *L*, *K*, *P*, *A*
are true

$$\Pr(L,K,P,A) = 0.3 \times 1 \times 0.8 \times 0.4 \times 0.25 = \mathbf{0.024}$$

Each node represents a normalized distribution!



P	A	$Pr(P, A)$
0	0	73.33%
0	1	0.00%
1	0	16.67%
1	1	10.00%

L	K	P	A	$Pr(L, K, P, A)$
0	0	0	0	0.00%
0	0	0	1	0.00%
0	0	1	0	6.00%
0	0	1	1	54.00%
0	1	0	0	0.00%
0	1	0	1	0.00%
0	1	1	0	0.00%
0	1	1	1	10.00%
1	0	0	0	4.40%
1	0	0	1	0.00%
1	0	1	0	1.00%
1	0	1	1	0.60%
1	1	0	0	17.6%
1	1	0	1	0.00%
1	1	1	0	4.00%
1	1	1	1	2.40%

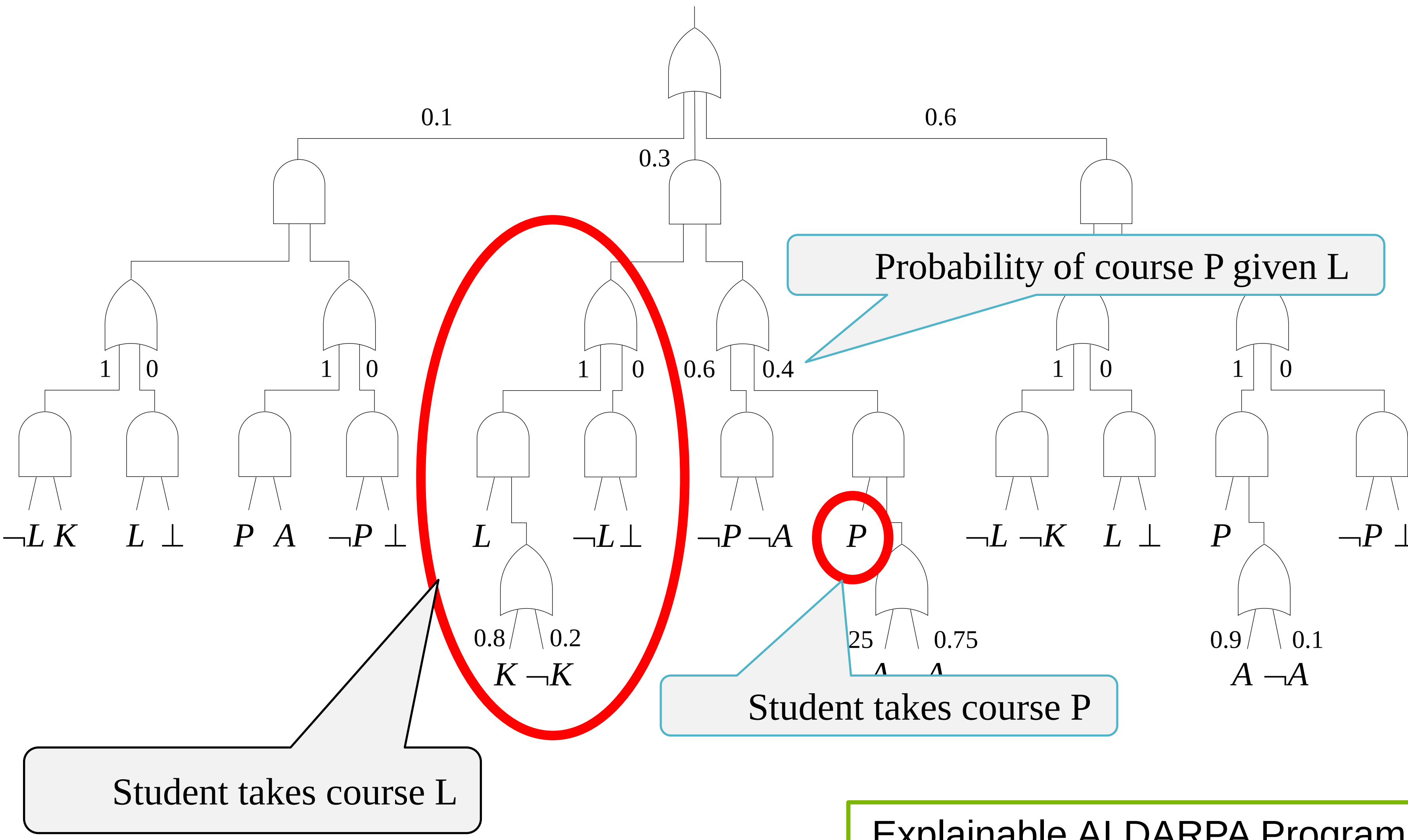
Can read probabilistic independences off the circuit structure

Tractable for Probabilistic Inference

- **MAP inference**:
  Find most-likely assignment to x given y
  (otherwise NP-hard)
- Computing **conditional probabilities** Pr(x|y)
  (otherwise #P-hard)
- **Sample** from Pr(x|y)

- Algorithms linear in circuit size ☺
  (pass up, pass down, similar to backprop)

Parameters are Interpretable



Explainable AI DARPA Program

Learning Probabilistic Circuit Parameters

Learning Algorithms

- Closed form
  max likelihood
  from complete 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

- One pass over data to estimate Pr(x|y)

Not a lot to say: very easy! ☺

- Where does the structure come from?

  For now: simply compiled from constraint…

Combinatorial Objects: Rankings

rank	sushi
1	fatty tuna
2	sea urchin
3	salmon roe
4	shrimp
5	tuna
6	squid
7	tuna roll
8	see eel
9	egg
10	cucumber roll

rank	sushi
1	shrimp
2	sea urchin
3	salmon roe
4	fatty tuna
5	tuna
6	squid
7	tuna roll
8	see eel
9	egg
10	cucumber roll

10 items:
3,628,800
rankings

20 items:
2,432,902,008,176,640,000
rankings

Combinatorial Objects: Rankings

rank	sushi
1	fatty tuna
2	sea urchin
3	salmon roe
4	shrimp
5	tuna
6	squid
7	tuna roll
8	see eel
9	egg
10	cucumber roll

- Predict Boolean Variables: A_{ij} - item i at position j
- Constraints:

  each item *i* assigned to a unique position (*n* constraints)

  $$\bigvee_j A_{ij} \wedge \left(\bigwedge_{k \neq j} \neg A_{ik} \right)$$

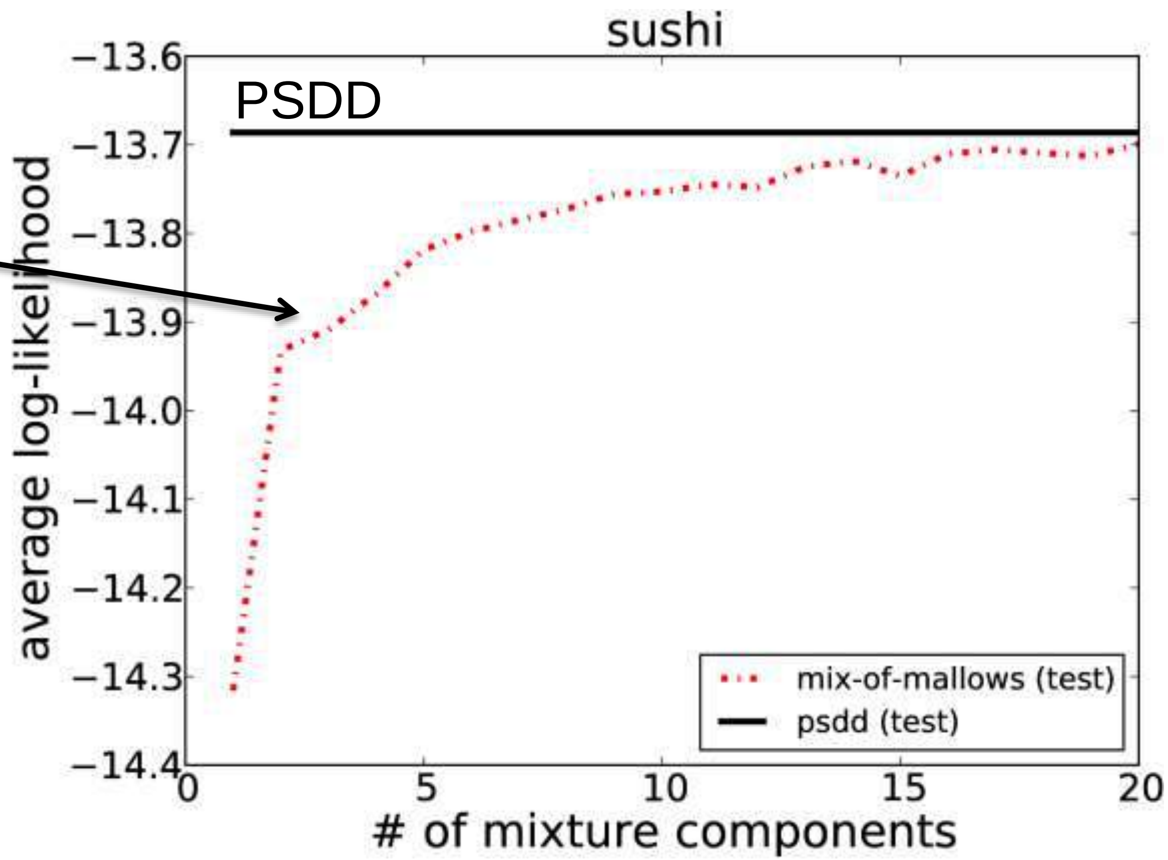
  each position *j* assigned a unique item (*n* constraints)

  $$\bigvee_i A_{ij} \wedge \left(\bigwedge_{k \neq i} \neg A_{kj} \right)$$

Learning Preference Distributions

Special-purpose distribution: Mixture-of-Mallows

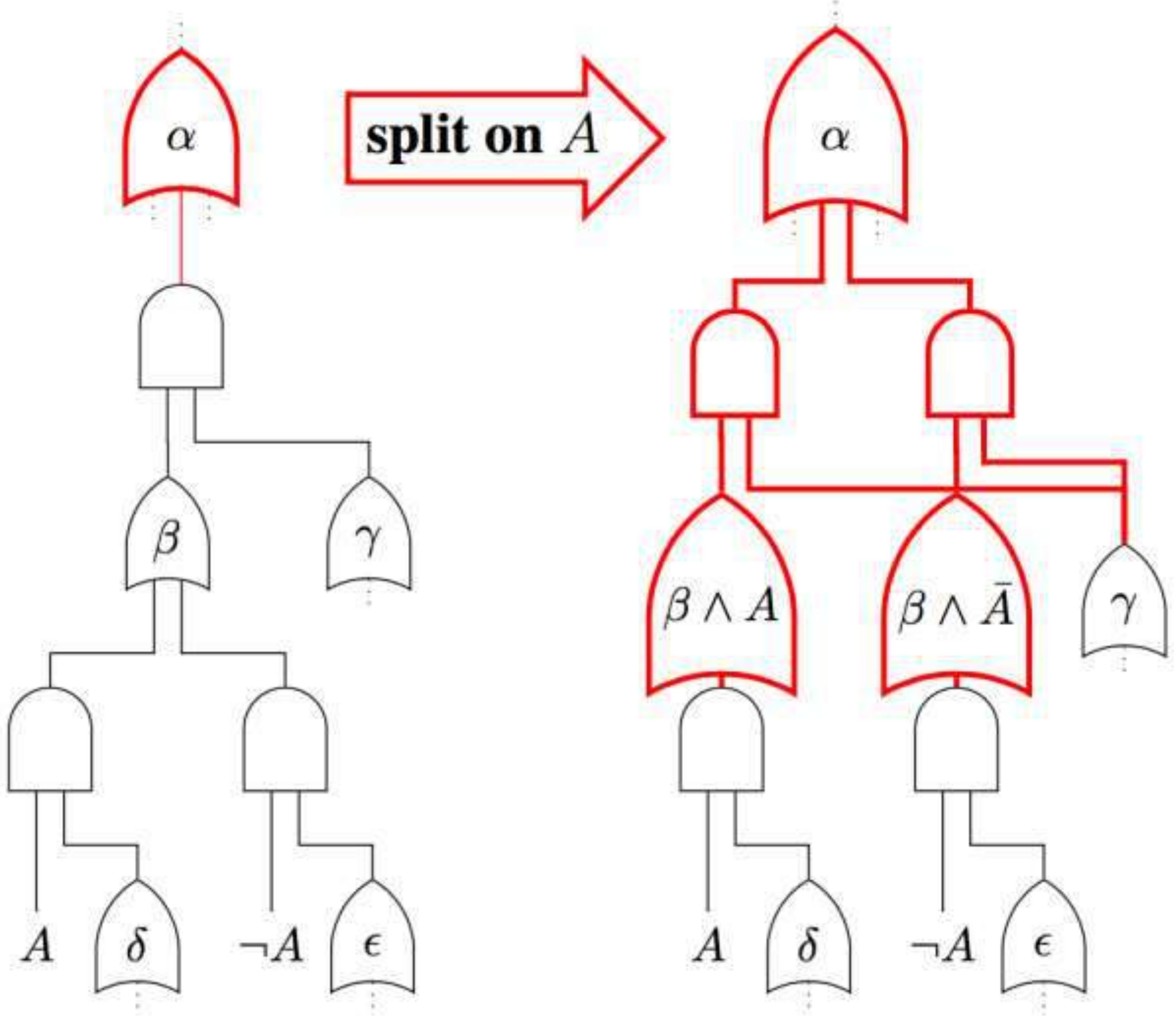
- # of components from 1 to 20
- EM with 10 random seeds
- Implementation of Lu & Boutilier



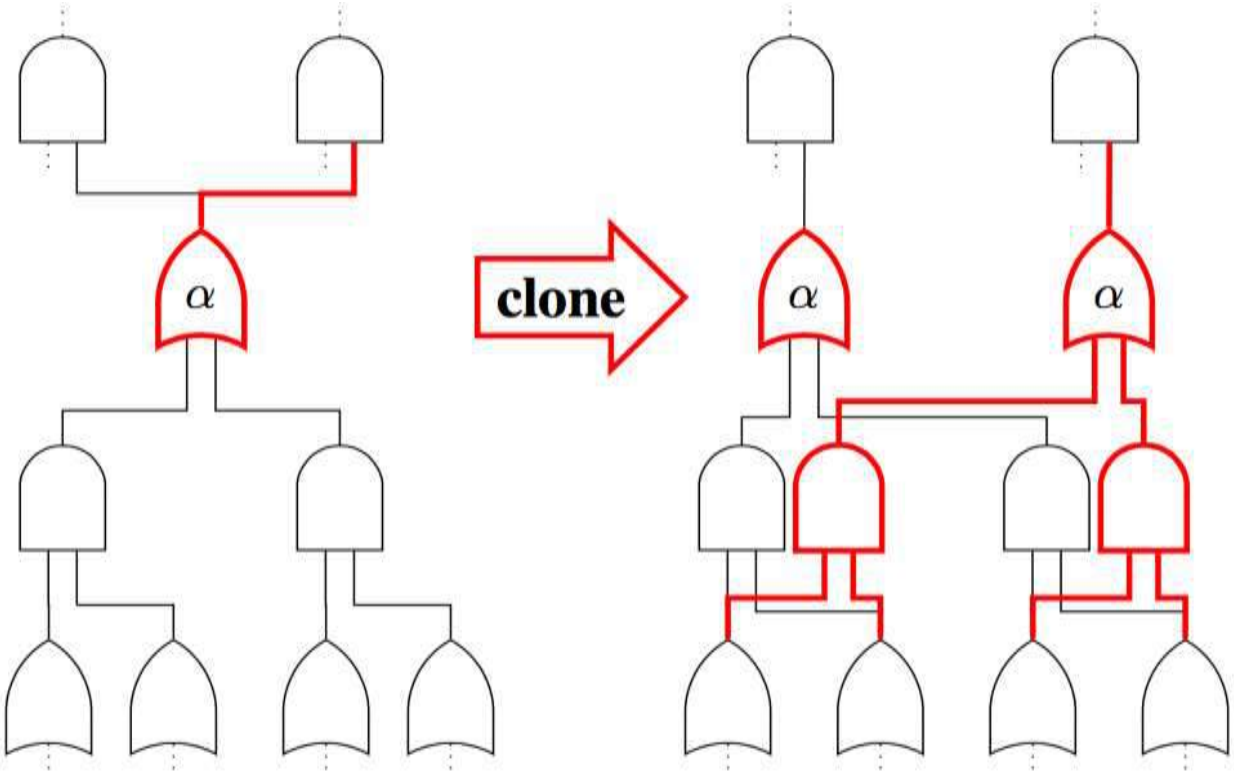
Circuit structure does not even depend on data!

Learning Probabilistic Circuit Structure

Structure Learning Primitive



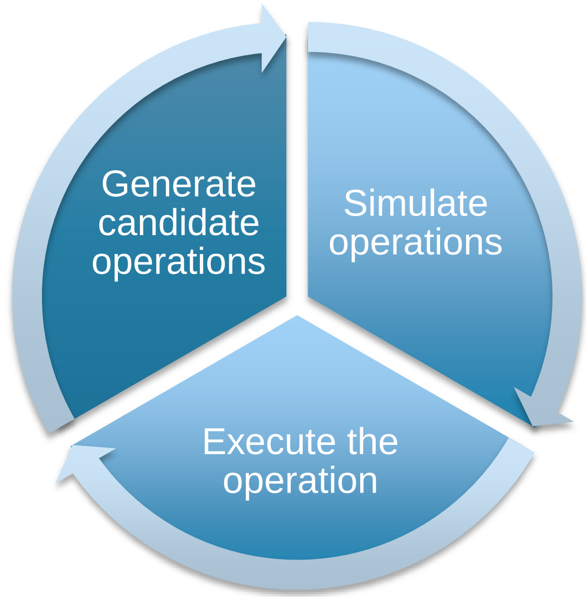
Structure Learning Primitive



Primitives maintain PSDD properties and constraint of root!

LearnPSDD Algorithm

1. (Vtree learning)*
2. Construct the most naïve PSDD
3. LearnPSDD (search for better structure)

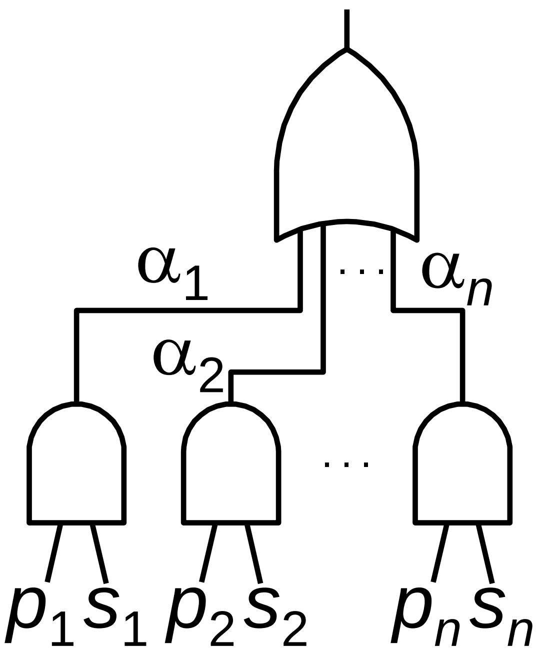


$$\text{score} = \frac{\ln \mathcal{L}(r' \mid \mathcal{D}) - \ln \mathcal{L}(r \mid \mathcal{D})}{\text{size}(r') - \text{size}(r)}$$

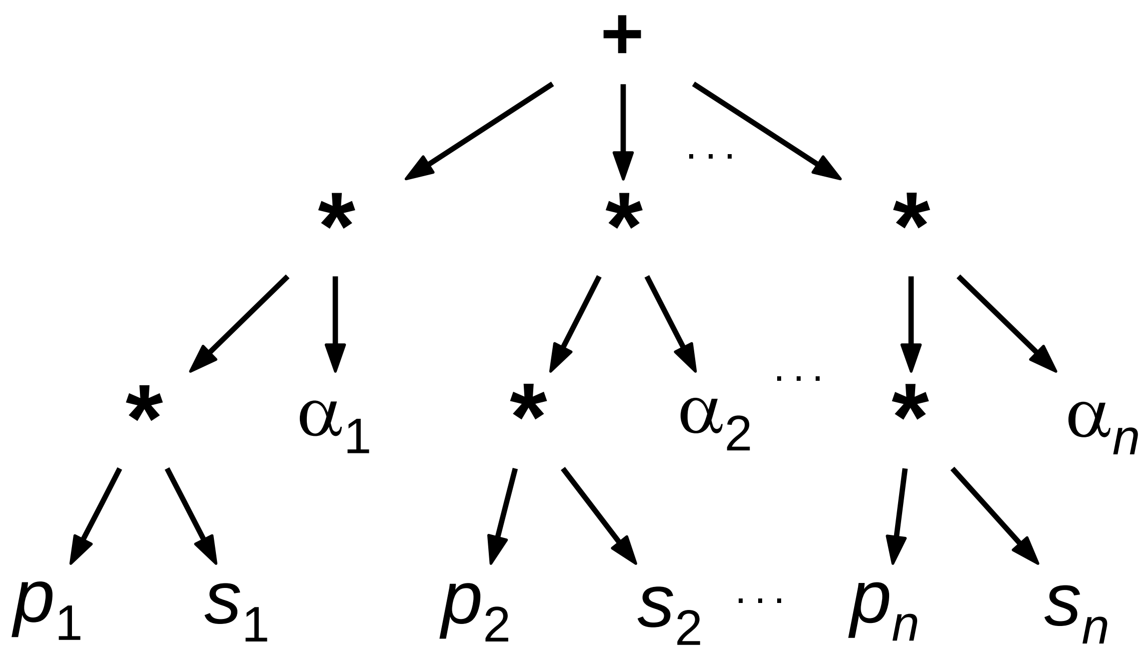
Works with or without logical constraint.

PSDDs

…are Sum-Product Networks
…are Arithmetic Circuits



PSDD



AC

Experiments on 20 datasets

Datasets	|Var|	|Train|	|Valid|	|Test|	LearnPSDD		EM-LearnPSDD		SearchSPN	Merged L-SPN		Merged O-SPN	
					LL	Size	LL	Size	LL	LL	Size	LL	Size
NLTCS	16	16181	2157	3236	−6.03†*	3170	**−6.03***	2147	−6.07	−6.04	3988	−6.05	1152
MSNBC	17	291326	38843	58265	−6.05†	8977	**−6.04***	3891	−6.06	−6.46	2440	−6.08	9478
KDD	64	1800992	19907	34955	−2.16†	14974	**−2.12***	9182	**−2.16**	−2.14	6670	−2.19	16608
Plants	69	17412	2321	3482	−14.93	13129	−13.79*	13951	−13.12†	**−12.69**	47802	−13.49	36960
Audio	100	15000	2000	3000	−42.53	13765	−41.98*	9721	−40.13†	**−40.02**	10804	−42.06	6142
Jester	100	9000	1000	4116	−57.67	11322	−53.47*	7014	−53.08†	**−52.97**	10002	−55.36	4996
Netflix	100	15000	2000	3000	−58.92	10997	−58.41*	6250	−56.91†	**−56.64**	11604	−58.64	6142
Accidents	111	12758	1700	2551	−34.13	10489	−33.64*	6752	−30.02†	**−30.01**	13322	−30.83	6846
Retail	135	22041	2938	4408	−11.13	4091	**−10.81***	7251	−10.97†	−10.87	2162	−10.95	3158
Pumsb-Star	163	12262	1635	2452	−34.11	10489	−33.67*	7965	−28.69†	**−24.11**	17604	−24.34	18338
DNA	180	1600	400	1186	−89.11*	6068	−92.67	14864	−81.76†	**−85.51**	4320	−87.49	1430
Kosarek	190	33375	4450	6675	−10.99†	11034	−10.81*	10179	−11.00	**−10.62**	5318	−10.98	6712
MSWeb	294	29441	32750	5000	−10.18†	11389	−9.97*	14512	−10.25	**−9.90**	16484	−10.06	12770
Book	500	8700	1159	1739	−35.90	15197	−34.97*	11292	−34.91†	**−34.76**	11998	−37.44	11916
EachMovie	500	4524	1002	591	−56.43*	12483	−58.01	16074	−53.28†	**−52.07**	15998	−58.05	19846
WebKB	839	2803	558	838	−163.42	10033	−161.09*	18431	−157.88†	**−153.55**	20134	−161.17	10046
Reuters-52	889	6532	1028	1530	−94.94	10585	−89.61*	9546	−86.38†	**−83.90**	46232	−87.49	28334
20NewsGrp.	910	11293	3764	3764	−161.41	12222	−161.09*	18431	−153.63†	**−154.67**	43684	−161.46	29016
BBC	1058	1670	225	330	−260.83	10585	**−253.19***	20327	−252.13†	−253.45	21160	−260.59	8454
AD	1556	2461	327	491	−30.49*	9666	−31.78	9521	−16.97†	−16.77	49790	**−15.39**	31070

Compared to SPN learners, LearnPSDD gives comparable performance yet smaller size

Learn Mixtures of PSDDs

Datasets	|Var|	LearnPSDD Ensemble	Best-to-Date
NLTCS	16	$-5.99^{\dagger}$	-6.00
MSNBC	17	$-6.04^{\dagger}$	$-6.04^{\dagger}$
KDD	64	$-2.11^{\dagger}$	-2.12
Plants	69	-13.02	$-11.99^{\dagger}$
Audio	100	-39.94	$-39.49^{\dagger}$
Jester	100	-51.29	$-41.11^{\dagger}$
Netflix	100	$-55.71^{\dagger}$	-55.84
Accidents	111	-30.16	$-24.87^{\dagger}$
Retail	135	$-10.72^{\dagger}$	-10.78
Pumsb-Star	163	-26.12	$-22.40^{\dagger}$
DNA	180	-88.01	$-80.03^{\dagger}$
Kosarek	190	$-10.52^{\dagger}$	-10.54
MSWeb	294	-9.89	$-9.22^{\dagger}$
Book	500	-34.97	$-30.18^{\dagger}$
EachMovie	500	-58.01	$-51.14^{\dagger}$
WebKB	839	-161.09	$-150.10^{\dagger}$
Reuters-52	889	-89.61	$-80.66^{\dagger}$
20NewsGrp.	910	-155.97	$-150.88^{\dagger}$
BBC	1058	-253.19	$-233.26^{\dagger}$
AD	1556	-31.78	$-14.36^{\dagger}$

State of the art on 6 datasets!

Q: "Help! I need to learn a discrete probability distribution…"
A: Learn mixture of PSDDs!

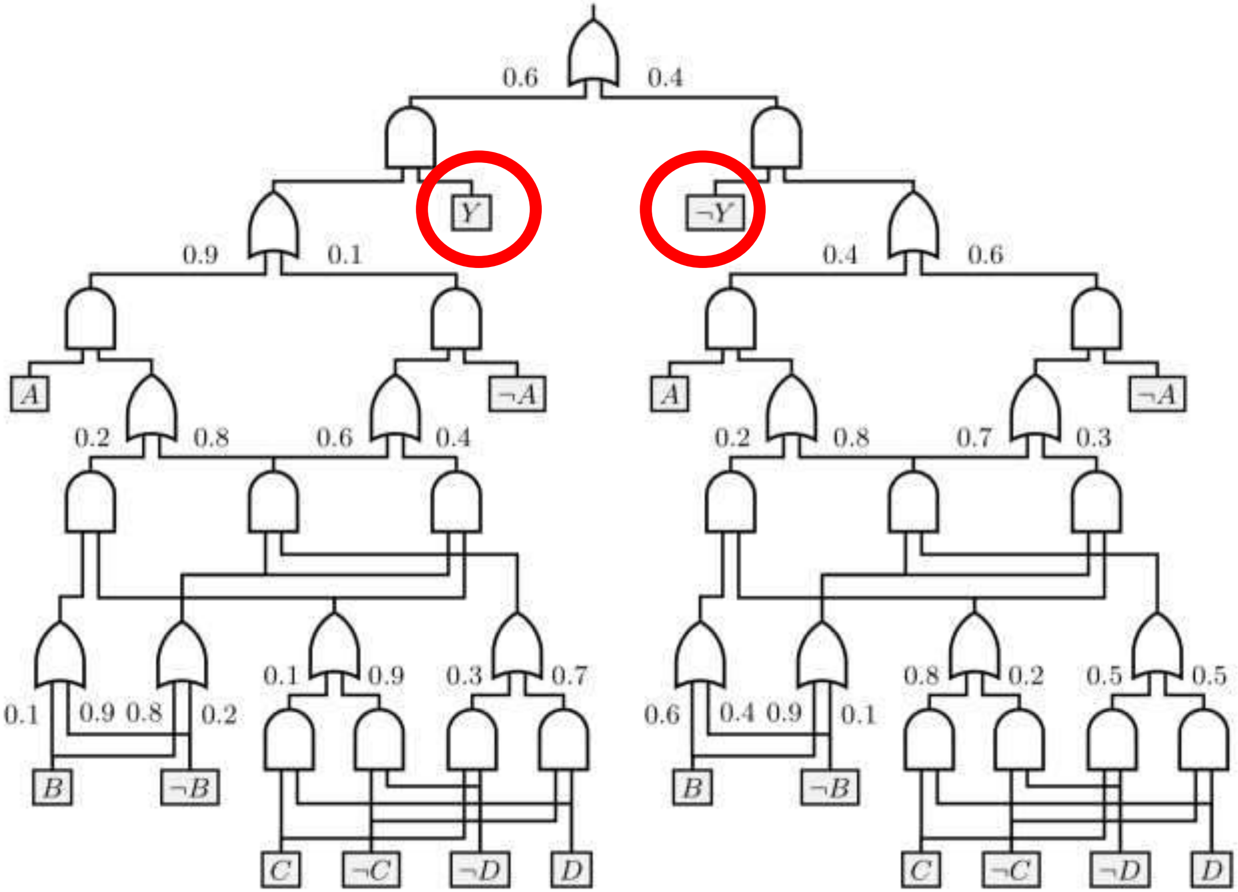
Strongly outperforms
- Bayesian network learners
- Markov network learners

Competitive with
- SPN learners
- Cutset network learners

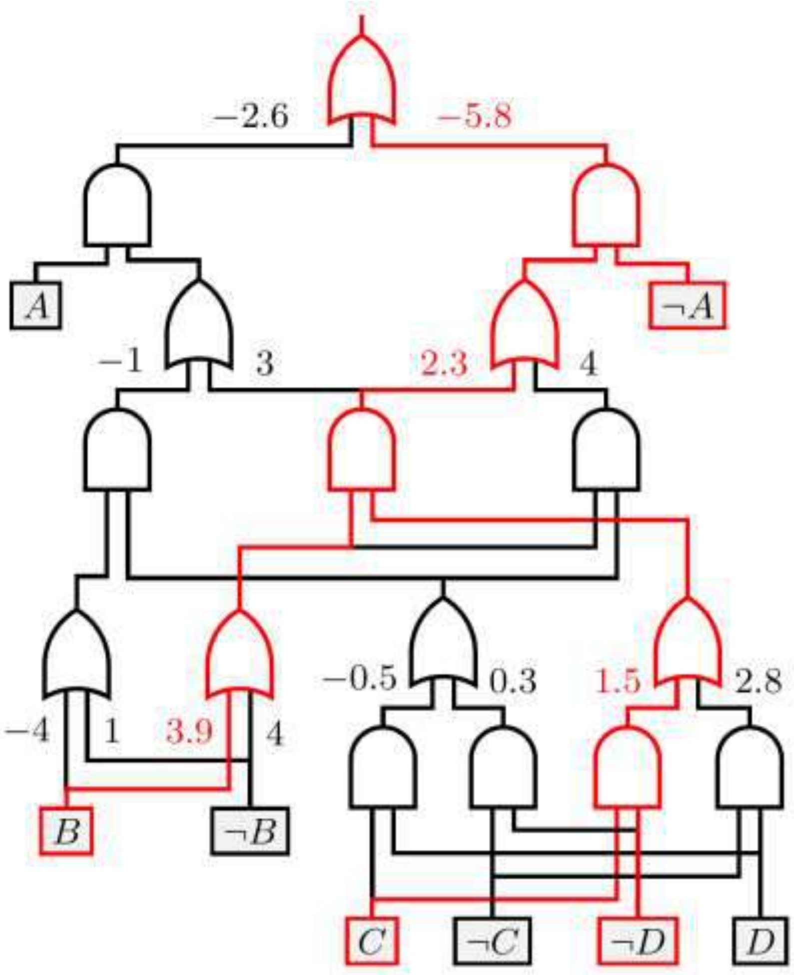
Logistic Circuits

What if I only want to classify Y?

$\Pr(Y, A, B, C, D)$



Logistic Circuits

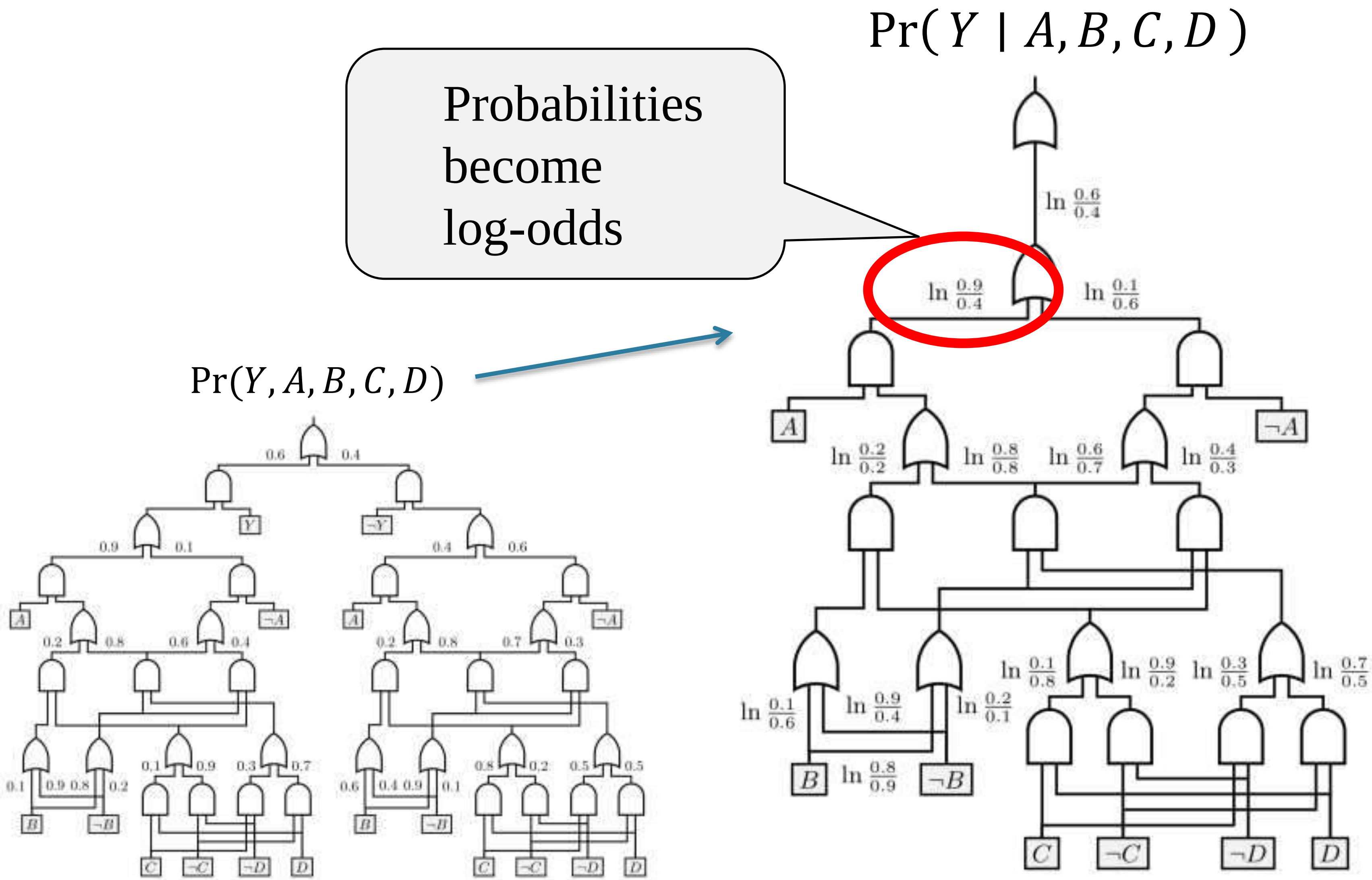


Represents $\Pr(Y \mid A, B, C, D)$

- Take all 'hot' wires
- Sum their weights
- Push through logistic function

A	B	C	D	$g_r(ABCD)$	$\Pr(Y=1 \mid ABCD)$
1	0	1	1	-3.1	4.31%
0	1	1	0	1.9	86.99%
1	1	1	0	5.8	99.70%

Logistic vs. Probabilistic Circuits



Parameter Learning

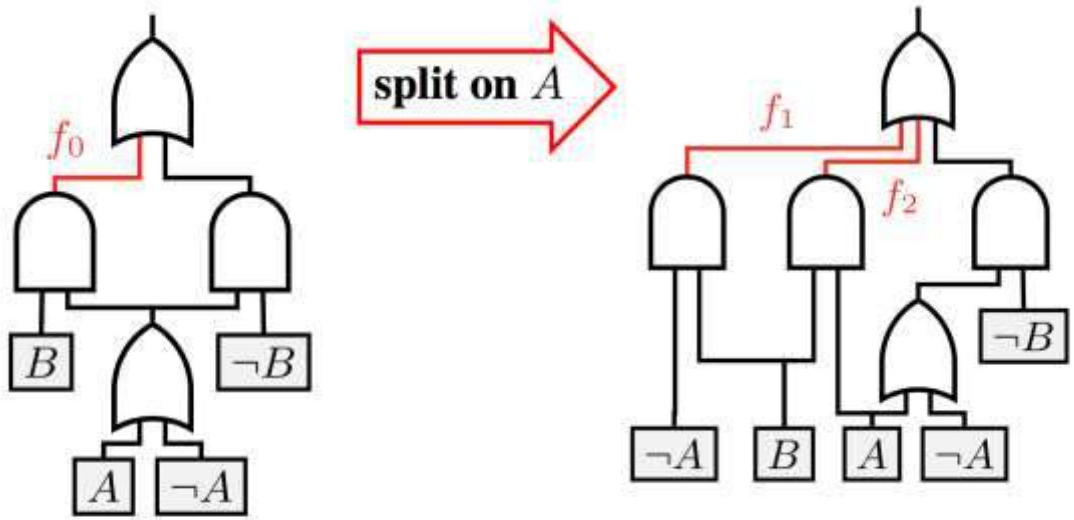
Reduce to logistic regression:

$$\Pr(Y = 1 \mid \mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{x} \cdot \boldsymbol{\theta})}$$

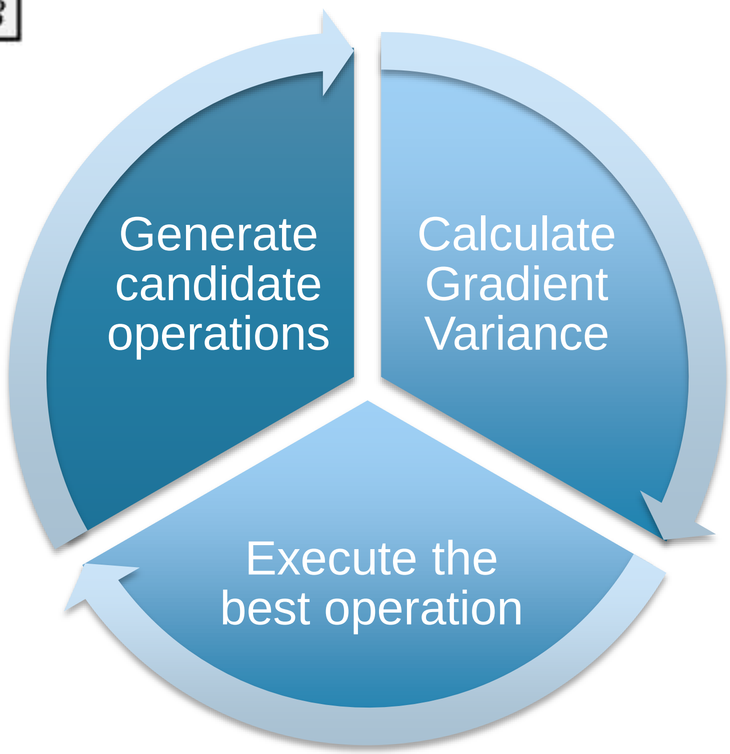
> Features associated with each wire
> "Global Circuit Flow" features

Learning parameters θ is convex optimization!

Logistic Circuit Structure Learning



Similar to LearnPSDD structure learning



Comparable Accuracy with Neural Nets

Accuracy % on Dataset	Mnist	Fashion
Baseline: Logistic Regression	85.3	79.3
Baseline: Kernel Logistic Regression	97.7	88.3
Random Forest	97.3	81.6
3-layer MLP	97.5	84.8
RAT-SPN (Peharz et al. 2018)	98.1	89.5
SVM with RBF Kernel	98.5	87.8
5-Layer MLP	99.3	89.8
Logistic Circuit (binary)	97.4	87.6
Logistic Circuit (real-valued)	99.4	91.3
CNN with 3 conv layers	99.1	90.7
Resnet (He et al. 2016)	99.5	93.6

Significantly Smaller in Size

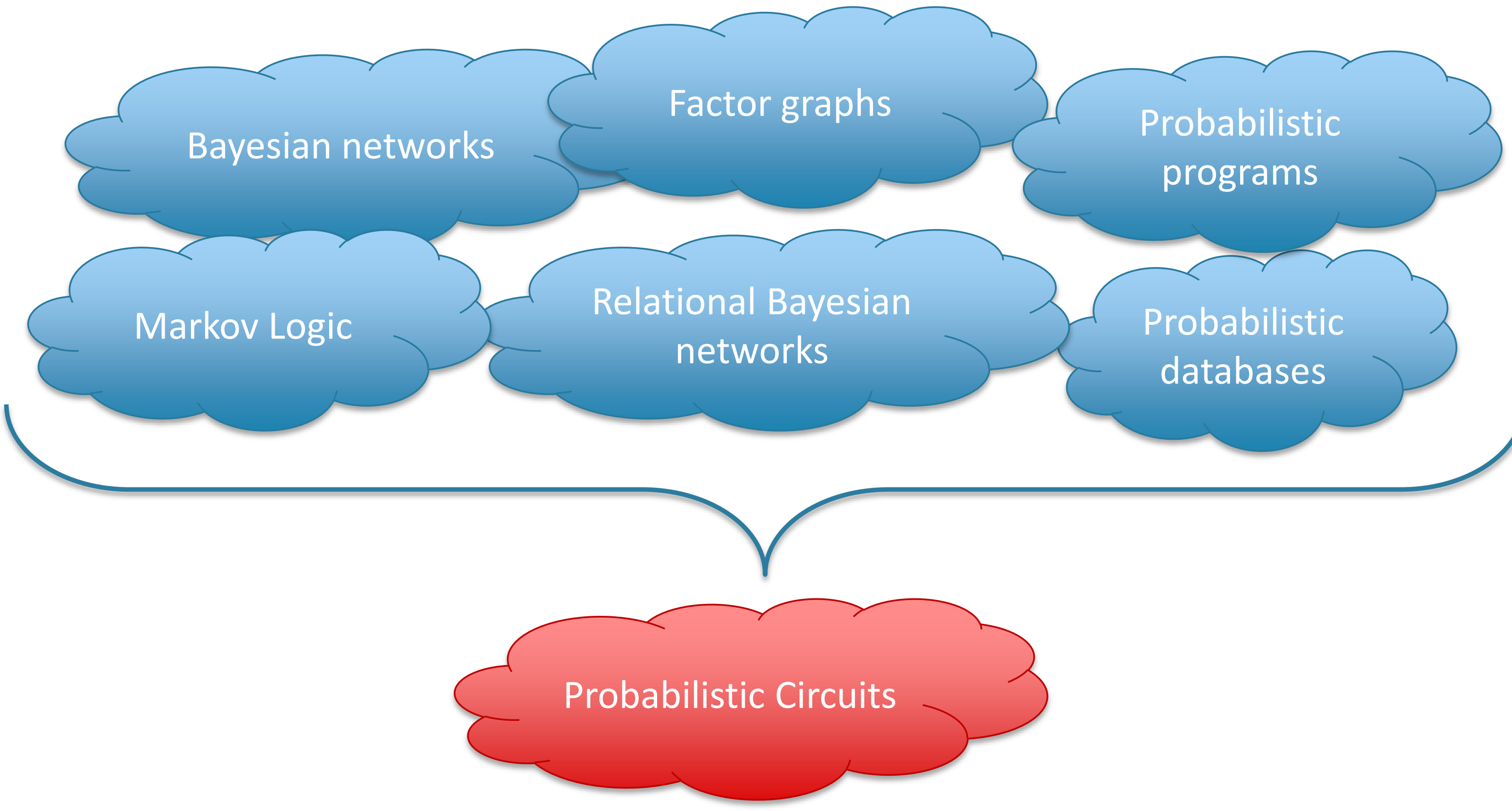
Number of Parameters	Mnist	Fashion
Baseline: Logistic Regression	<1K	<1K
Baseline: Kernel Logistic Regression	1,521 K	3,930K
Logistic Circuit (real-valued)	182K	467K
Logistic Circuit (binary)	268K	614K
3-layer MLP	1,411K	1,411K
RAT-SPN (Peharz et al. 2018)	8,500K	650K
CNN with 3 conv layers	2,196K	2,196K
5-Layer MLP	2,411K	2,411K
Resnet (He et al. 2016)	4,838K	4,838K

Better Data Efficiency

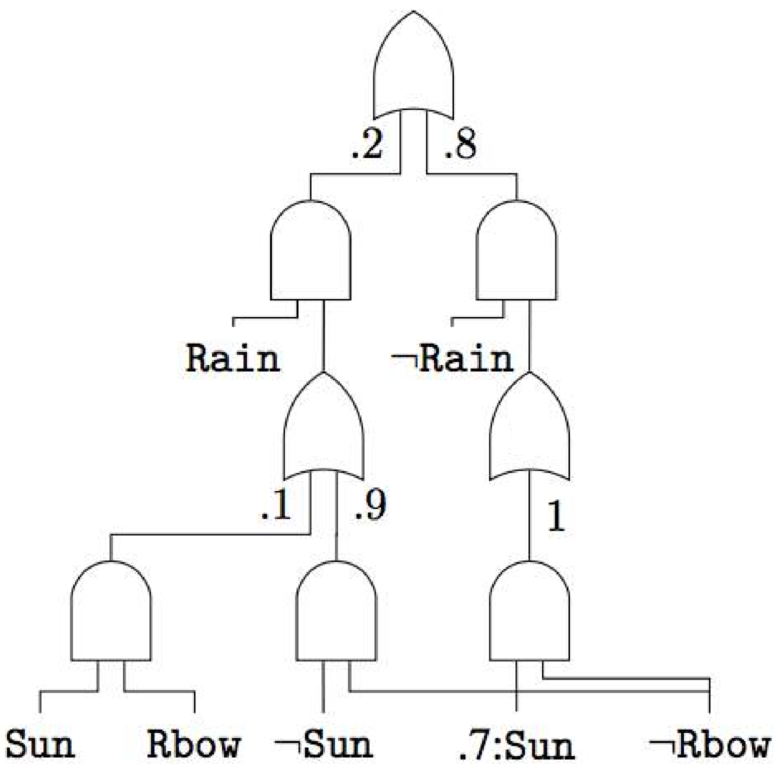
Accuracy % with % of Training Data	MNIST			Fashion		
	100%	10%	2%	100%	10%	2%
5-layer MLP	99.3	**98.2**	94.3	89.8	86.5	80.9
CNN with 3 Conv Layers	99.1	98.1	95.3	90.7	87.6	83.8
Logistic Circuit (Binary)	97.4	96.9	94.1	87.6	86.7	83.2
Logistic Circuit (Real-Valued)	**99.4**	97.6	**96.1**	**91.3**	**87.8**	**86.0**

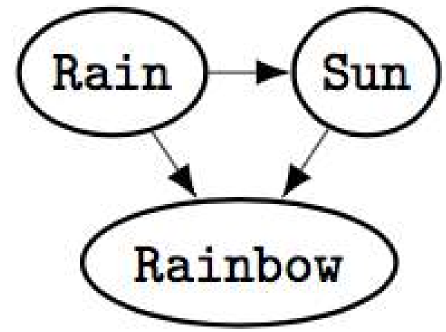
Reasoning with Probabilistic Circuits

Compilation target for probabilistic reasoning



Compilation for Prob. Inference





$$\Pr(\texttt{Rain}) = 0.2,$$

$$\Pr(\texttt{Sun} \mid \texttt{Rain}) = \begin{cases} 0.1 \text{ if } \texttt{Rain} \\ 0.7 \text{ if } \neg\texttt{Rain} \end{cases}$$

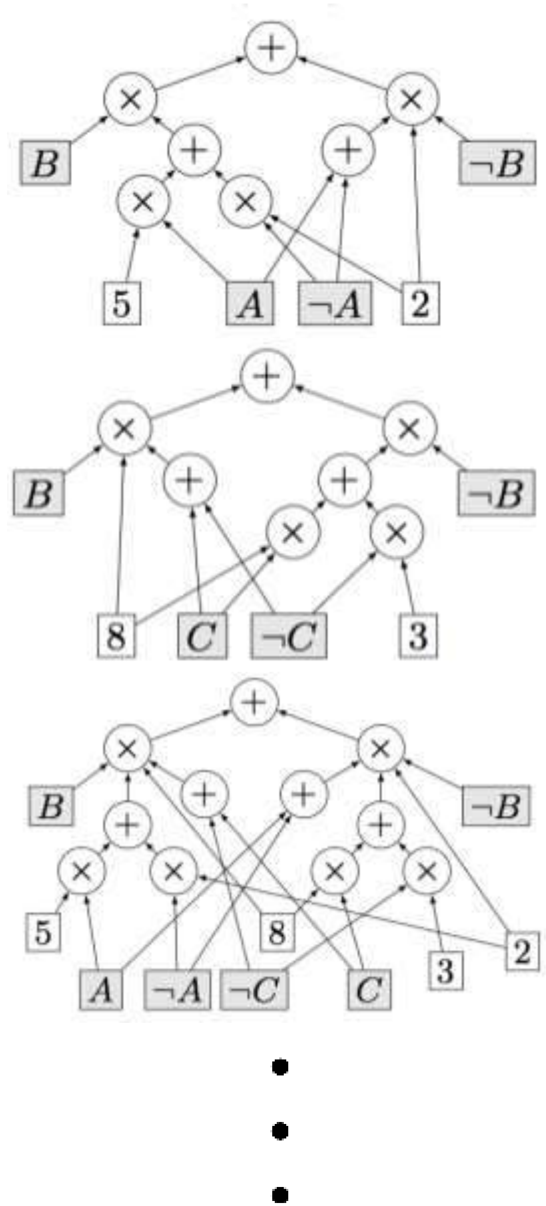
$$\Pr(\texttt{Rbow} \mid \texttt{R}, \texttt{S}) = \begin{cases} 1 \text{ if } \texttt{Rain} \wedge \texttt{Sun} \\ 0 \text{ otherwise} \end{cases}$$

Collapsed Compilation

To sample a circuit:

1. Compile bottom up until you reach the size limit
2. Pick a variable you want to sample
3. Sample it according to its marginal distribution in the current circuit
4. Condition on the sampled value
5. (Repeat)

Asymptotically unbiased importance sampler ☺



Circuits +
importance weights
approximate any query

Experiments

Table 2: Hellinger distances across methods with internal treewidth and size bounds

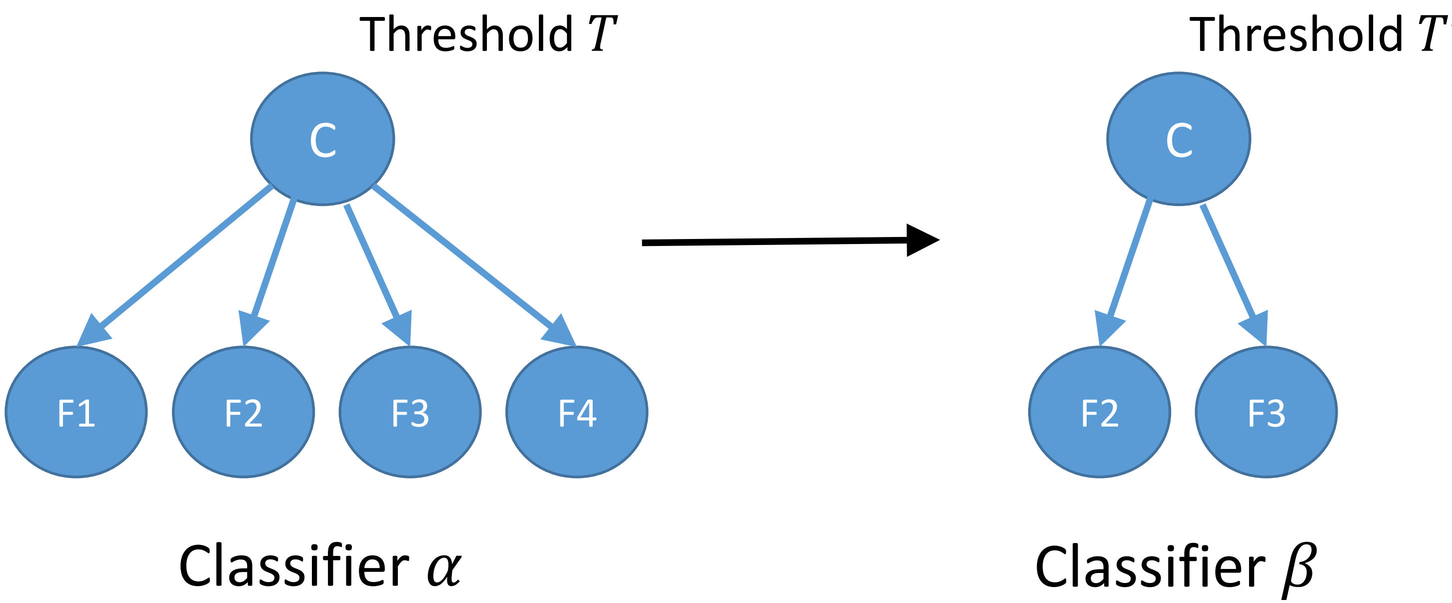
Method	50-20	75-26	DBN	Grids	Segment	linkage	frust
EDBP-100k	2.19e−3	3.17e−5	6.39e−1	1.24e−3	1.63e−6	6.54e−8	4.73e−3
EDBP-1m	7.40e−7	2.21e−4	6.39e−1	1.98e−7	1.93e−7	5.98e−8	4.73e−3
SS-10	2.51e−2	2.22e−3	6.37e−1	3.10e−1	3.11e−7	4.93e−2	1.05e−2
SS-12	6.96e−3	1.02e−3	6.27e−1	2.48e−1	3.11e−7	1.10e−3	5.27e−4
SS-15	9.09e−6	1.09e−4	(Exact)	8.74e−4	3.11e−7	4.06e−6	6.23e−3
FD	9.77e−6	1.87e−3	1.24e−1	1.98e−4	6.00e−8	5.99e−6	5.96e−6
MinEnt	1.50e−5	3.29e−2	1.83e−2	3.61e−3	3.40e−7	6.16e−5	3.10e−2
RBVar	2.66e−2	4.39e−1	6.27e−3	1.20e−1	3.01e−7	2.02e−2	2.30e−3

Competitive with state-of-the-art approximate inference in graphical models. Outperforms it on several benchmarks!

Reasoning About Classifiers

Classifier Trimming

$$C_T(\mathbf{features}) = \mathbb{I}\left(\Pr(C \,|\, \mathbf{features}) \geq T\right)$$



Trim features while maintaining classification behavior

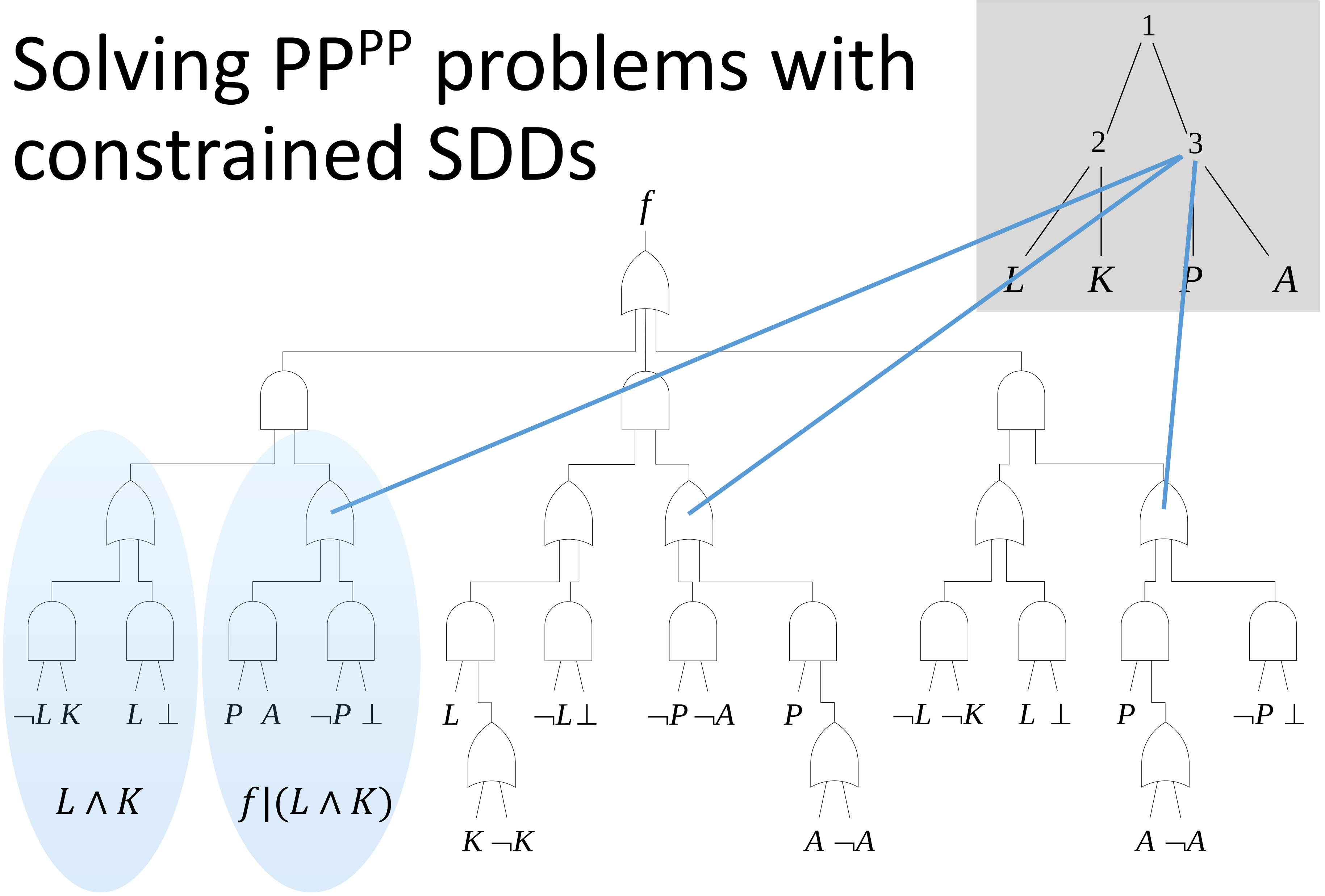
How to measure Similarity?

"Expected Classification Agreement"

$$\mathrm{ECA}(\alpha, \beta) = \sum_{\mathbf{f}} \mathbb{I}(C_T(\mathbf{f}) = C_{T'}(\mathbf{f}')) \cdot \Pr(\mathbf{f})$$

What is the expected probability that a classifier α will agree with its trimming β?

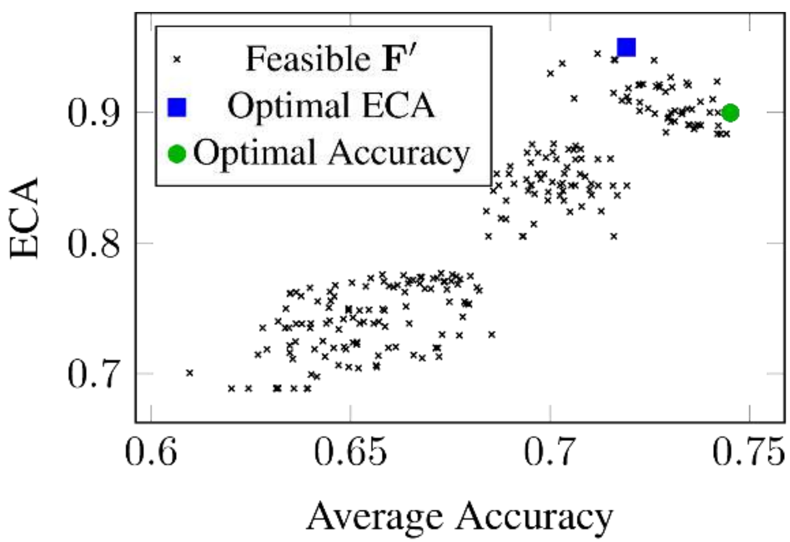
Solving PP^{PP} problems with constrained SDDs

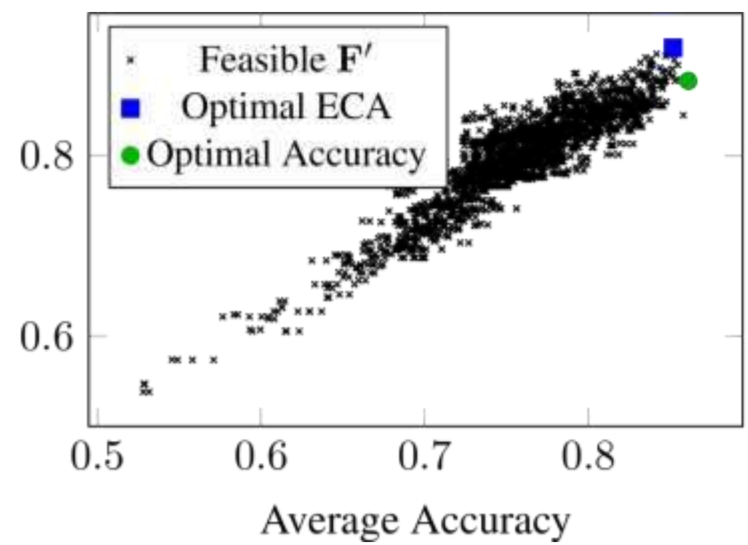


SDD method faster than traditional jointree inference

Network	# nodes	naive	FS-SDD
alarm	37	143.920	19.061
win95pts	76	23.581	14.732
tcc4e	98	48.508	2.384
emdec6g	168	28.072	3.688
diagnose	203	105.660	6.667

Classification agreement and accuracy

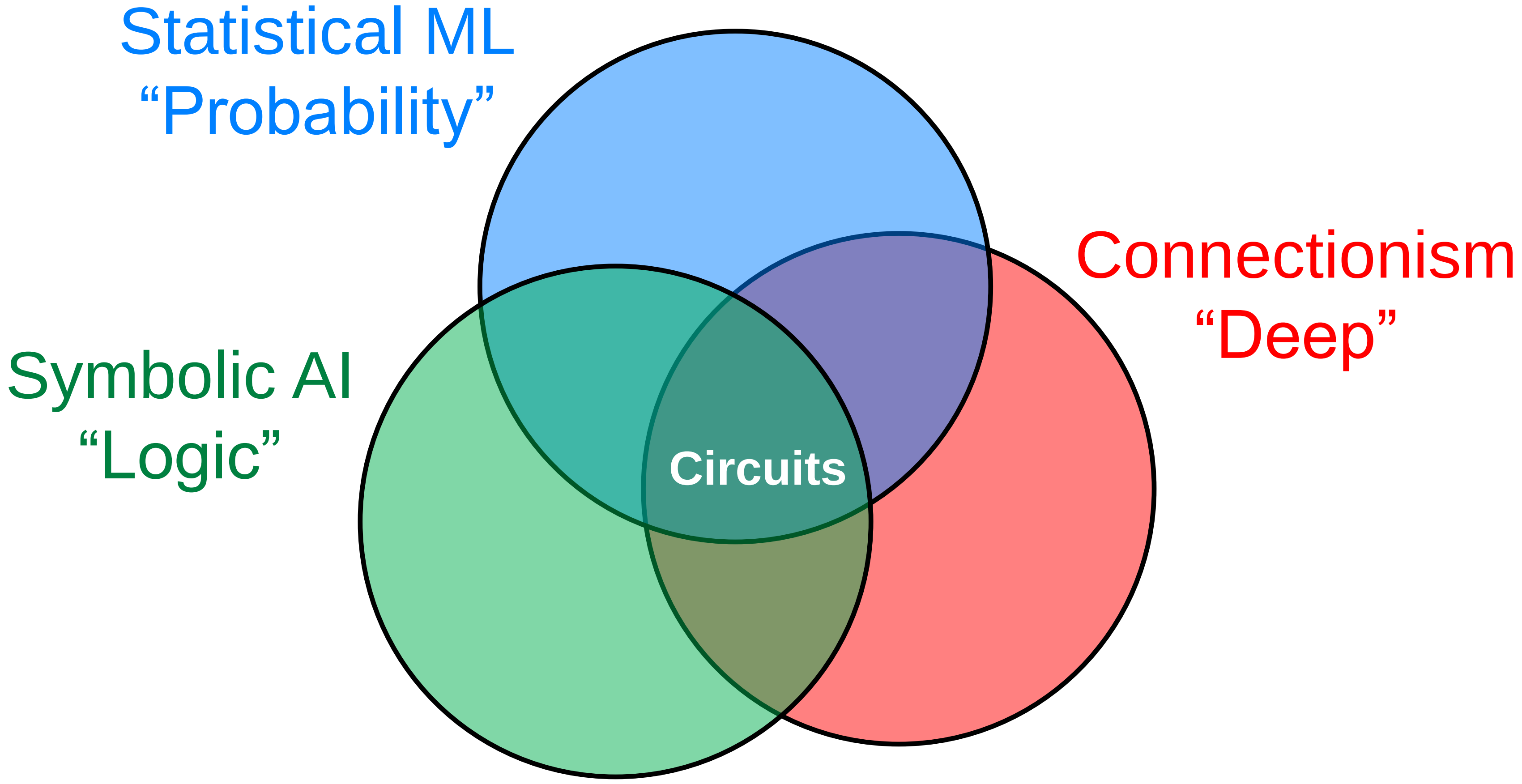




Higher agreement tends to get higher accuracy

Additional dimension for feature selection

Conclusions



Questions?

PSDD with 15,000 nodes

References

- Doga Kisa, Guy Van den Broeck, Arthur Choi and Adnan Darwiche. **Probabilistic sentential decision diagrams**, *In Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 2014.
- Arthur Choi, Guy Van den Broeck and Adnan Darwiche. **Tractable Learning for Structured Probability Spaces: A Case Study in Learning Preference Distributions**, *In Proceedings of 24th International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.
- Arthur Choi, Guy Van den Broeck and Adnan Darwiche. **Probability Distributions over Structured Spaces**, *In Proceedings of the AAAI Spring Symposium on KRR*, 2015.
- Jessa Bekker, Jesse Davis, Arthur Choi, Adnan Darwiche and Guy Van den Broeck. **Tractable Learning for Complex Probability Queries**, *In Advances in Neural Information Processing Systems 28 (NIPS)*, 2015.
- YooJung Choi, Adnan Darwiche and Guy Van den Broeck. **Optimal Feature Selection for Decision Robustness in Bayesian Networks**, *In Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, 2017.

References

- Yitao Liang, Jessa Bekker and Guy Van den Broeck. **Learning the Structure of Probabilistic Sentential Decision Diagrams**, *In Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.
- Yitao Liang and Guy Van den Broeck. **Towards Compact Interpretable Models: Shrinking of Learned Probabilistic Sentential Decision Diagrams**, *In IJCAI 2017 Workshop on Explainable Artificial Intelligence (XAI)*, 2017.
- YooJung Choi and Guy Van den Broeck. **On Robust Trimming of Bayesian Network Classifiers**, *In Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.
- Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang and Guy Van den Broeck. **A Semantic Loss Function for Deep Learning with Symbolic Knowledge**, *In Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.
- Yitao Liang and Guy Van den Broeck. **Learning Logistic Circuits**, *In Proceedings of the UAI 2018 Workshop: Uncertainty in Deep Learning*, 2018.
- Tal Friedman and Guy Van den Broeck. **Approximate Knowledge Compilation by Online Collapsed Importance Sampling**, *In Advances in Neural Information Processing Systems 31 (NIPS)*, 2018.