Circuit Languages as a Synthesis of Learning and Reasoning

Guy Van den Broeck

Simons Symposium on New Directions in Theoretical Machine Learning

May 10, 2019

UCLA

STAR AI
RESEARCH LAB
UCLA

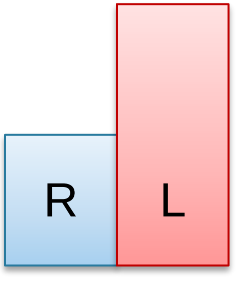
- How can we build on the recent success in supervised learning for perceptual anc related tasks?
- What's next for ML if perception gets solved?
- Is the current set of methods sufficient to take us to the next level of "intelligent" reasoning?
- If not, what is missing, and how can we rectify it?
- What role can classical ideas in Reasoning, Representation Learning, Reinforcement Learning, Interactive Learning, etc. have to play?
- What modes of analyses do we need to even conceptualize the next level of

How are ideas about automated **reasoning** from GOFAI relevant to modern statistical machine learning?

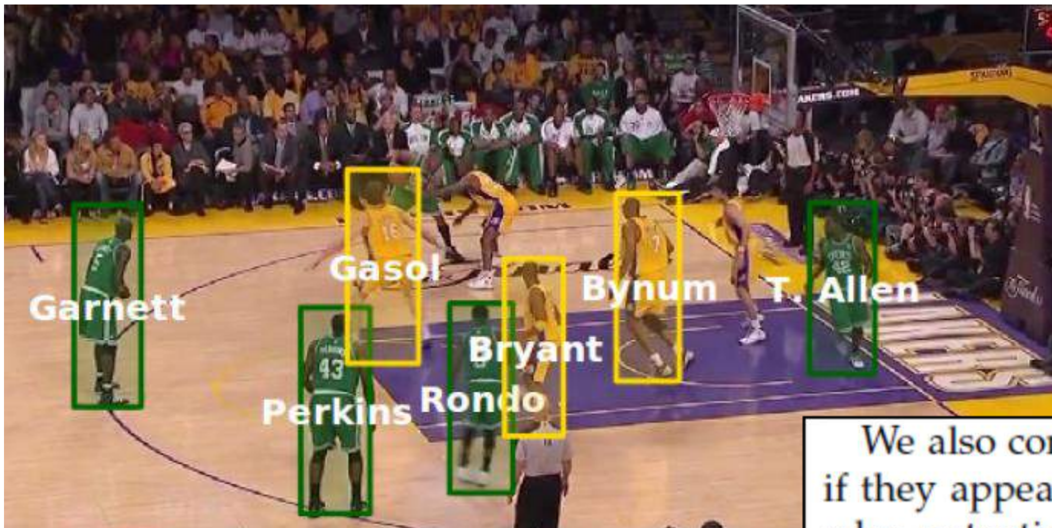
Outline: Reasoning ∩ Learning

1. Deep Learning with Symbolic Knowledge
2. Efficient Reasoning During Learning
3. Probabilistic and Logistic Circuits

Deep Learning with Symbolic Knowledge



Motivation: Vision



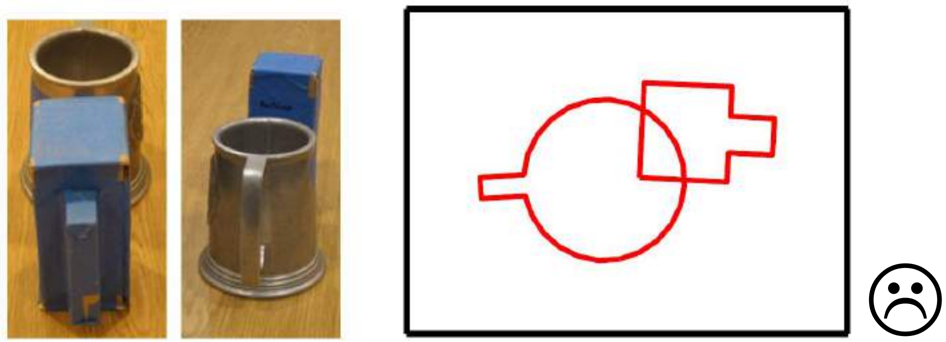
We also connect all pairs of identity nodes $y_{t,i}$ and $y_{t,j}$ if they appear in the same time t. We then introduce an edge potential that enforces mutual exclusion:

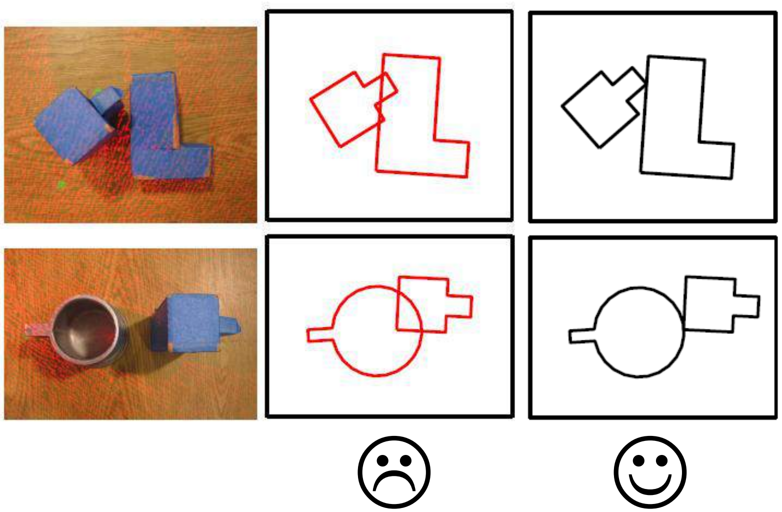
$$\psi_{\text{mutex}}(y_{t,i}, y_{t,j}) = \begin{cases} 1 & \text{if } y_{t,i} \neq y_{t,j} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

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

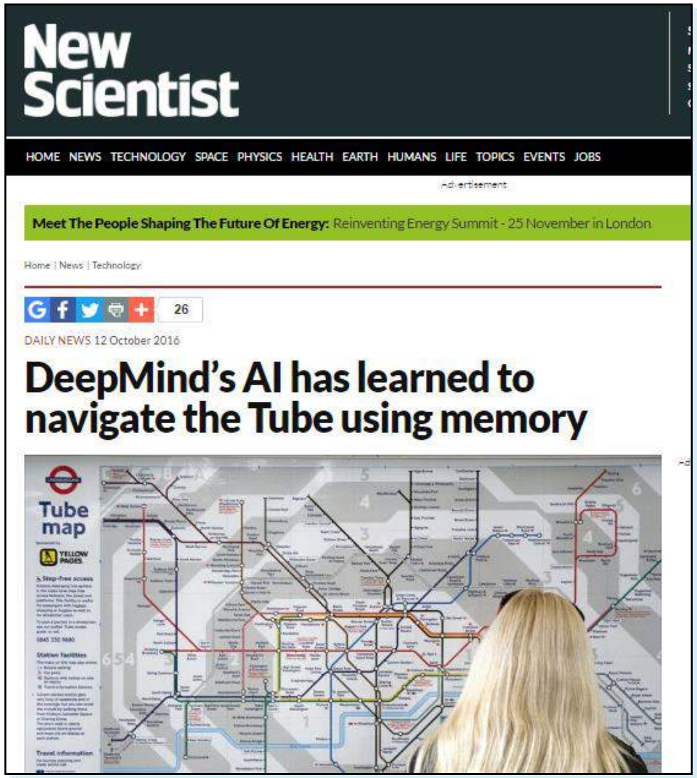
  “If a modifier is kept, its subject is also kept”

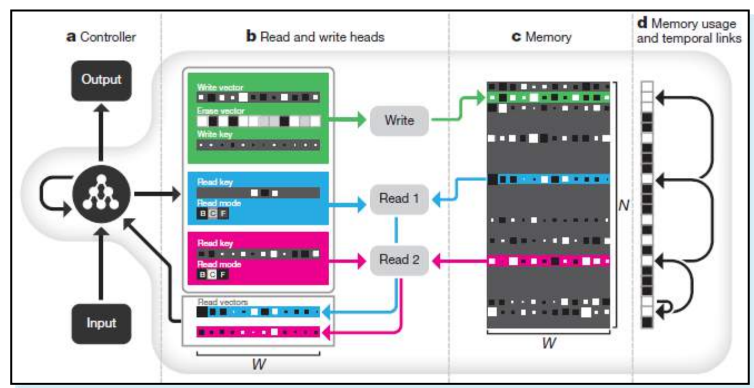
- NELL ontology and rules

… and much more!

[Chang, M., Ratinov, L., & Roth, D. (2008). Constraints as prior knowledge],
[Ganchev, K., Gillenwater, J., & Taskar, B. (2010). Posterior regularization for structured latent variable models]
… and many many more!

Motivation: Deep Learning



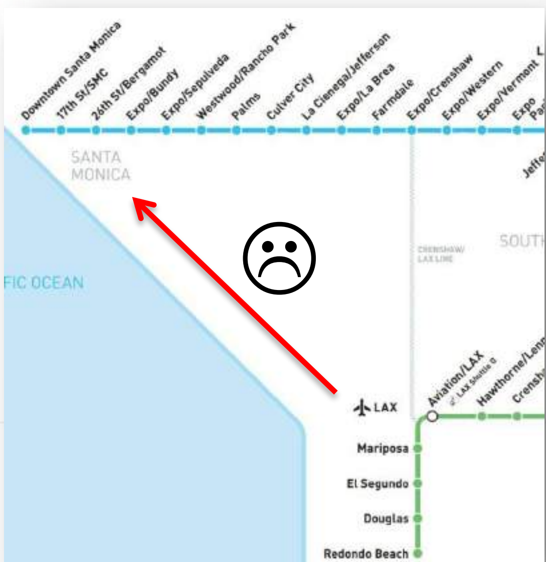


[Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwińska, A., et al.. (2016). Hybrid computing using a neural network with dynamic external memory. *Nature*, *538*(7626), 471-476.]

Motivation: Deep Learning

DeepMind's latest technique uses external memory to **solve tasks that require logic and reasoning — a step toward more human-like AI.**

… but …



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.]

Learning with Symbolic Knowledge

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)

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

1. Must take at least one of Probability (**P**) or Logic (**L**).
2. Probability (**P**) is a prerequisite for AI (**A**).
3. The prerequisites for KR (**K**) is either AI (**A**) or Logic (**L**).

Learning with Symbolic Knowledge

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)

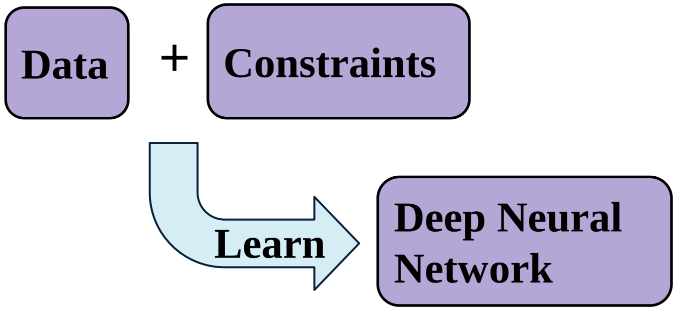
$$P \vee L$$
$$A \Rightarrow P$$
$$K \Rightarrow (P \vee L)$$

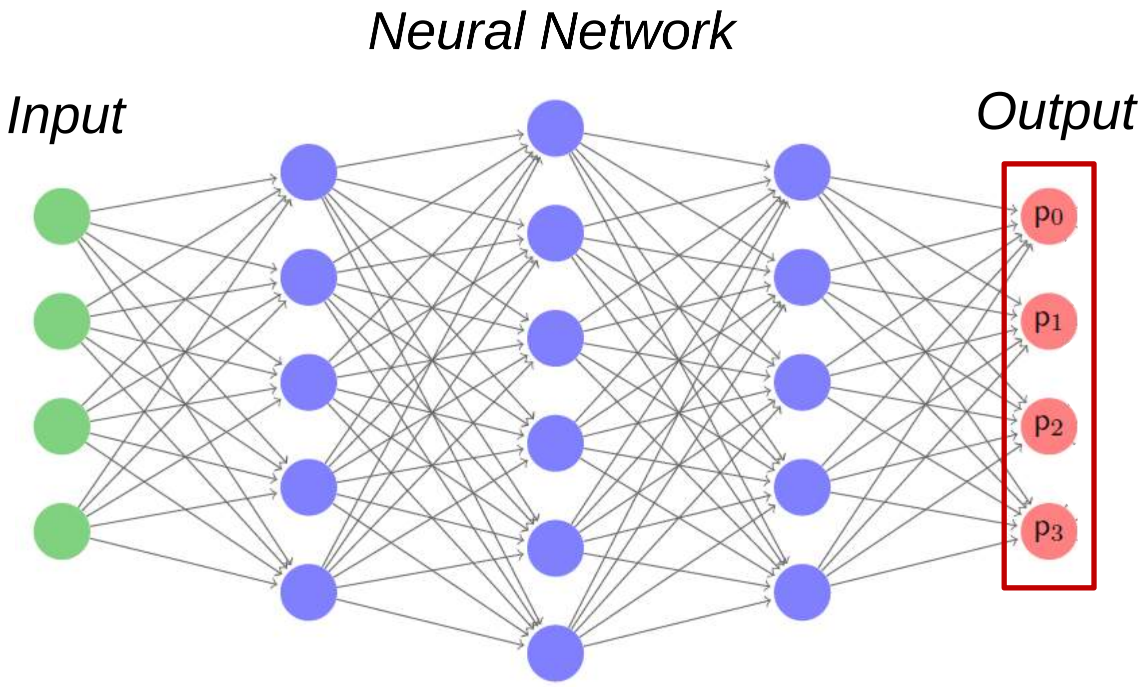
Learn

ML Model

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

Deep Learning with Symbolic Knowledge





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

Semantic Loss

*<u>Q</u>: How close is output **p** to satisfying constraint α?*

*<u>Answer</u>: Semantic loss function L(α,**p**)*

- Axioms, for example:
  - If α fixes the labels, then L(α,**p**) is cross-entropy
  - If α implies β then L(α,**p**) ≥ L(β,**p**) (*α more strict*)

- Implied Properties:
  - If α is equivalent to β then L(α,**p**) = L(β,**p**) ***SEMANTIC*** Loss!
  - If **p** is Boolean and satisfies α then L(α,**p**) = 0

Semantic Loss: Definition

Theorem: 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 state **x** after flipping coins with probabilities **p**

Probability of satisfying α after flipping coins with probabilities **p**

Simple 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:

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

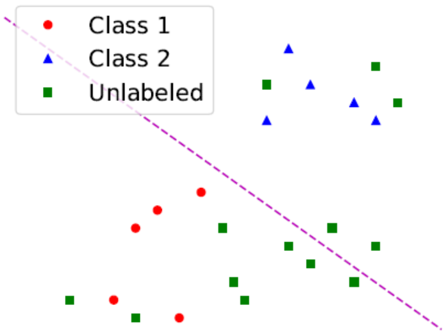
Only $x_i = 1$ after flipping coins

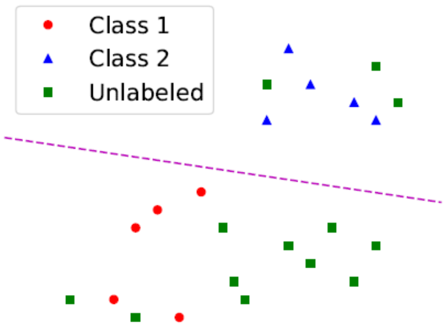
Exactly one true x after flipping coins

Semi-Supervised Learning

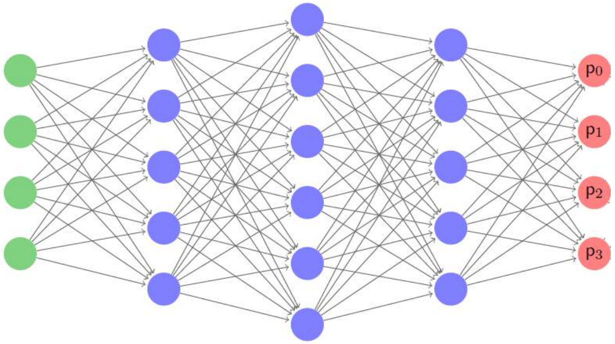
- Intuition: Unlabeled data must have some label

  Cf. entropy minimization, manifold learning





- Minimize exactly-one semantic loss on unlabeled data



Train with

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

Experimental Evaluation

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)	
Baseline: MLP with Entropy Regularizer	96.27 (±0.64)	98.32 (±0.34)	99.37 (±0.12)
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

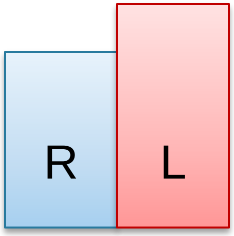
Accuracy % with # of used labels	100	500	1000	ALL
Ladder Net (Rasmus et al., 2015)	81.46 (±0.64)	85.18 (±0.27)	86.48 (±0.15)	90.46
Baseline: MLP, Gaussian Noise	69.45 (±2.03)	78.12 (±1.41)	80.94 (±0.84)	89.87
MLP with Semantic Loss	**86.74** (±0.71)	**89.49** (±0.24)	**89.67** (±0.09)	89.81

Outperforms SoA!

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

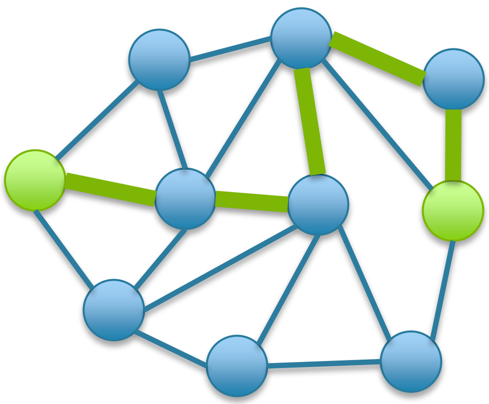
Efficient Reasoning During Learning



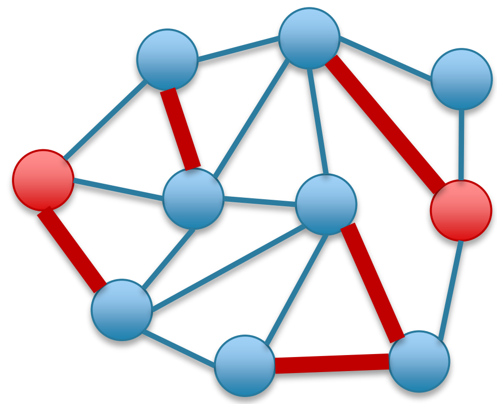
But what about *real* constraints?

- Path constraint

cf. Nature paper



vs.



- Example: 4x4 grids

  2^{24} = 184 paths + 16,777,032 non-paths

- Easily encoded as logical constraints ☺

[Nishino et al., Choi et al.]

How to Compute Semantic Loss?

- In general: #P-hard ☹

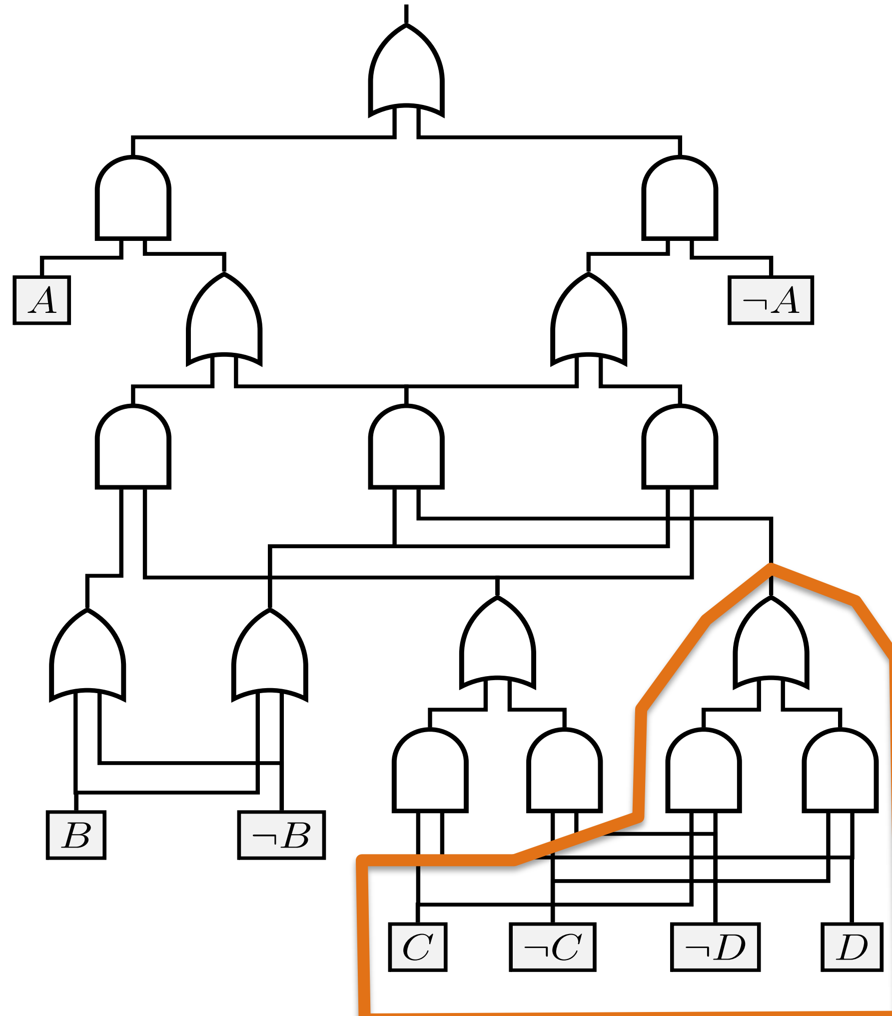
$$\mathrm{L^s}(\alpha, \mathsf{p}) \propto -\log \sum_{\mathbf{x} \models \alpha} \prod_{i:\mathbf{x} \models X_i} \mathsf{p}_i \prod_{i:\mathbf{x} \models \neg X_i} (1 - \mathsf{p}_i)$$

Reasoning Tool: Logical Circuits

Representation of logical sentences:

$(C \land \neg D) \lor (\neg C \land D)$

C XOR D

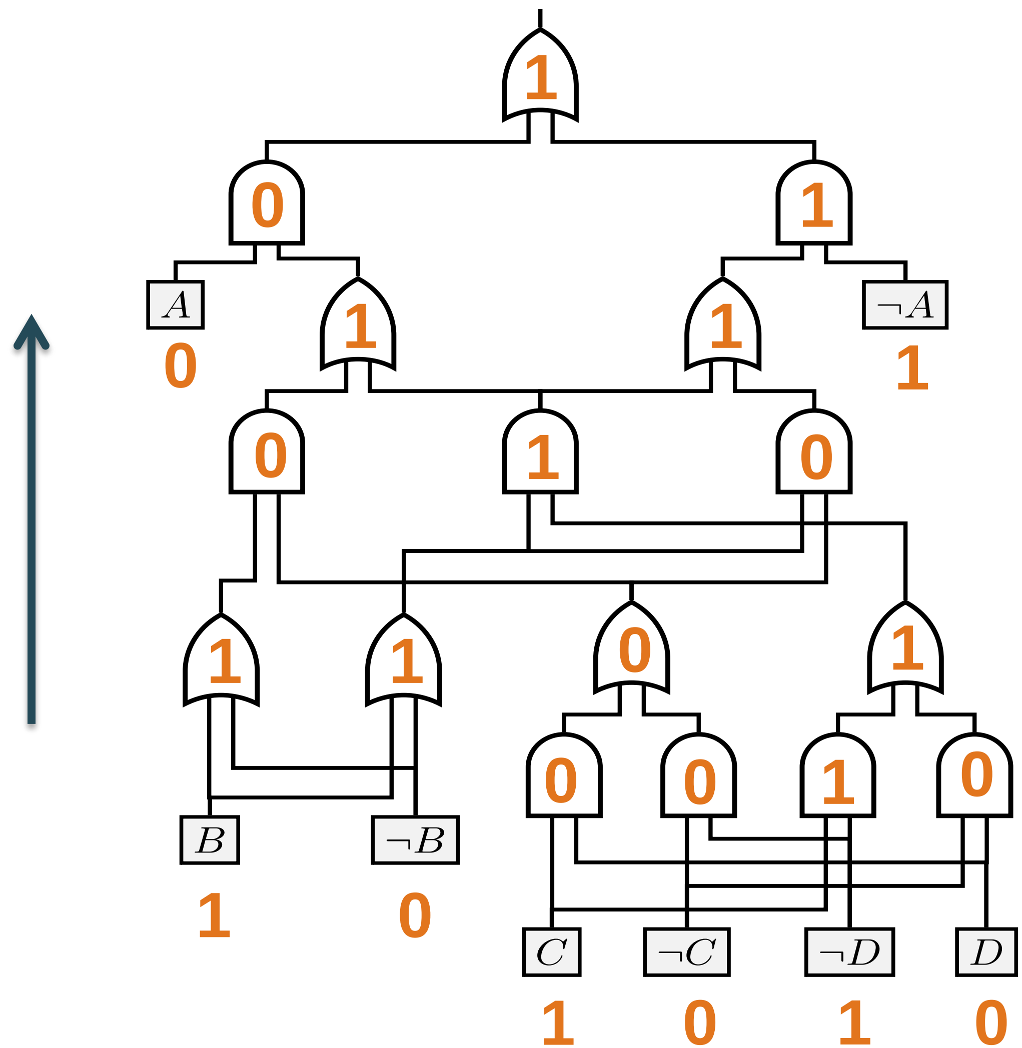


Reasoning Tool: Logical Circuits

Representation of logical sentences:

Input:

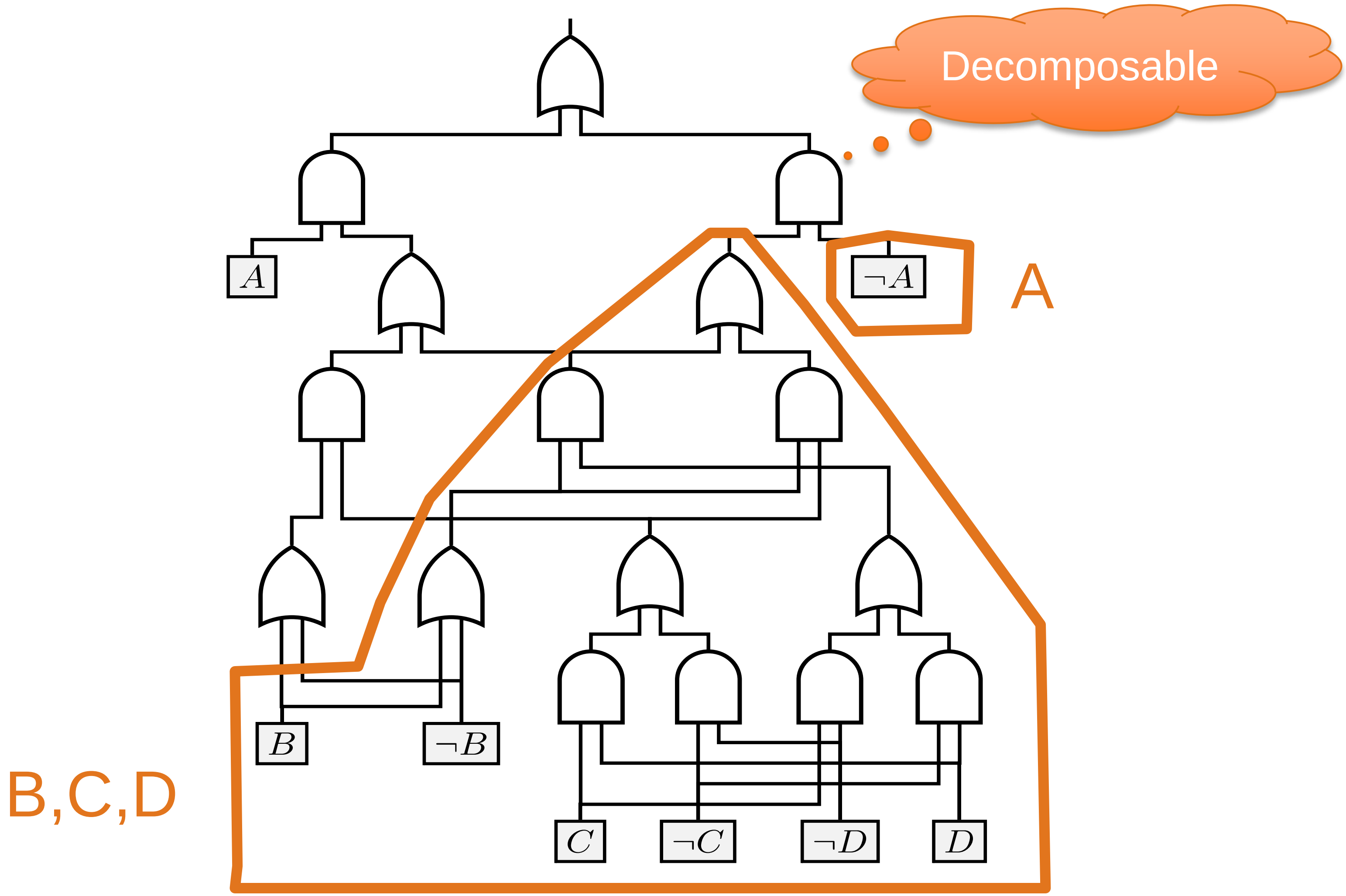
A	B	C	D
0	1	1	0



Tractable for Logical Inference

- Is there a solution? (SAT)
  - SAT($\alpha \lor \beta$) iff SAT(α) or SAT(β)   (*always*)
  - SAT($\alpha \land \beta$) iff ***???***

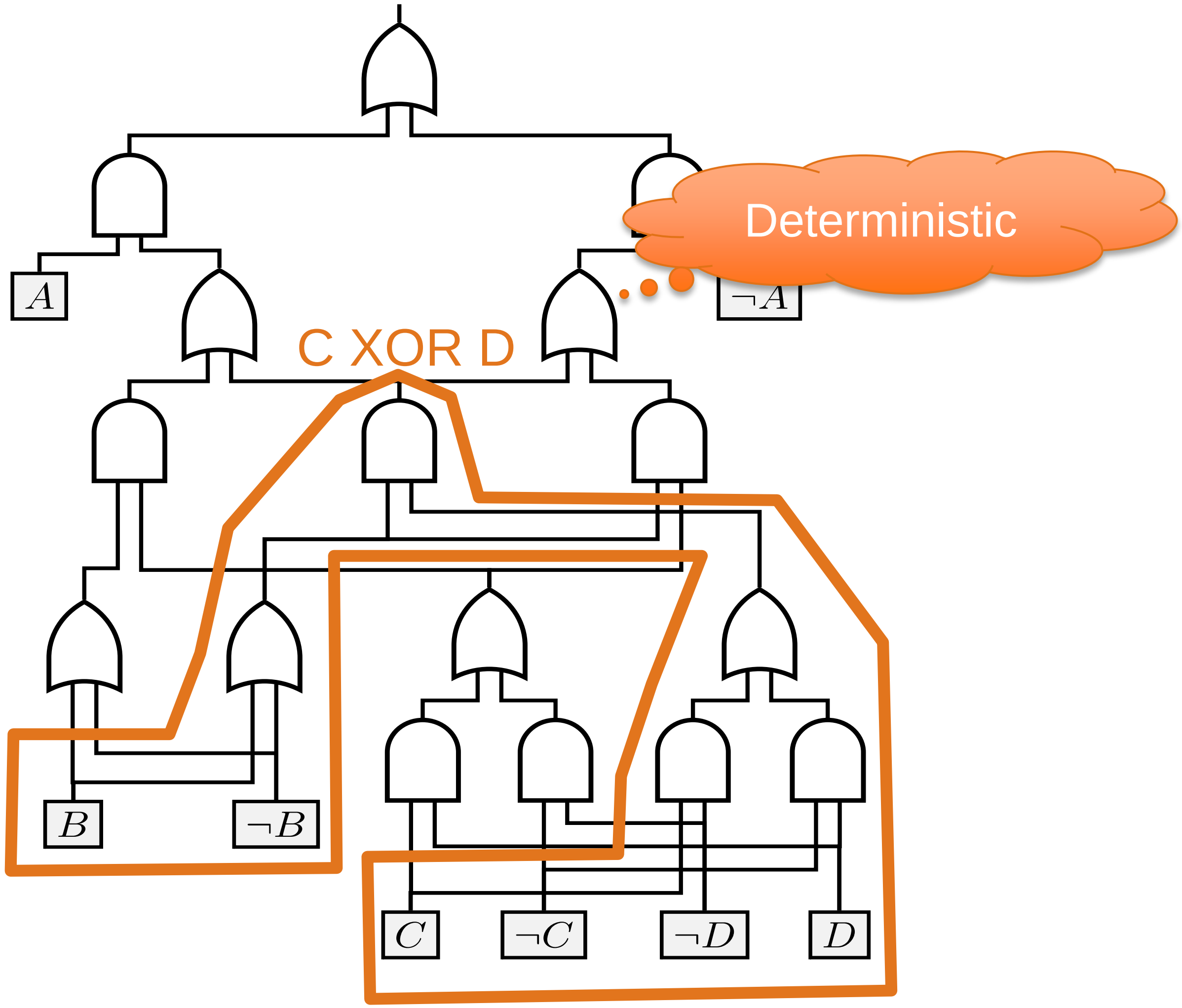
Decomposable Circuits



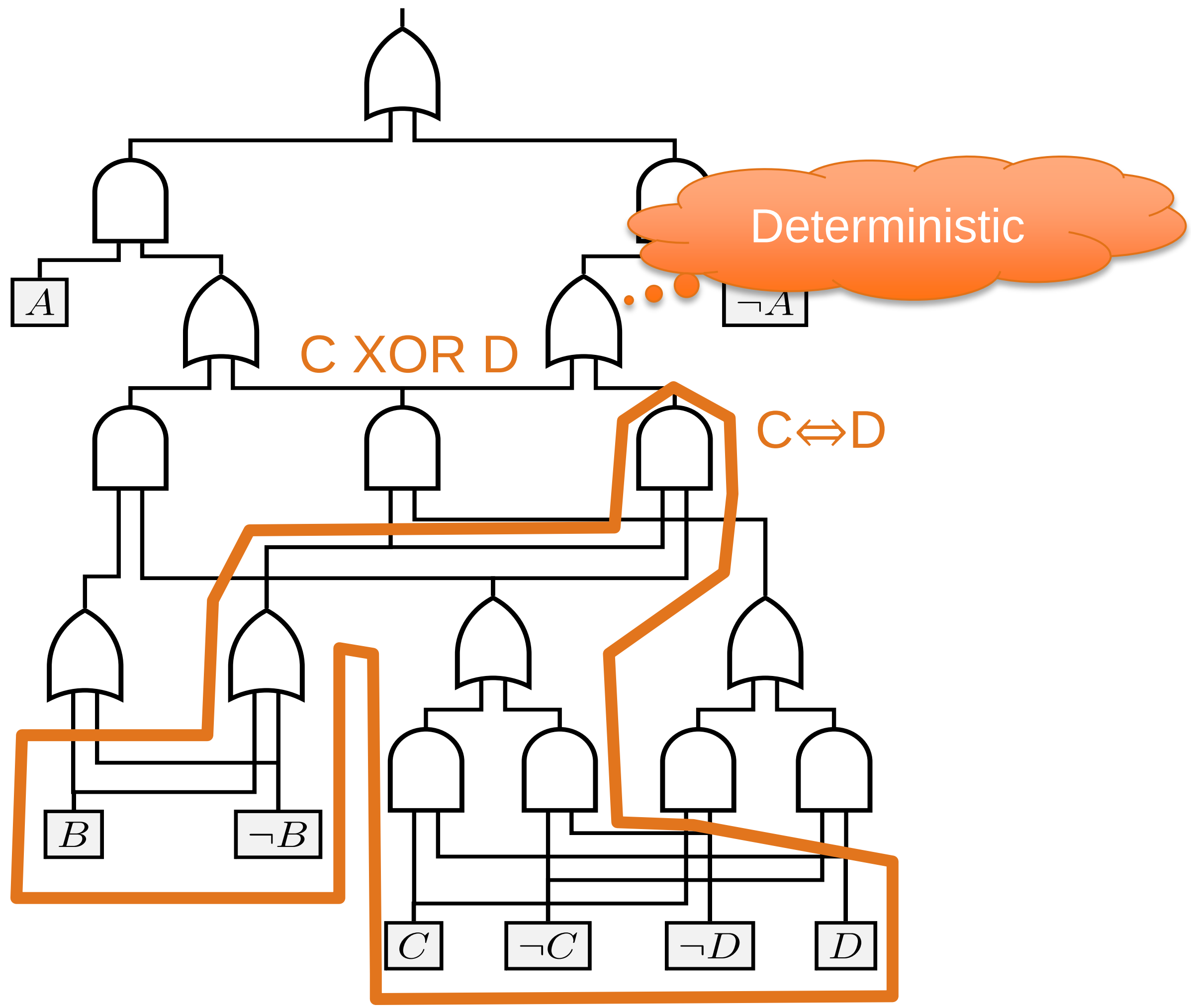
Tractable for Logical Inference

- Is there a solution? (SAT) ✓
  - SAT($\alpha \lor \beta$) iff SAT(α) or SAT(β) (*always*)
  - SAT($\alpha \land \beta$) iff SAT(α) and SAT(β) (*decomposable*)
- How many solutions are there? (#SAT)
- Complexity linear in circuit size ☺

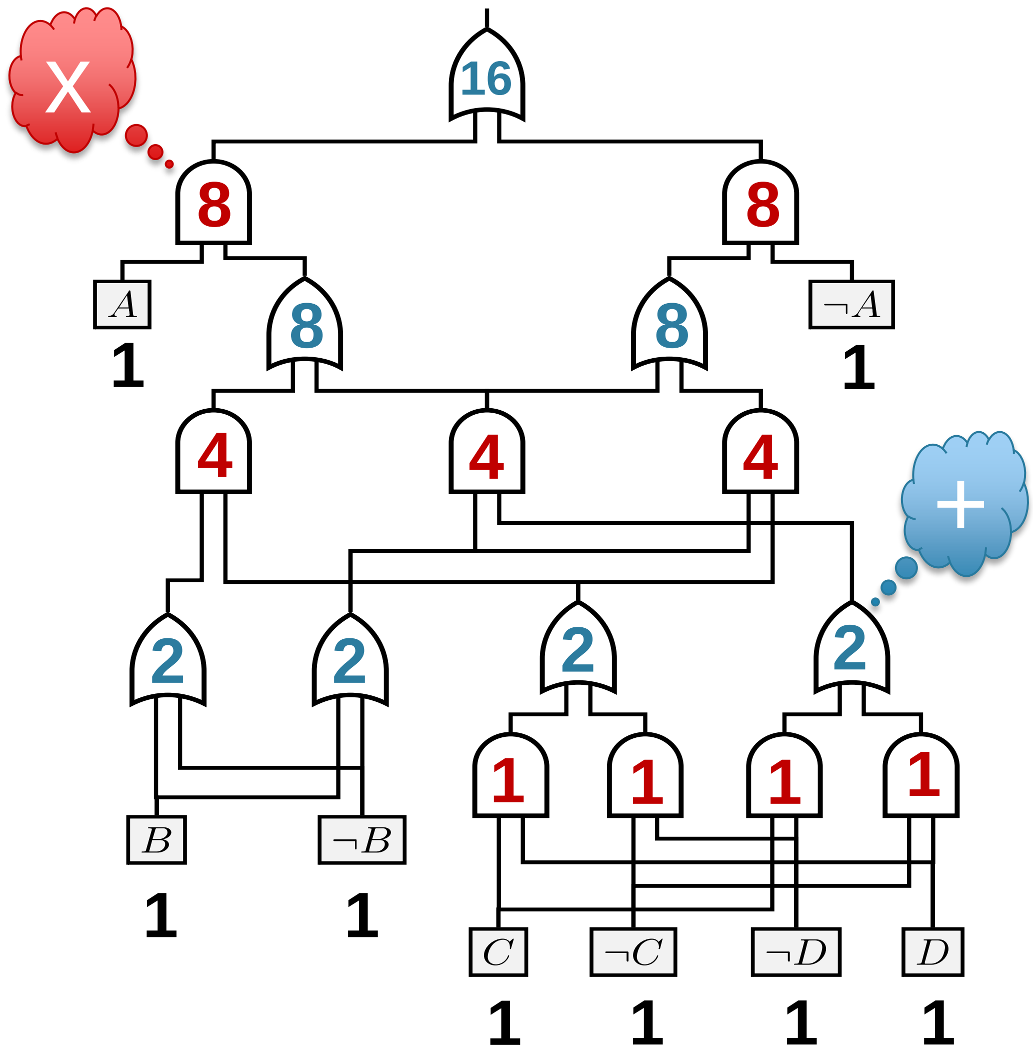
Deterministic Circuits



Deterministic Circuits



How many solutions are there? (#SAT)



Tractable for Logical Inference

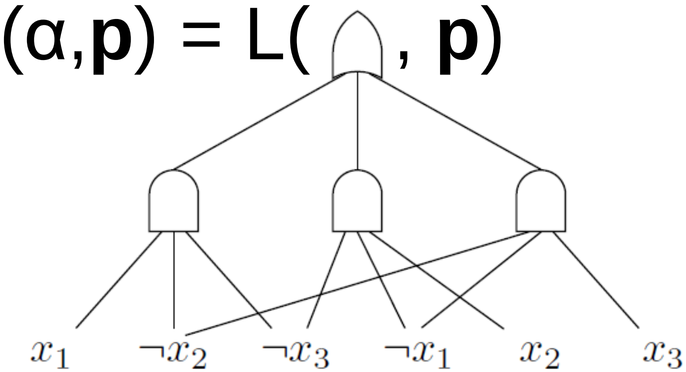
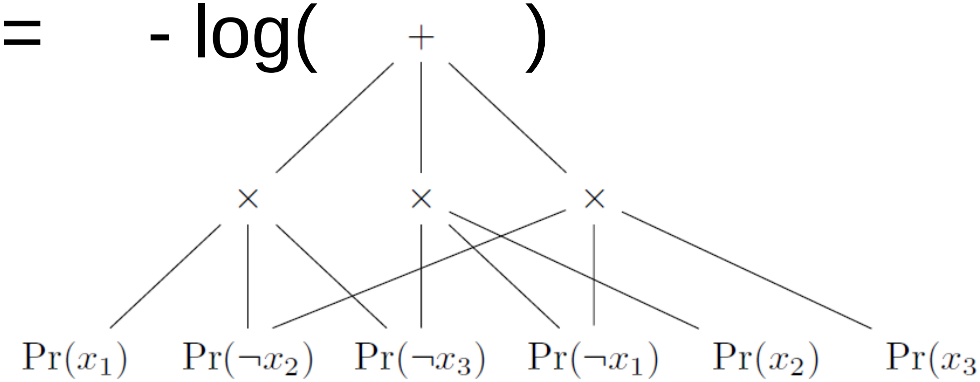
- Is there a solution? (SAT) ✓
- How many solutions are there? (#SAT) ✓
- Conjoin, disjoin, equivalence checking, etc. ✓
- Complexity linear in circuit size ☺

- Compilation into circuit by
  - ↓ exhaustive SAT solver
  - ↑ conjoin/disjoin/negate

[Darwiche and Marquis, JAIR 2002]

How to Compute Semantic Loss?

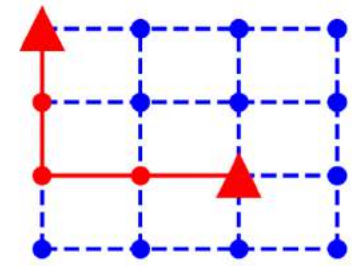
- In general: #P-hard ☹
- With a logical circuit for α: Linear ☺
- Example: exactly-one constraint:

$L(\alpha,\mathbf{p})$ = 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?

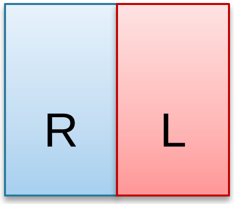
Is output a path?

(same conclusion for predicting sushi preferences, see paper)

Conclusions 1

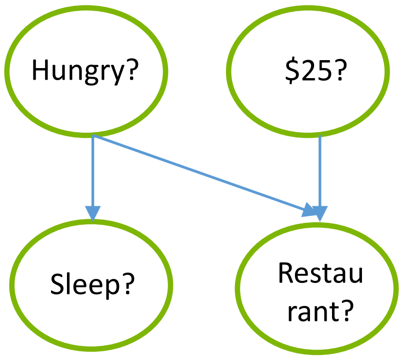
- Knowledge is (hidden) everywhere in ML
- Semantic loss makes logic differentiable
- Performs well semi-supervised
- Requires hard reasoning in general
  - Reasoning can be encapsulated in a circuit
  - No overhead during learning
- Performs well on structured prediction
- A little bit of reasoning goes a long way!

Probabilistic and Logistic Circuits

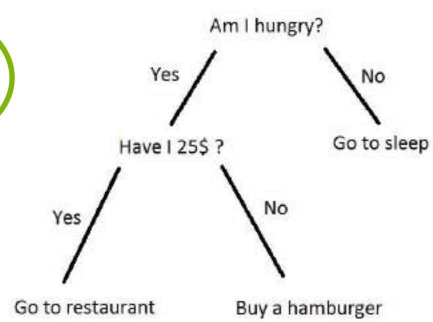


A False Dilemma?

Classical AI Methods

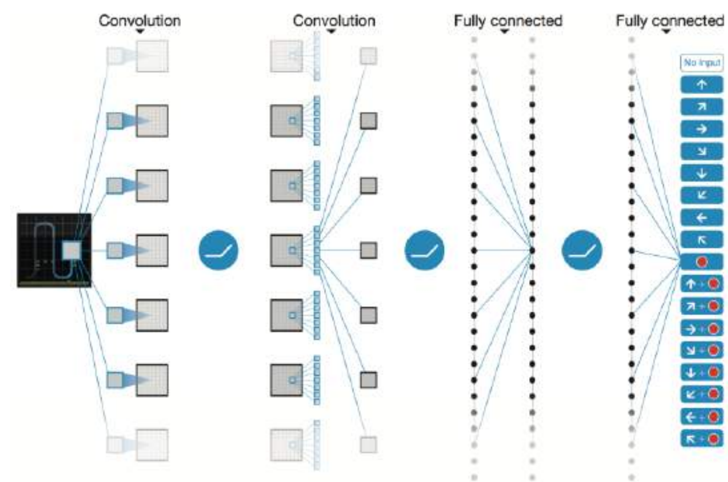


...



Clear Modeling Assumption
Well-understood

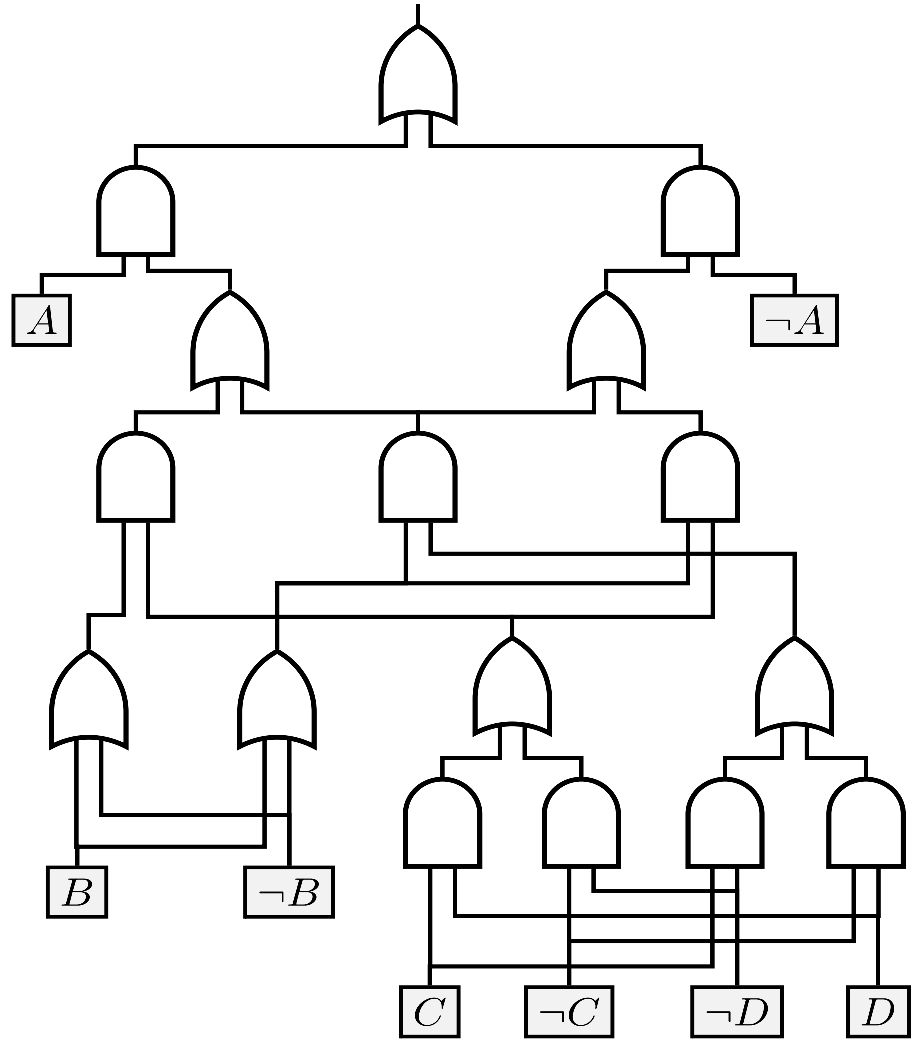
Neural Networks



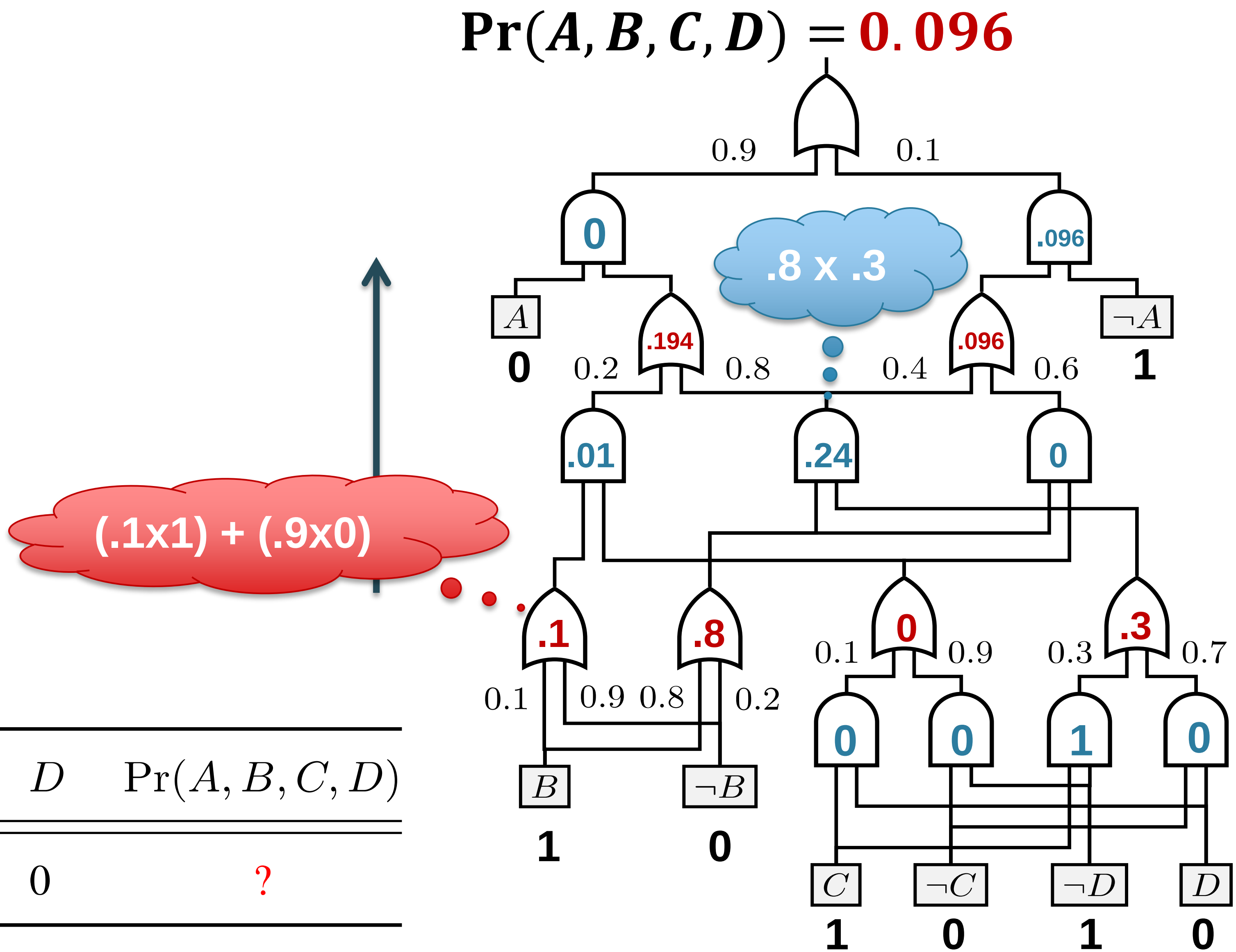
“Black Box”
Empirical performance

Inspiration: Probabilistic Circuits

Can we turn **logic circuits** into a **statistical model**?



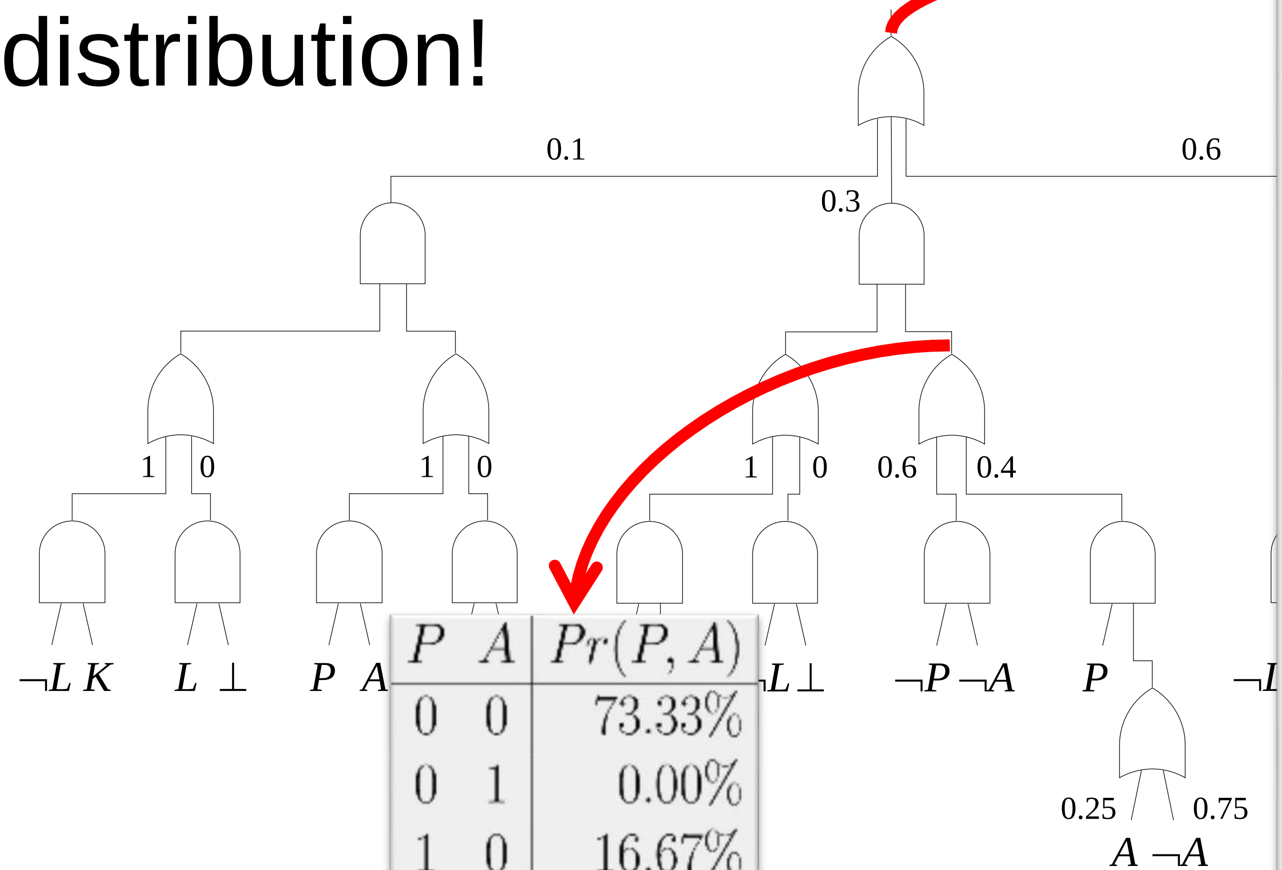
Probabilistic Circuits



Input:

A	B	C	D	$\Pr(A,B,C,D)$
0	1	1	0	?

Each node represents a normalized distribution!

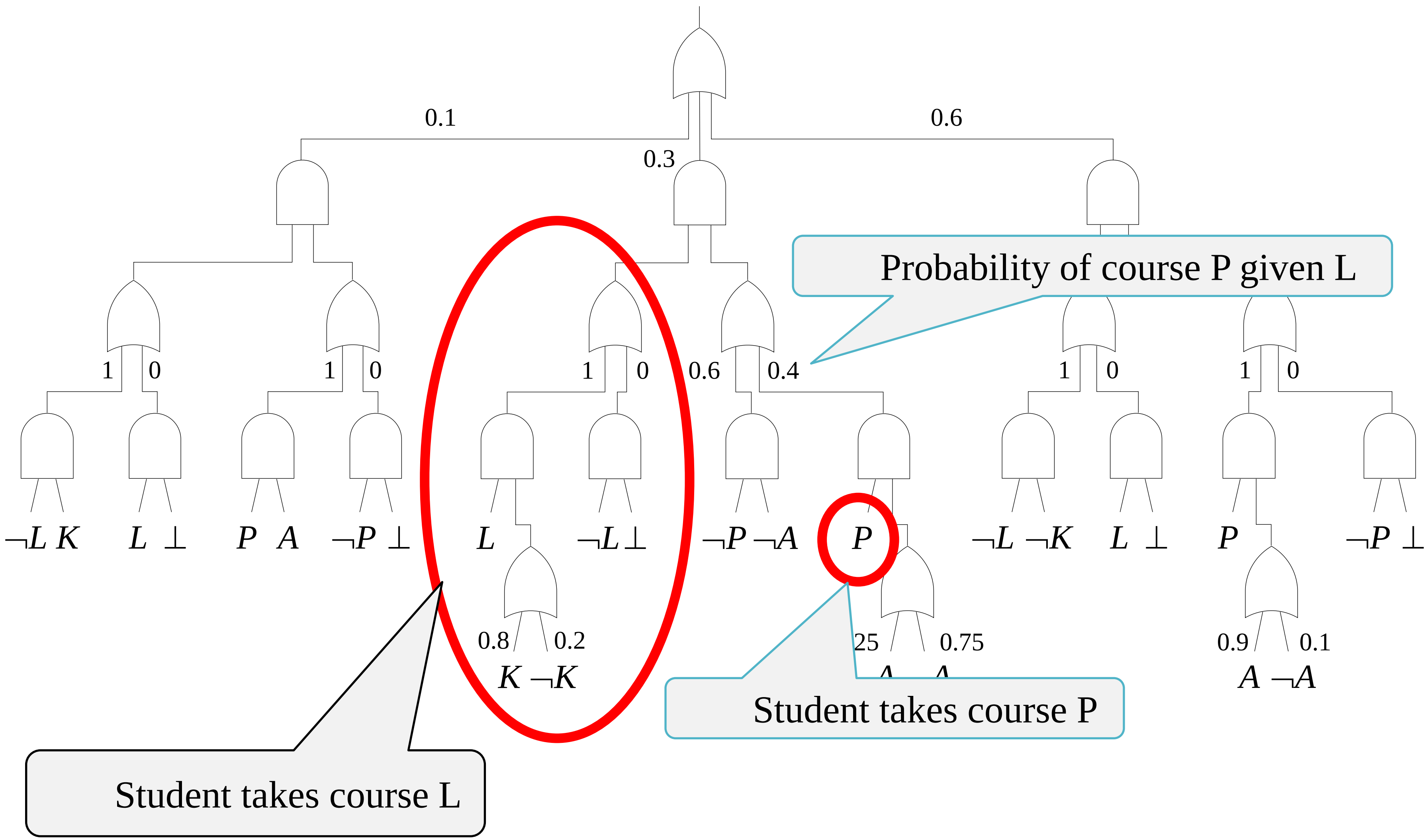


P	A	$Pr(P, A)$
0	0	73.33%
0	1	0.00%
1	0	16.67%
1	1	10.00%

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

Parameters are Interpretable



Properties, Properties, Properties!

- Read conditional independencies from structure
- Interpretable parameters (XAI)
  (conditional probabilities of logical sentences)
- Closed-form parameter learning
- Efficient reasoning

  – **MAP inference**: most-likely assignment to x given y (otherwise NP-hard)
  – Computing **conditional probabilities** Pr(x|y) (otherwise #P-hard)
  – Algorithms linear in circuit size ☺
  – x and y could even be complex logical circuits

Discrete Density Estimation

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	$-14.36^\dagger$

LearnPSDD
state of the art
on 6 datasets!

Q: "Help! I need to learn a discrete probability distribution…"
A: Learn probabilistic circuits!

Strongly outperforms

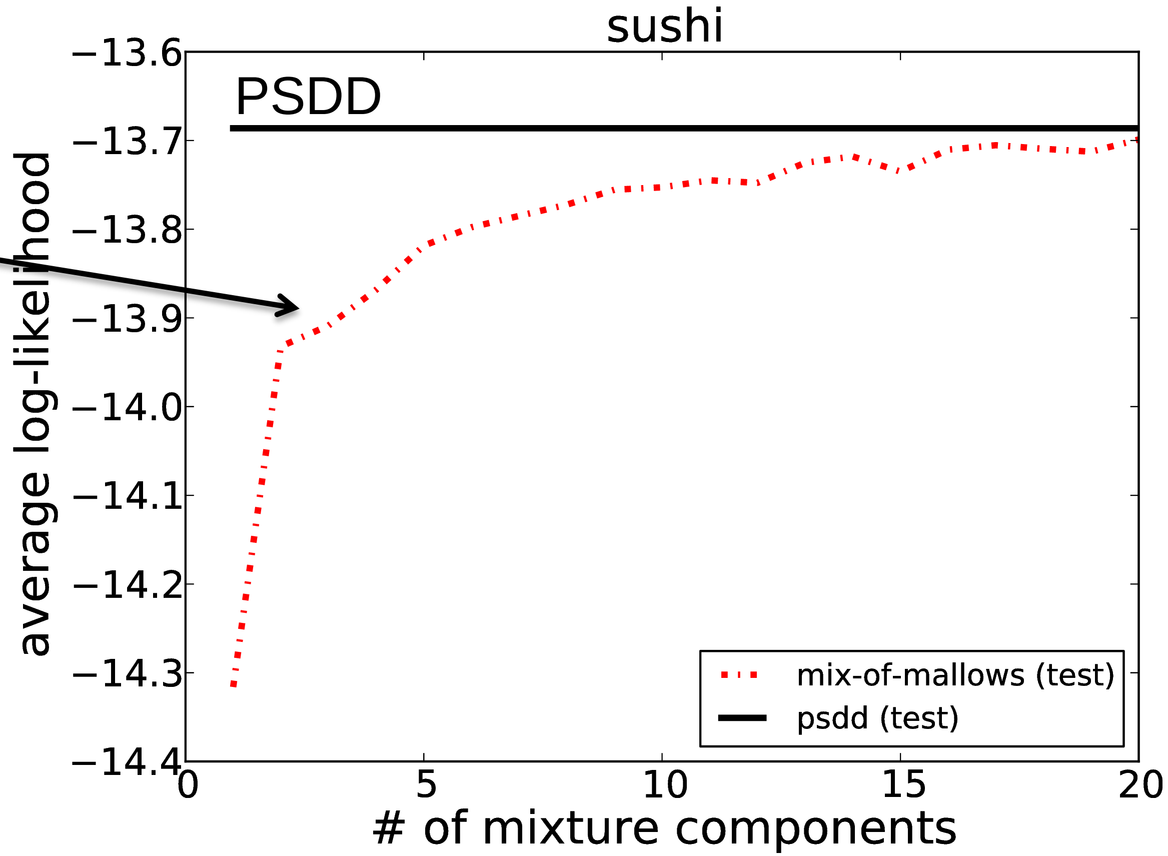
- Bayesian network learners
- Markov network learners

Competitive SPN learner

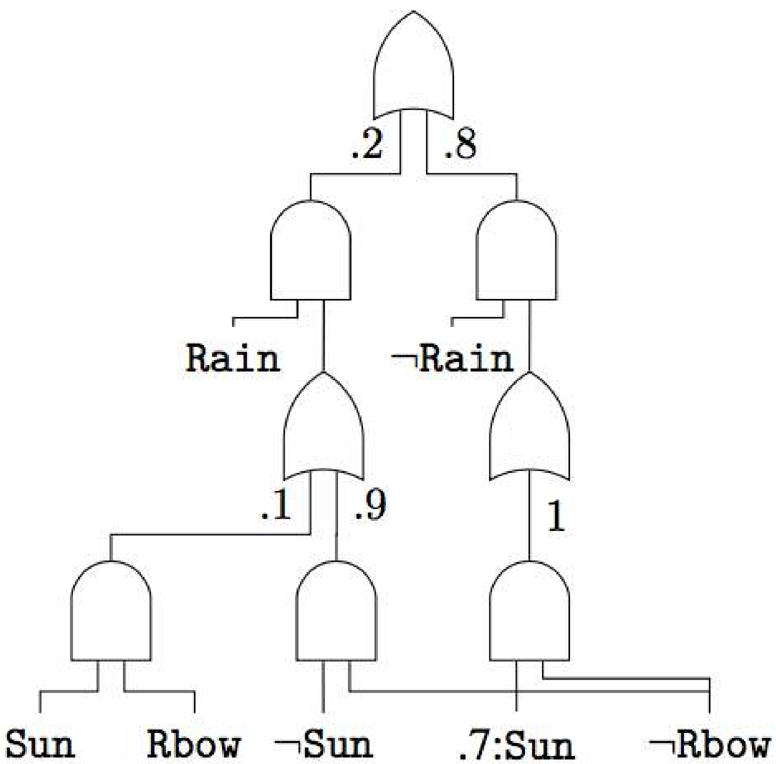
Learning Preference Distributions

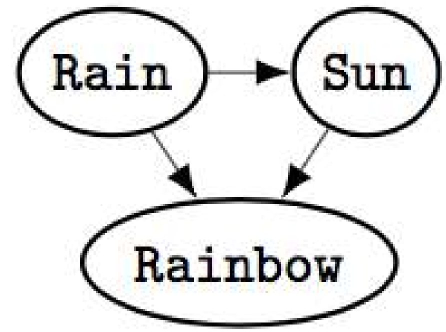
Special-purpose distribution: Mixture-of-Mallows

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



Compilation for Prob. Inference





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

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

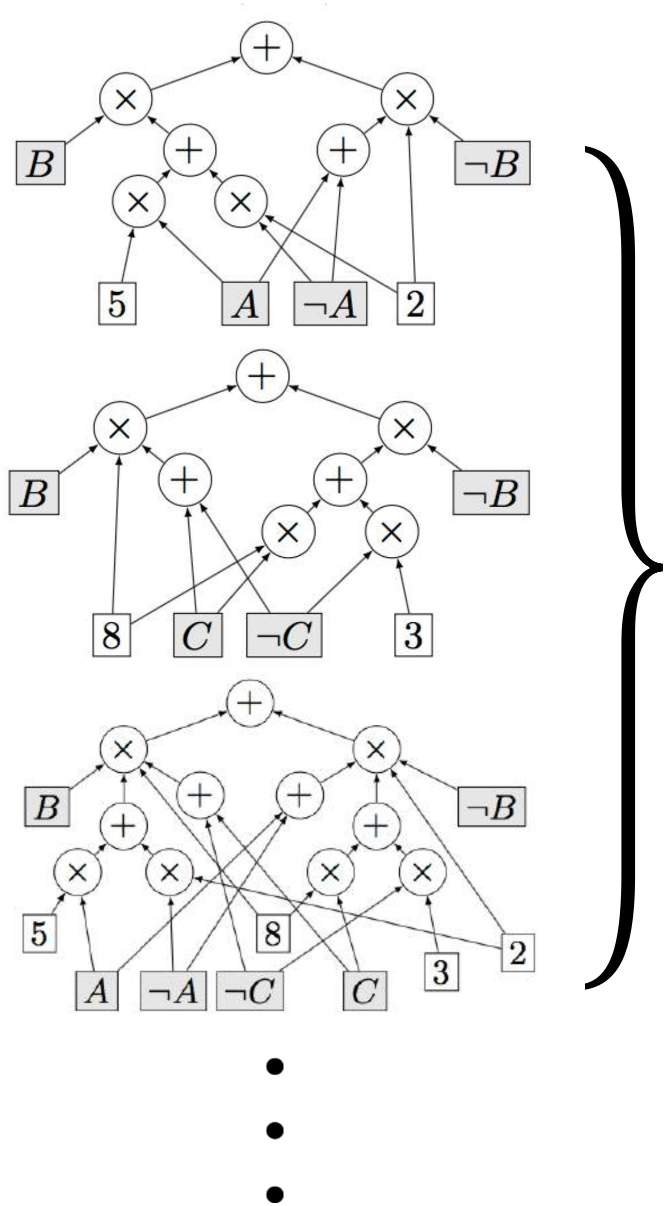
$$\Pr(\texttt{Rbow} \mid \texttt{R}, \texttt{S}) = \begin{cases} 1 \text{ if } \texttt{Rain} \wedge \texttt{Sun} \\ 0 \text{ otherwise} \end{cases}$$

Collapsed Compilation [NeurIPS 2018]

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

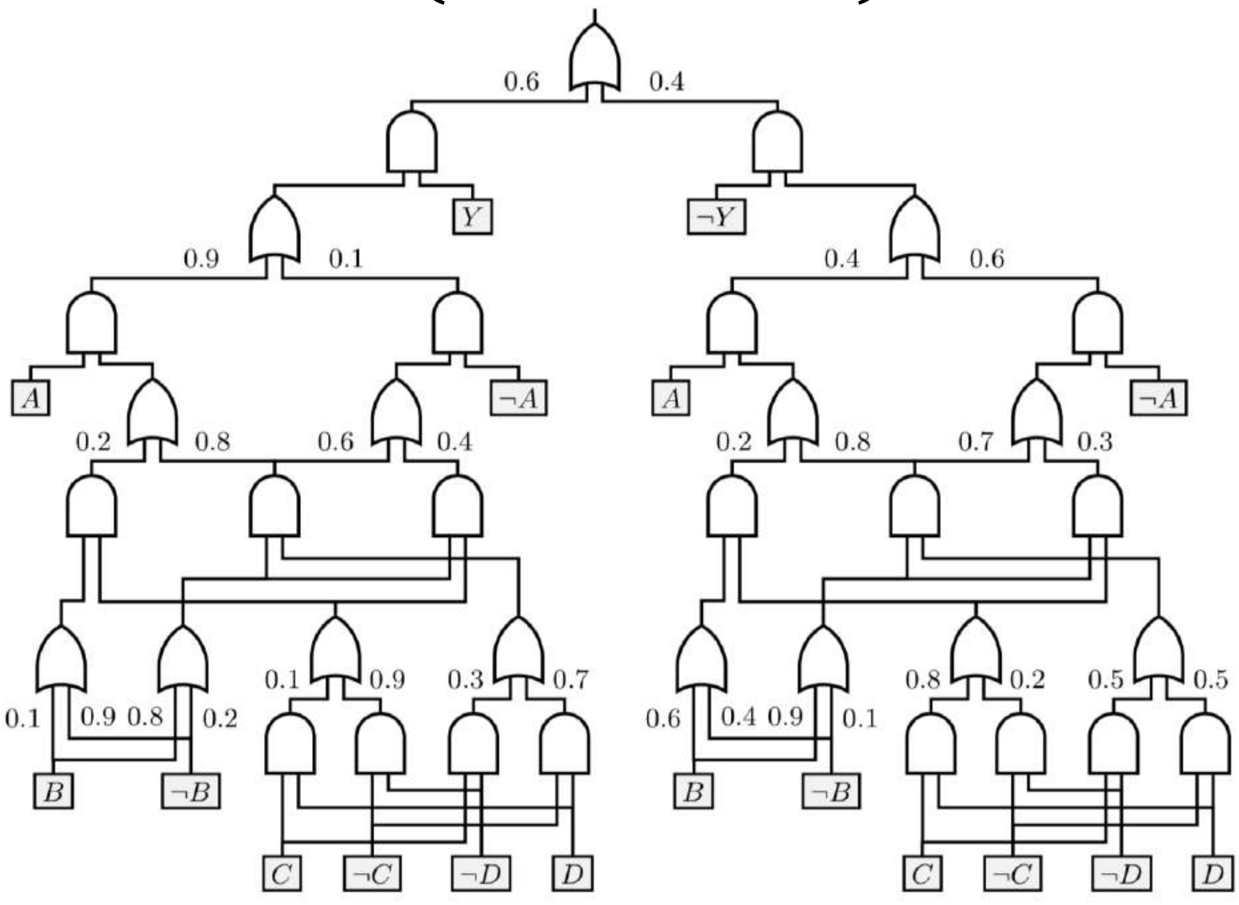
Method	50-20	75-26	DBN	Grids	Segment	linkage	frust
EDBP-100k	2.19e−3	3.17e−5	6.39e−1	1.24e−3	1.63e−6	6.54e−8	4.73e−3
EDBP-1m	7.40e−7	2.21e−4	6.39e−1	1.98e−7	1.93e−7	5.98e−8	4.73e−3
SS-10	2.51e−2	2.22e−3	6.37e−1	3.10e−1	3.11e−7	4.93e−2	1.05e−2
SS-12	6.96e−3	1.02e−3	6.27e−1	2.48e−1	3.11e−7	1.10e−3	5.27e−4
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!

But what if I only want to classify Y?

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

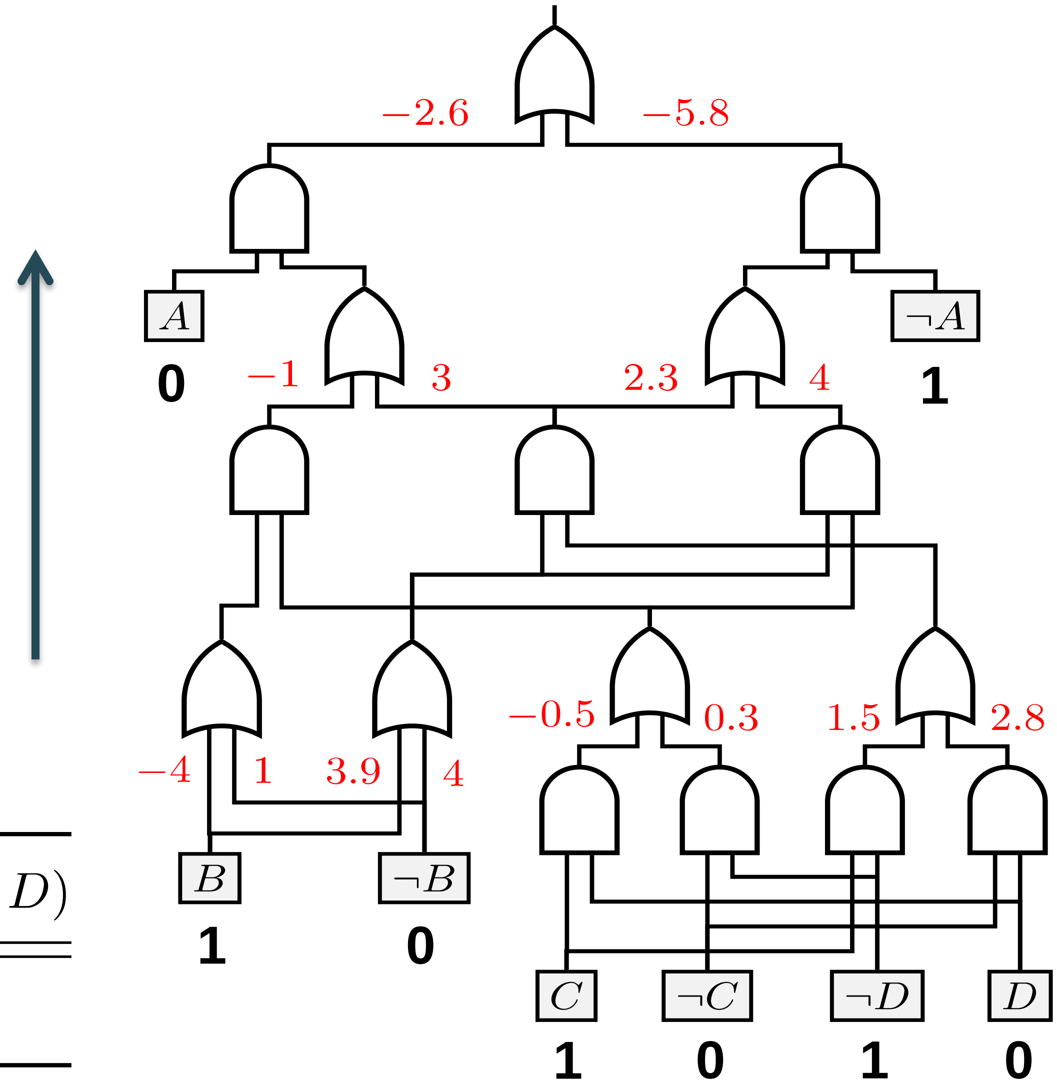
~~$\Pr(Y,A,B,C,D)$~~



Logistic Circuits

$$\Pr(Y = 1 \mid A, B, C, D) = \frac{1}{1 + exp(-1.9)} = 0.869$$

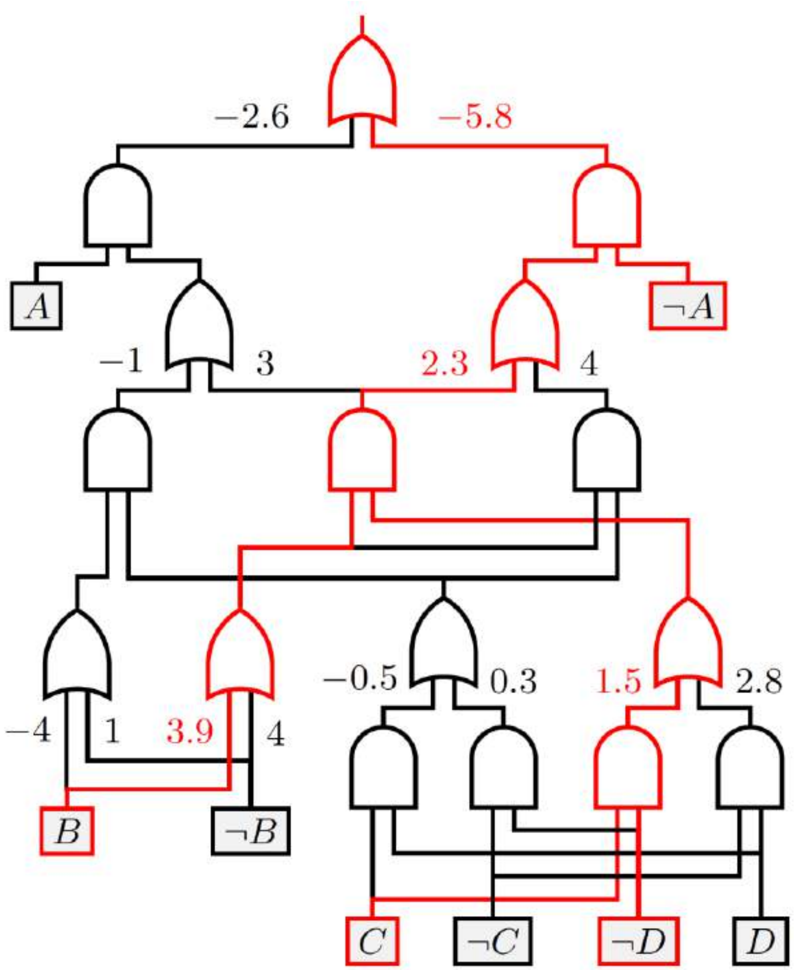
Logistic function on output weight



Input:

A	B	C	D	$\Pr(Y \mid A, B, C, D)$
0	1	1	0	?

Alternative Semantics

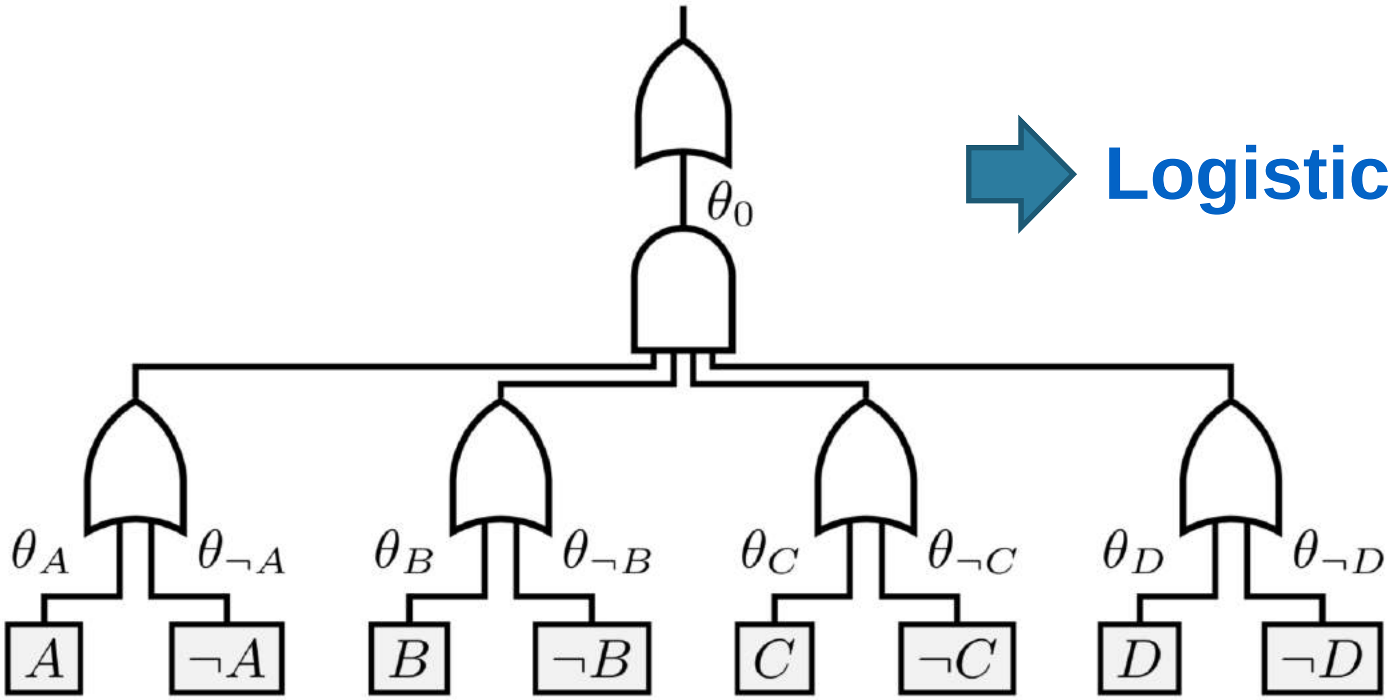


Represents $\Pr(Y \mid A, B, C, D)$

- 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%

Special Case: Logistic Regression



Logistic Regression

$$\Pr(Y = 1 | A, B, C, D) = \frac{1}{1 + \exp(-A * \theta_A - \neg A * \theta_{\neg A} - B * \theta_B - \cdots)}$$

Is this a coincidence?

What about more general circuits?

Parameter Learning

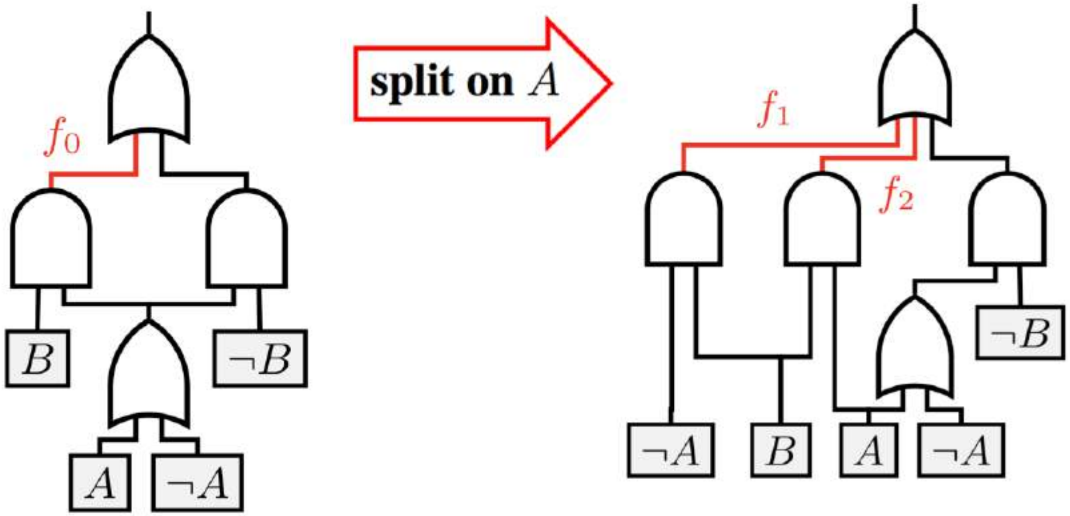
Reduce to logistic regression:

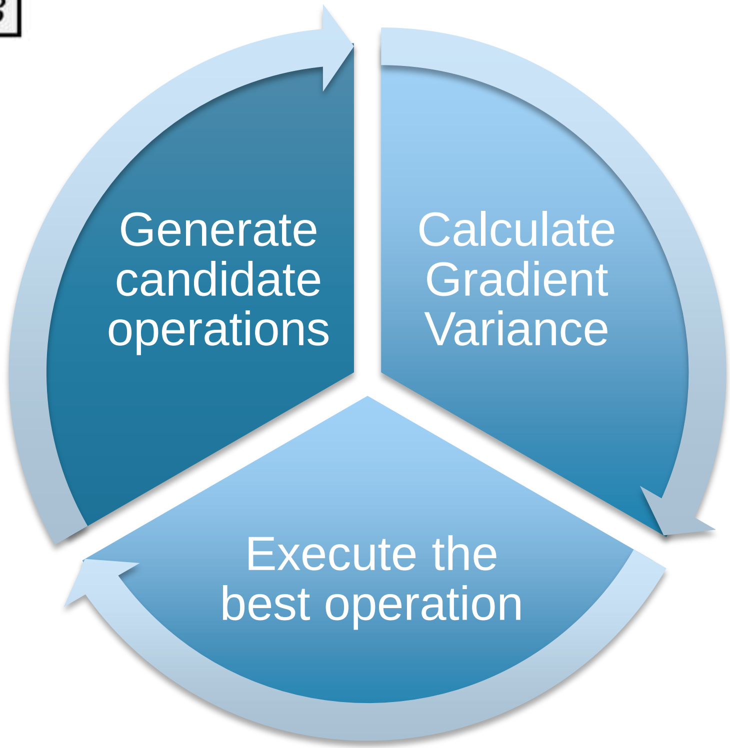
$$\Pr(Y = 1 \mid \mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{x} \cdot \boldsymbol{\theta})}$$

Features associated with each wire
"Global Circuit Flow" features

Learning parameters θ is convex optimization!

Logistic Circuit Structure Learning





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

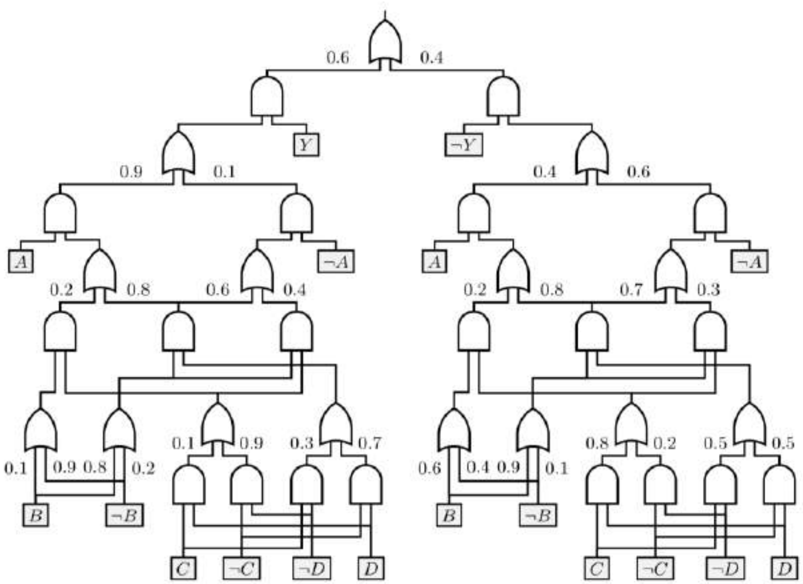
Accuracy % with % of Training Data	MNIST			Fashion		
	100%	10%	2%	100%	10%	2%
5-layer MLP	99.3	**98.2**	94.3	89.8	86.5	80.9
CNN with 3 Conv Layers	99.1	98.1	95.3	90.7	87.6	83.8
Logistic Circuit (Binary)	97.4	96.9	94.1	87.6	86.7	83.2
Logistic Circuit (Real-Valued)	**99.4**	97.6	**96.1**	**91.3**	**87.8**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

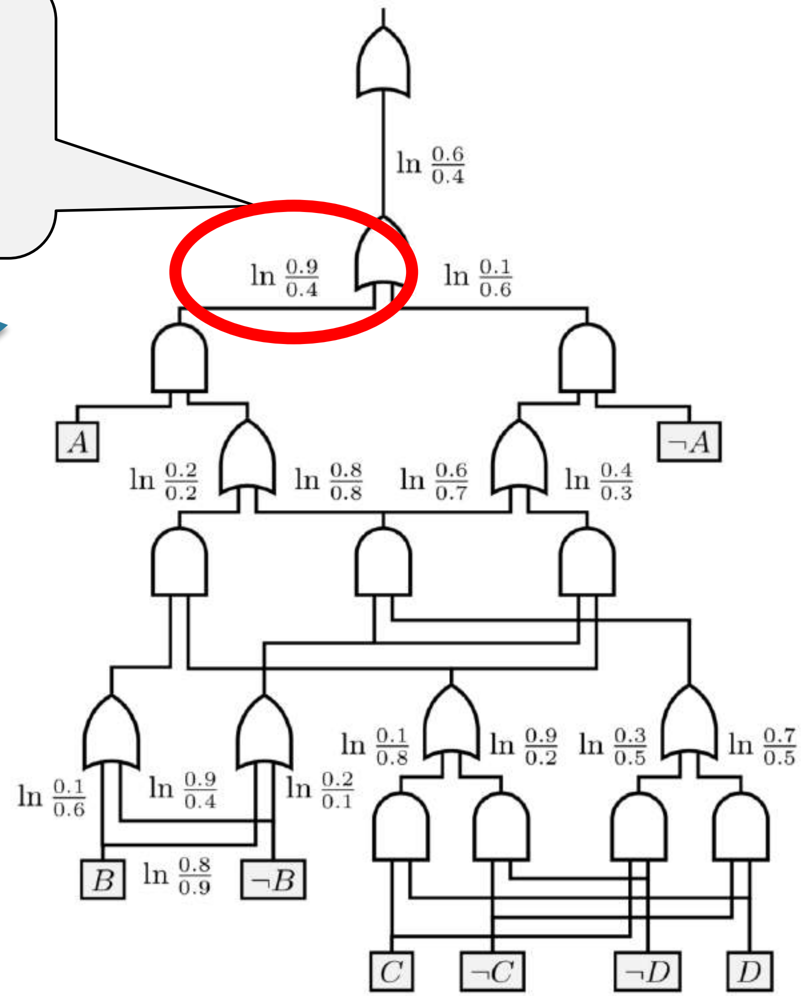
Logistic vs. Probabilistic Circuits

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

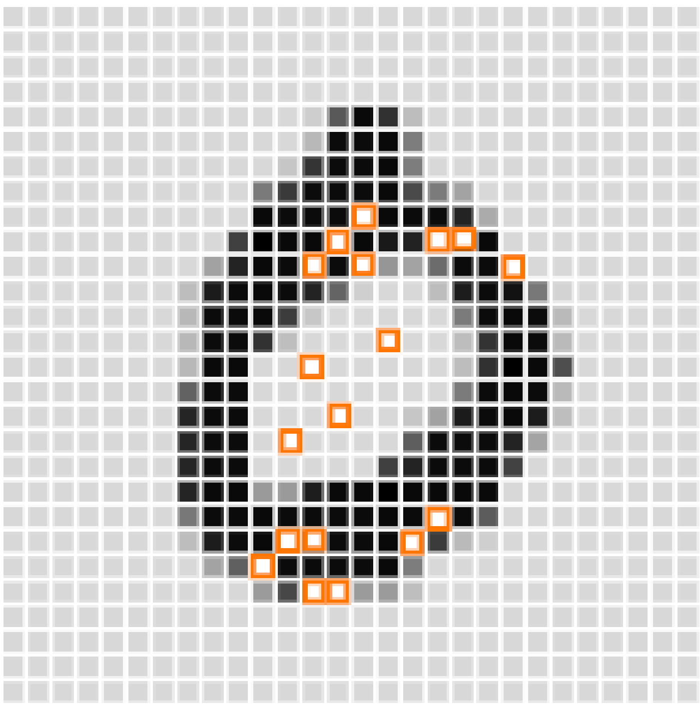
Probabilities become log-odds

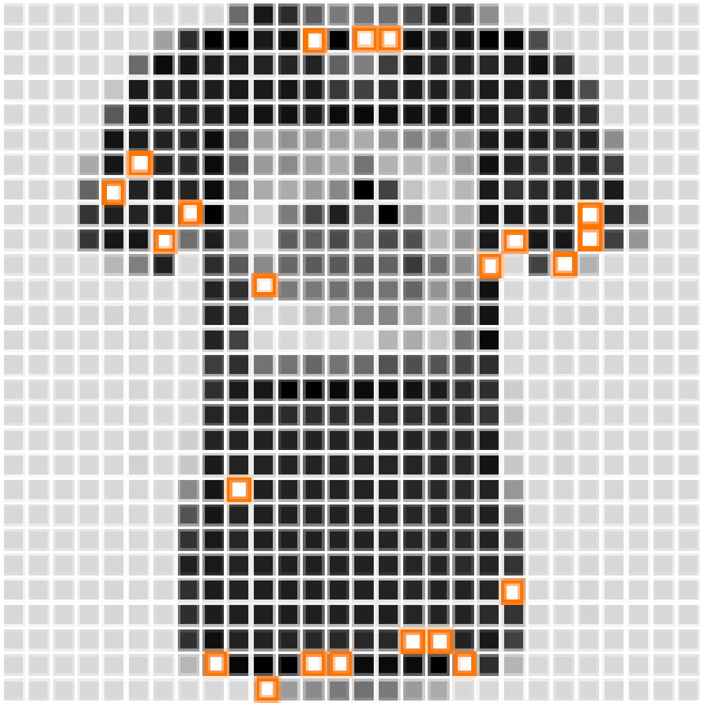
$\Pr(Y, A, B, C, D)$





Interpretable?





2+2 = Reasoning About Classifiers

2 = State-of-the-art (discrete) densities

2 = Non-compromising classifiers

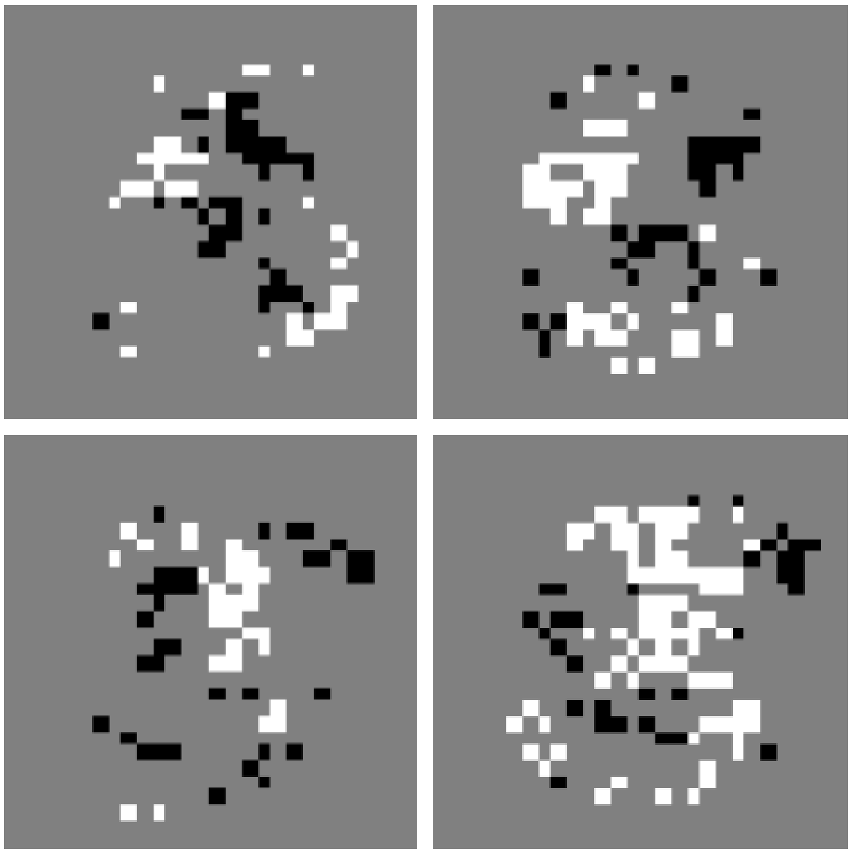
2+2= Tools for reasoning about how a classifier acts on a distribution

- Fairness
- Robustness
- Unknown unknowns
- Selection bias
- Adversarial
- Missing data
- Active sensing
- Explainability

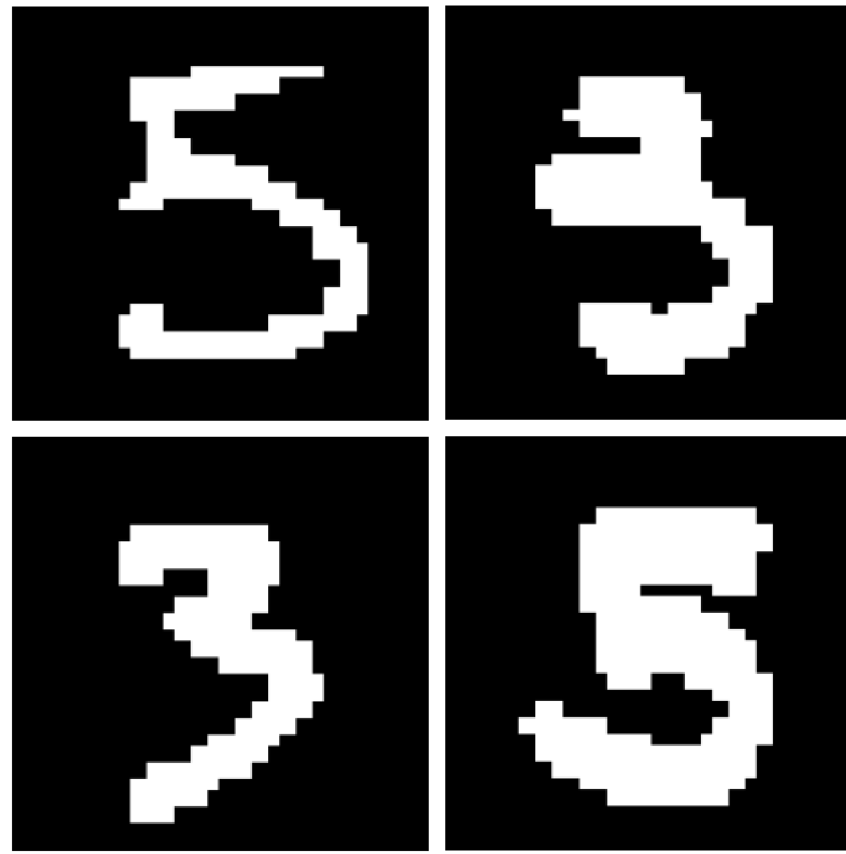
What to expect of classifiers? [IJCAI19]

- Given a predictor Y=F(X), a distribution P(X)
- What is expected prediction of F in P(X|e)?
- Computationally hard
  - Even with trivial F (#P-hard)
  - Even with trivial P (#P-hard)
  - Even with trivial F and P (NP-hard)
- But: we can do this efficiently
  on regression circuit F and
  probabilistic circuit P!

XAI User Study: 5 or 3?



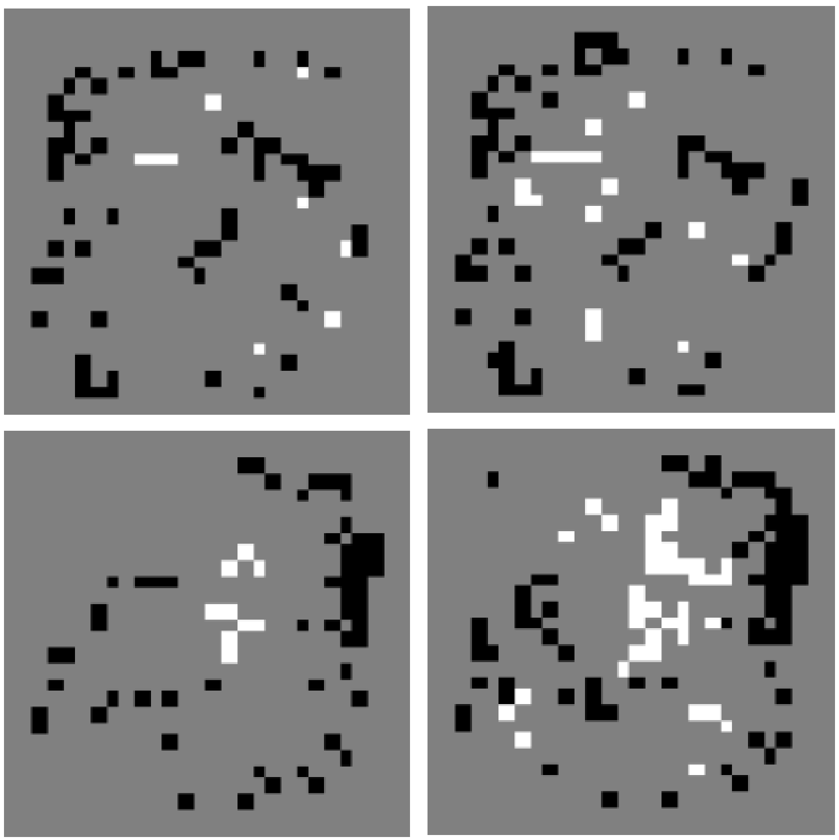
Sufficient Explanations

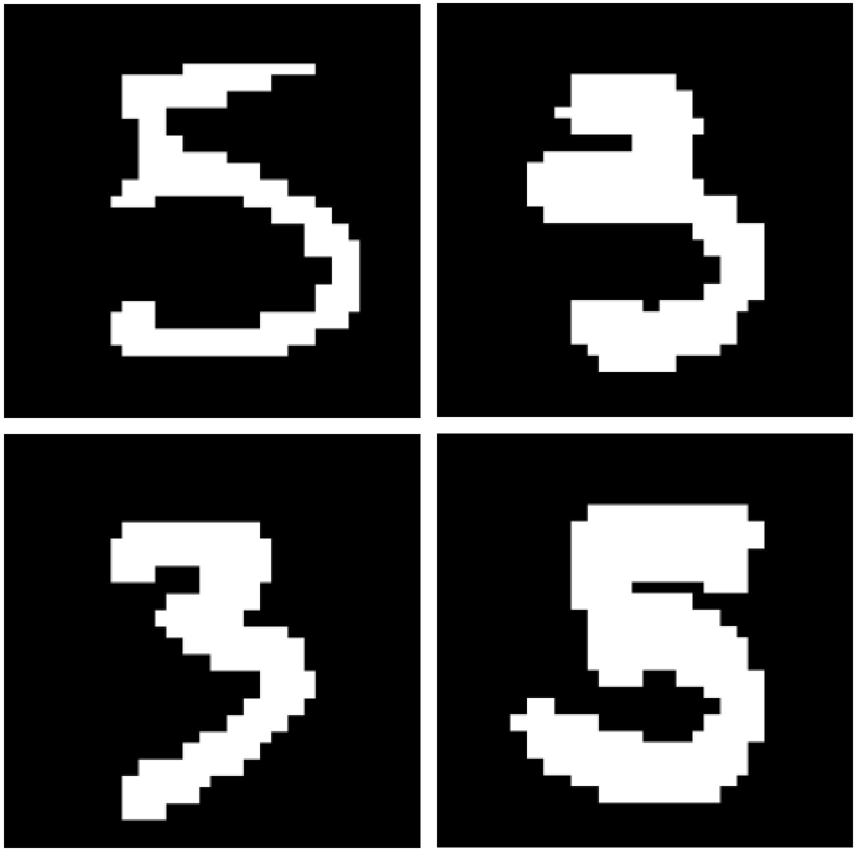


Correctly Classified

Misclassified

Compare to
Data Distribution-Unaware explanations

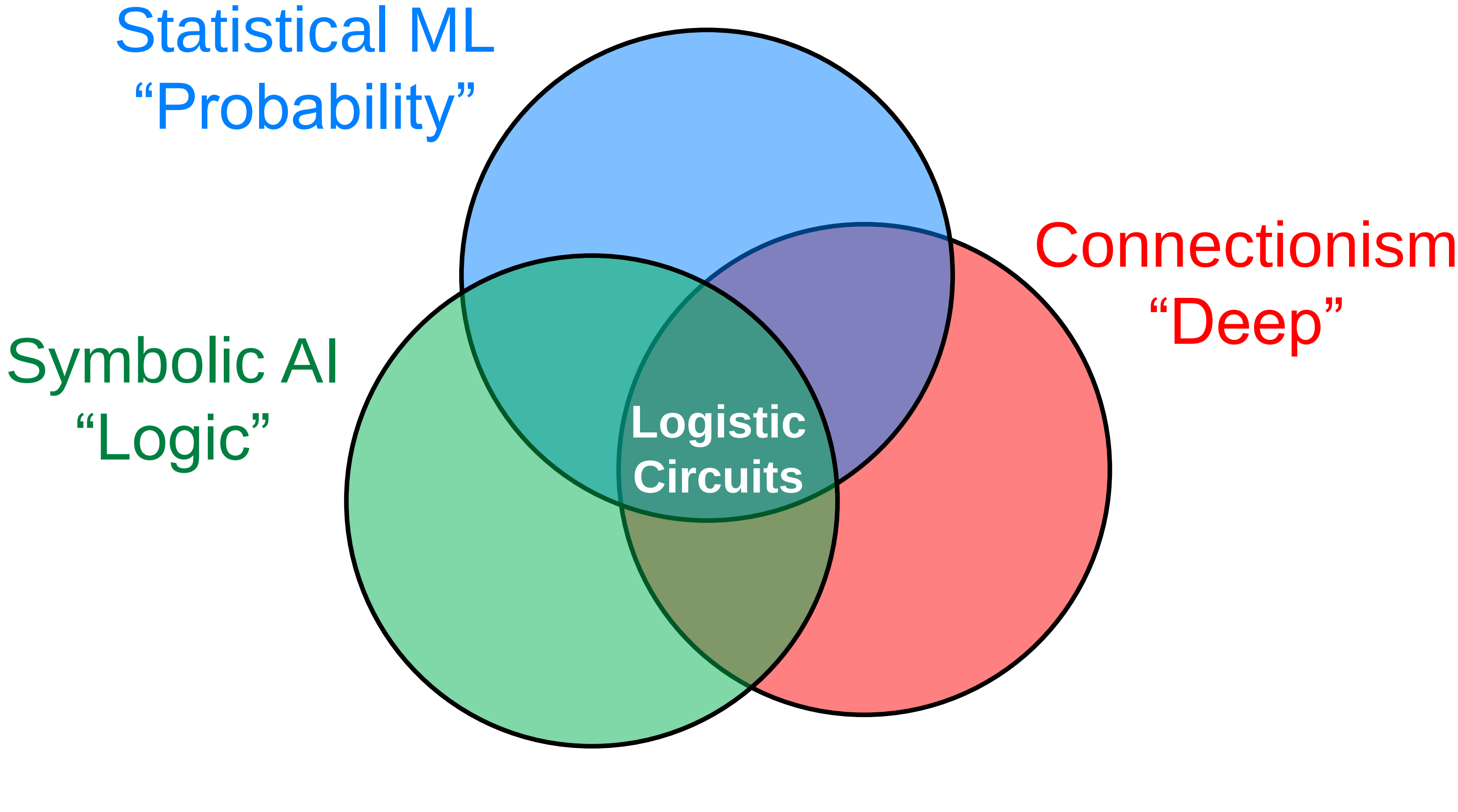




Correctly Classified

Misclassified

Conclusions 2



Final Conclusions

- Knowledge is everywhere in learning
- Some concepts not easily learned from data
- Make knowledge first-class citizen in ML

- Logical circuits turned statistical models
- Strong properties produce strong learners
- There is no dilemma between understanding and accuracy?

- A wealth of high-level reasoning approaches are still absent from ML discussion

Acknowledgements

Thanks to my students and collaborators!

Thanks for your attention!

Questions?