UCLA Computer Science

STAR AI
RESEARCH LAB
UCLA

Reasoning About the Probabilistic Behavior of Classifiers

Guy Van den Broeck

CPAIOR Master Class - Jul 5, 2021

The AI Dilemma

Pure Logic **Pure Learning**

The AI Dilemma

Pure Logic **Pure Learning**

- Slow thinking: deliberative, cognitive, model-based, extrapolation
- Amazing achievements until this day
- “*Pure logic is brittle*”
  noise, uncertainty, incomplete knowledge, …

The AI Dilemma

Pure Logic **Pure Learning**

- Fast thinking: instinctive, perceptive, model-free, interpolation
- Amazing achievements recently
- “*Pure learning is brittle*”

  bias, algorithmic fairness, interpretability, explainability, adversarial attacks, unknown unknowns, calibration, verification, missing features, missing labels, data efficiency, shift in distribution, general robustness and safety

  fails to incorporate a sensible model of the world

Pure Logic **Probabilistic World Models** **Pure Learning**

A New Synthesis of Learning and Reasoning

- “*Pure learning is brittle*”

  bias, **algorithmic fairness**, interpretability, **explainability**, adversarial attacks, unknown unknowns, calibration, **verification**, **missing features**, missing labels, data efficiency, shift in distribution, general robustness and safety

  fails to incorporate a sensible model of the world

Outline

1. Theoretical motivation
   Tractability of SHAP explanations [AAAI'21]
2. Tractable reasoning about classifier behavior
   Reasoning about missing features [NeurIPS'19]
   Latent fair decisions [AAAI'21]
   Probabilistic sufficient explanations [IJCAI'21]
3. Practical neuro-symbolic verification
   Learning monotonic neural networks [NeurIPS'20]

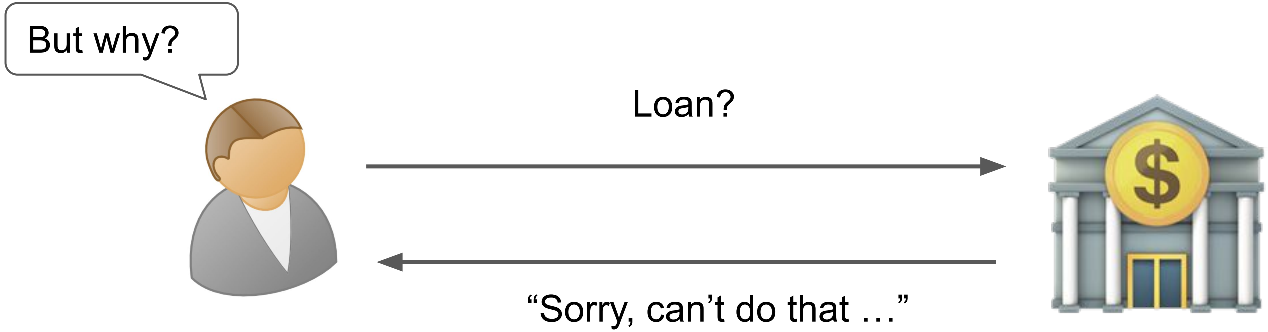
Outline

1. Theoretical motivation

   Tractability of SHAP explanations [AAAI'21]

   with Anton Lykov, Maximilian Schleich, Dan Suciu

Motivation: Explainable AI



F() = no loan

Instance **x**
with Features X1, ... ,Xn

Input

Prediction F(**x**) **+ Explanation**

Classifier F

We study:
Computational Complexity of **SHAP Explanations**

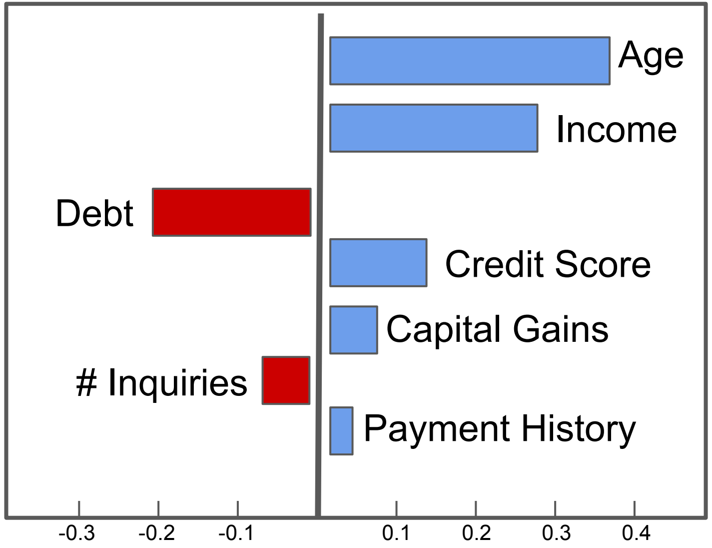
What are SHAP explanations?

Feature-Based Attribution Score

- How much does ith feature influence F(**x**)?
- Based on Shapley values from Game Theory

Benefits

- Model-agnostic
- Intuitive
- Successfully applied in practice



Computing SHAP Explanations

Intuition:

- Assume a total order π of the features
- Compute effect on **E**[F] of presenting one feature at a time following π

Example:

- Assume π = [*X*1,*X*2,...,*Xn*]
- Contribution of X2 w.r.t. π

$$c_\pi(X2) = \mathbf{E}[F \mid X1, X2] - \mathbf{E}[F \mid X1]$$

SHAP-score for X2:

Average contribution of X2 over all possible permutations

$$SHAP_{F,\mathbf{x}}(X2) = \frac{1}{n!} \sum_\pi c_\pi(X2)$$

The Challenge

Various algorithms proposed to compute SHAP explanations:

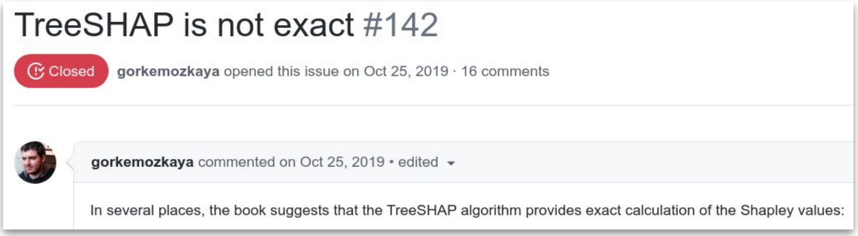
approximately, exactly, efficiently, …, for different machine learning models

There is considerable confusion about the tractability of computing SHAP explanations

- Are the exact algorithms exact, correct, and efficient?
- Are the approximations needed?

<u>Example</u>: TreeSHAP [ICML 2017]

How can we clear this up?



The Main Actors

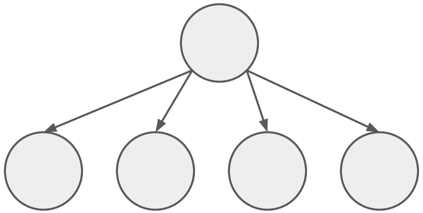
1. The machine learning model class for function F

Linear regression, decision and regression trees, random forests, additive tree ensembles, logistic regression, neural nets with sigmoid activation functions, naive Bayes classifiers, factorization machines, regression circuits, logistic circuits, Boolean functions in d-DNNF, binary decision diagrams, bounded treewidth Boolean functions in CNF, Boolean functions in CNF or DNF, and arbitrary functions

2. The data distribution Pr to compute $\mathbf{E}[F|\mathbf{y}] = \sum_{\mathbf{x}} \Pr(\mathbf{x}|\mathbf{y})\, F(\mathbf{x})$

Fully-factorized distributions

Graphical models (naive Bayes)



Empirical data distribution

Fully-factorized distributions

Key result:

For any classifier F, the following problems have the same complexity:

- Computing **SHAP** explanations of F
- Computing the expectation **E** of F

Expectations **E** are efficient to compute for

- linear regression
- decision trees, random forests, additive tree ensembles
- Boolean functions in d-DNNF form, bounded-treewidth CNF
- … and more

therefore

SHAP explanations are efficient to compute on those same models!

Fully-factorized distributions

Key result:

For any classifier F, the following problems have the same complexity:

- Computing **SHAP** explanations of F
- Computing the expectation **E** of F

We prove that expectations **E** are #P-hard to compute for

- logistic regression
- naive Bayes classifiers
- neural networks with sigmoid activations
- Boolean functions in CNF or DNF

therefore

SHAP explanations are #P-hard to compute on those same models!

Intuition: Expectation of Logistic Regression

Consider the number partitioning problem for {1,2,3,2}

- {1,3} and {2,2} partition the set into subsets with the same sum
- Counting such partitions is #P-hard

Consider the logistic regression model:

$F(\mathbf{X}) = \text{sigmoid}(1000\ X1 + 2000\ X2 + 3000\ X3 + 2000\ X4 - 4500)$

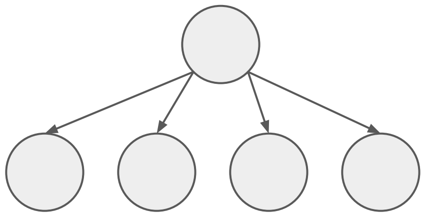
- $\mathbf{x} = [1,1,0,1]$ and $\mathbf{x'} = [0,0,1,0]$ correspond to non-partitions: $F(\mathbf{x}) \approx 1$ and $F(\mathbf{x'}) \approx 0$
- Under a uniform distribution $\mathbf{E}[F] \approx 0.5$
- $\mathbf{x} = [1,0,1,0]$ and $\mathbf{x'} = [0,1,0,1]$ correspond to partitions: $F(\mathbf{x}) = F(\mathbf{x'}) \approx 0$
- Missing probability mass 0.5 - $\mathbf{E}[F]$ tells us how many partitions there are
- Computing $\mathbf{E}[F]$ is #P-hard

Going Beyond Fully-Factorized Distributions

Idea: the real world is not fully-factorized: features depend on each other

Consider the simplest case:

1. Simplest possible classifier: F(**X**) = X1
2. Simplest tractable distribution: naive Bayes



SHAP explanations are NP-hard to compute.

SHAP explanations are NP-hard to compute for all probabilistic graphical models, even all tractable probabilistic models, even on simple function classes

Trivial function classes do not make **SHAP** tractable...

Empirical Distributions

Idea: Properties of distributions are often estimated on sampled data.
Perhaps the empirical data distribution is easier to work with?

The # of possible worlds is limited by the number of rows (samples) in data

Computing **SHAP** is #P-hard in the size of the empirical distribution.

The problem that TreeSHAP is trying to solve efficiently is in fact #P-hard

Summary of Contributions

-

	Distribution Pr		
Predictive Model F	Fully Factorized	Naive-Bayes	Empirical
Linear regression Regression circuits Factorization machines	**Tractable**	**Intractable**	**Intractable**
Decision Tree Random Forest,Boosted Tree	**Tractable**	**Intractable**	**Intractable**
Boolean functions in d-DNNF, BDD, Bounded treewidth CNF	**Tractable**	**Intractable**	**Intractable**
Logistic regression Logistic circuits, Naive Bayes	**Intractable**	**Intractable**	**Intractable**
Neural Networks with sigmoid activation	**Intractable**	**Intractable**	**Intractable**

- Proved connections between SHAP and the expectation of classifiers
- … and more theoretical insights of independent interest

Then how can we reason about the behavior of classifiers under a non-trivial feature distribution?

Outline

1. Theoretical motivation
   Tractability of SHAP explanations [AAAI'21]
2. **Tractable reasoning about classifier behavior**

Fully factorized

NaiveBayes AndOrGraphs PDGs

Trees PSDDs CNets LTMs SPNs NADEs

Thin Junction Trees ACs MADEs MAFs VAEs

DPPs FVSBNs TACs IAFs NAFs RAEs

Mixtures BNs NICE FGs GANs

RealNVP MNs

The Alphabet Soup of probabilistic models

Fully factorized

NaiveBayes AndOrGraphs PDGs

Trees PSDDs CNets LTMs SPNs NADEs

Thin Junction Trees ACs MADEs MAFs VAEs

DPPs FVSBNs TACs IAFs NAFs RAEs

Mixtures BNs NICE FGs GANs

RealNVP MNs

Intractable and *tractable* models

Fully factorized

NaiveBayes AndOrGraphs PDGs

Trees PSDDs CNets LTMs SPNs NADEs

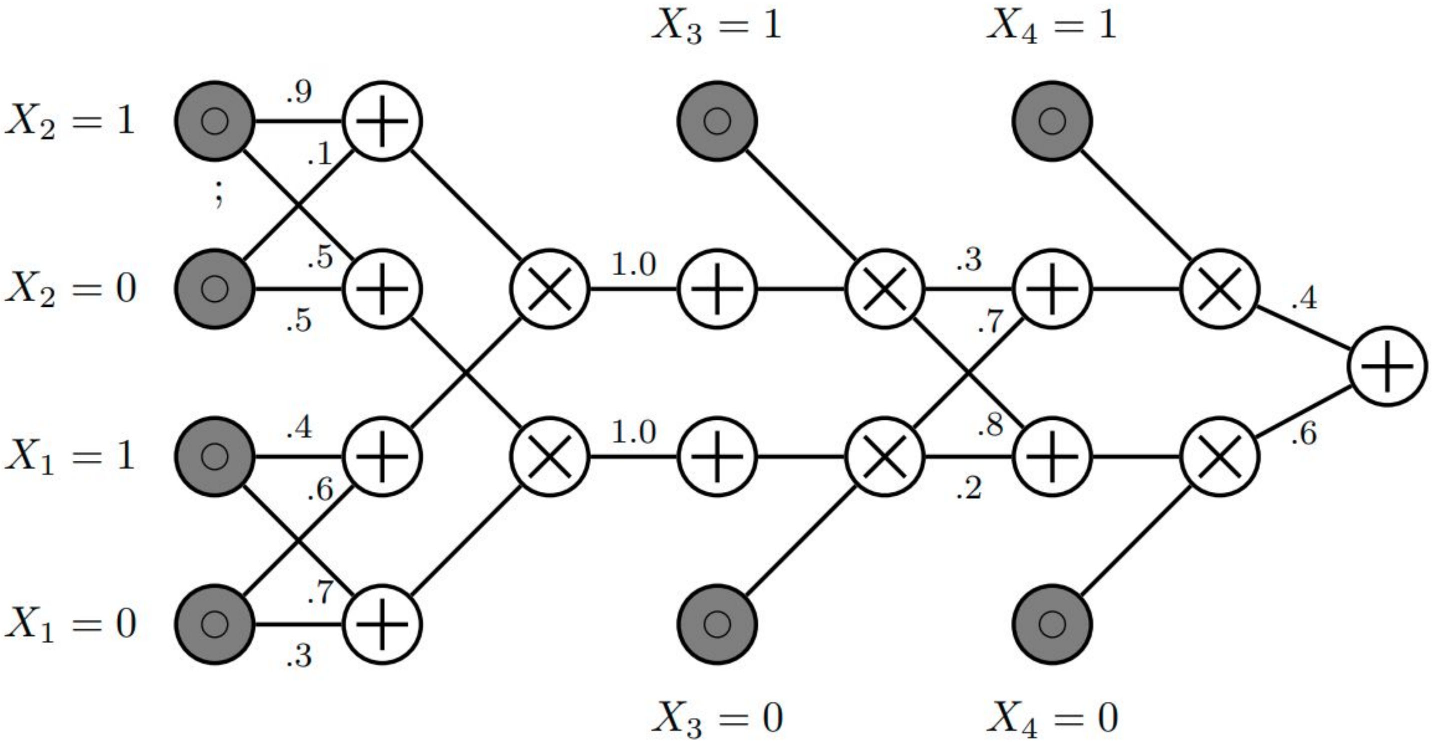
Thin Junction Trees ACs MADEs MAFs VAEs

DPPs FVSBNs TACs IAFs NAFs RAEs

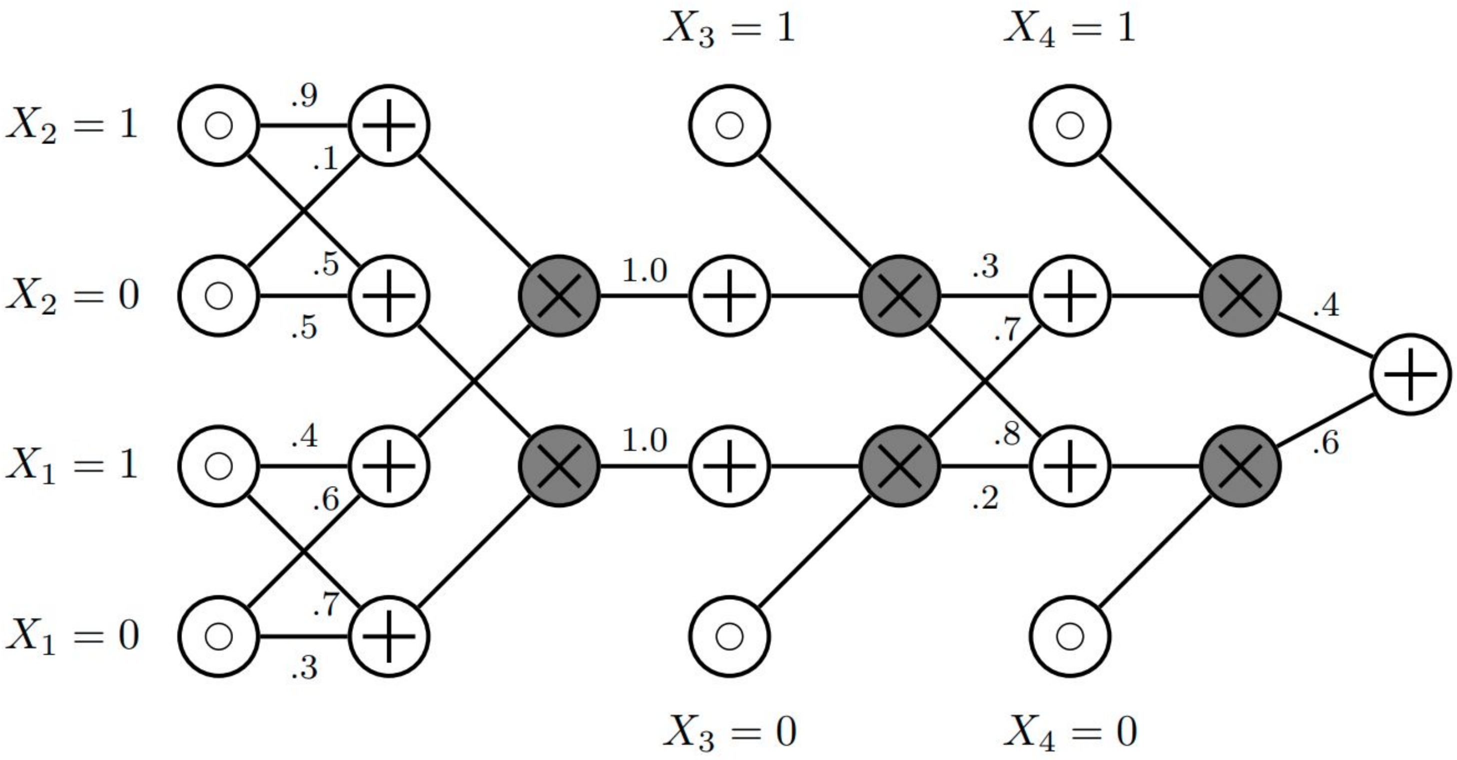
Mixtures BNs NICE FGs GANs

RealNVP MNs

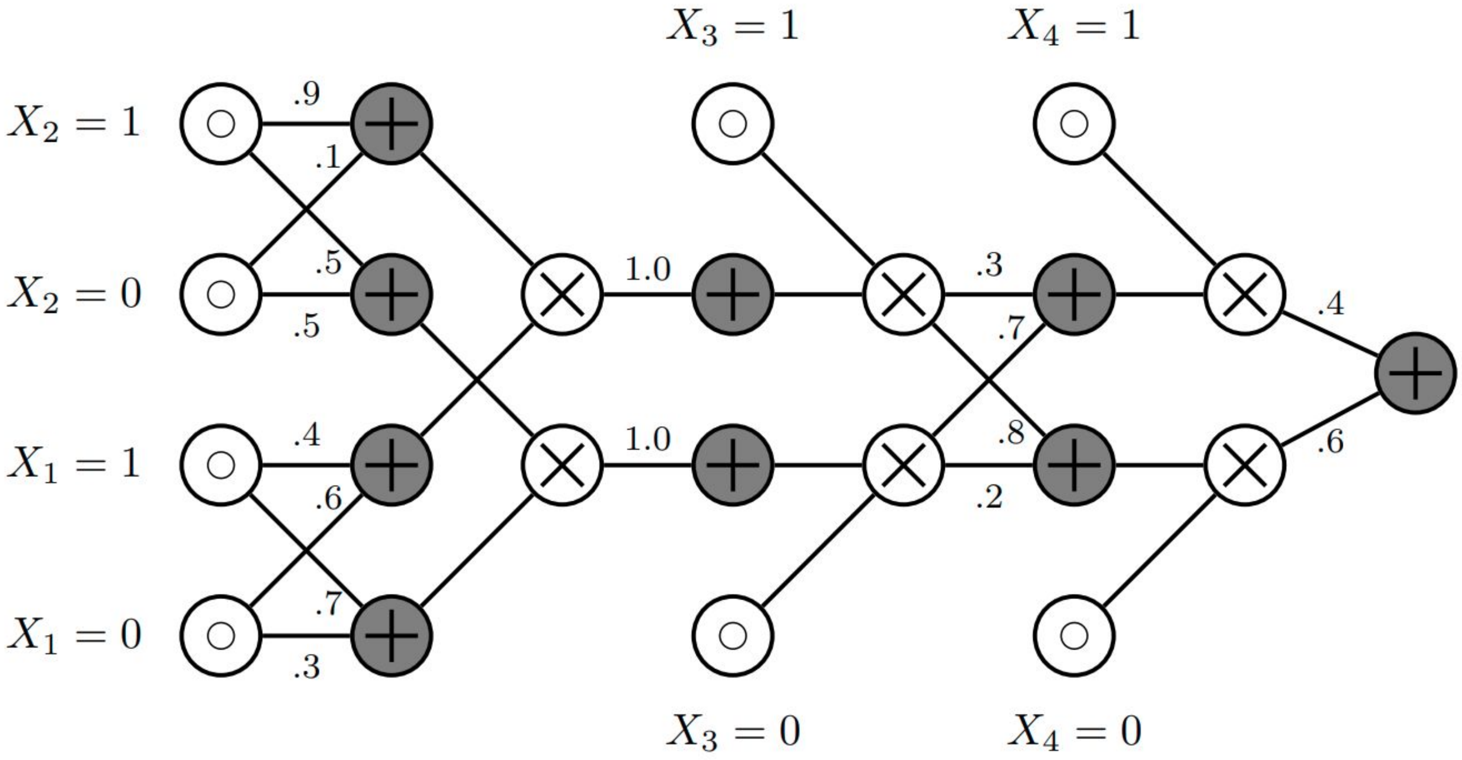
a *unifying framework* for tractable models



Input nodes are tractable (simple) distributions,
e.g., univariate gaussian or indicator p(X=1) = [X=1]

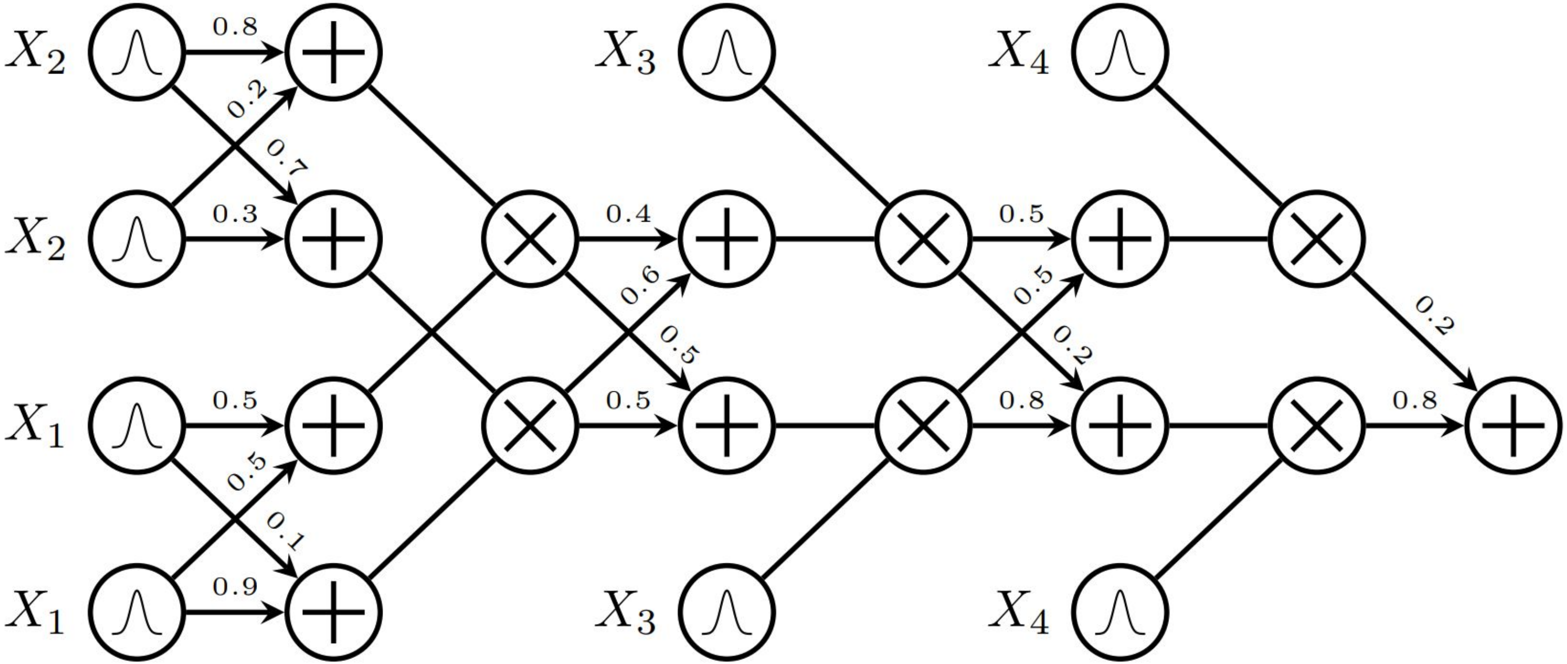


Product nodes are factorizations $\prod_{c \in \mathsf{in}(n)} \mathrm{p}_c(\mathbf{x})$

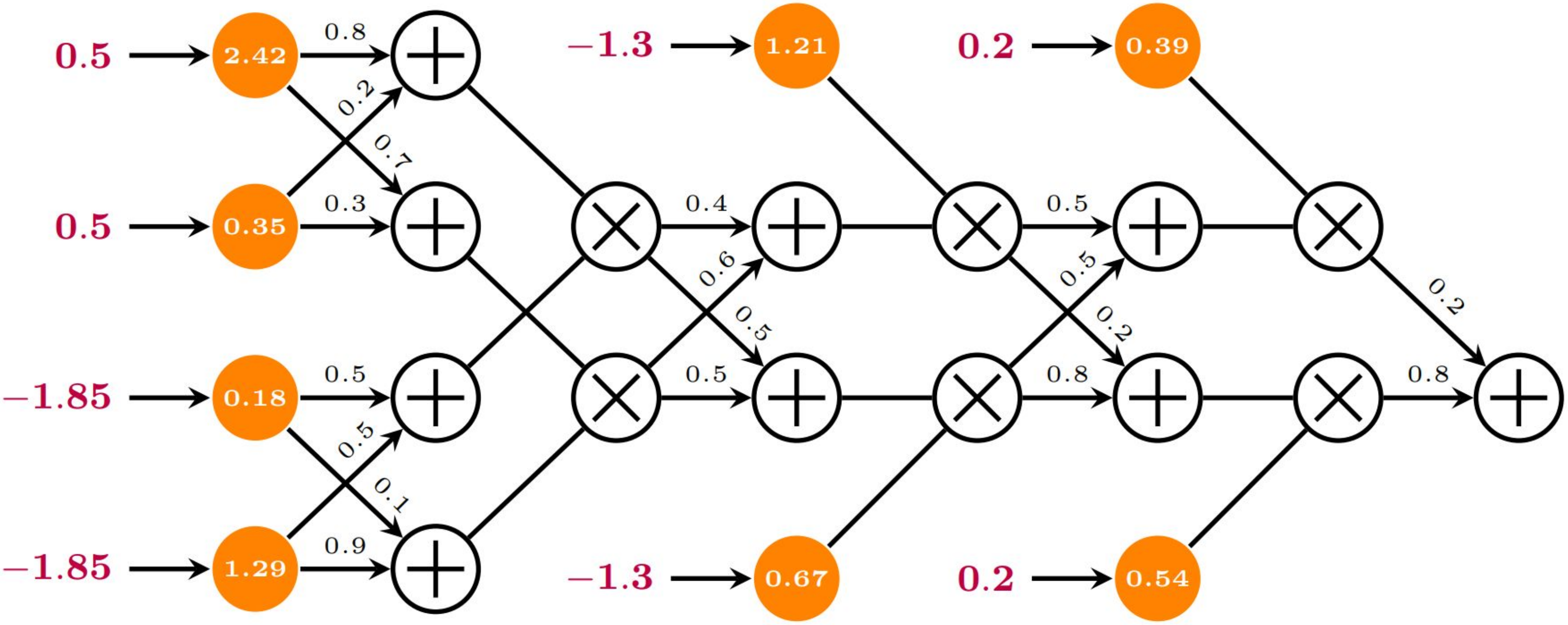


Sum nodes are mixture models $\sum_{c \in \mathsf{in}(n)} \theta_{n,c} \, \mathrm{p}_c(\mathbf{x})$

Feedforward $p(X_1 = \mathbf{-1.85}, X_2 = \mathbf{0.5}, X_3 = \mathbf{-1.3}, X_4 = \mathbf{0.2})$

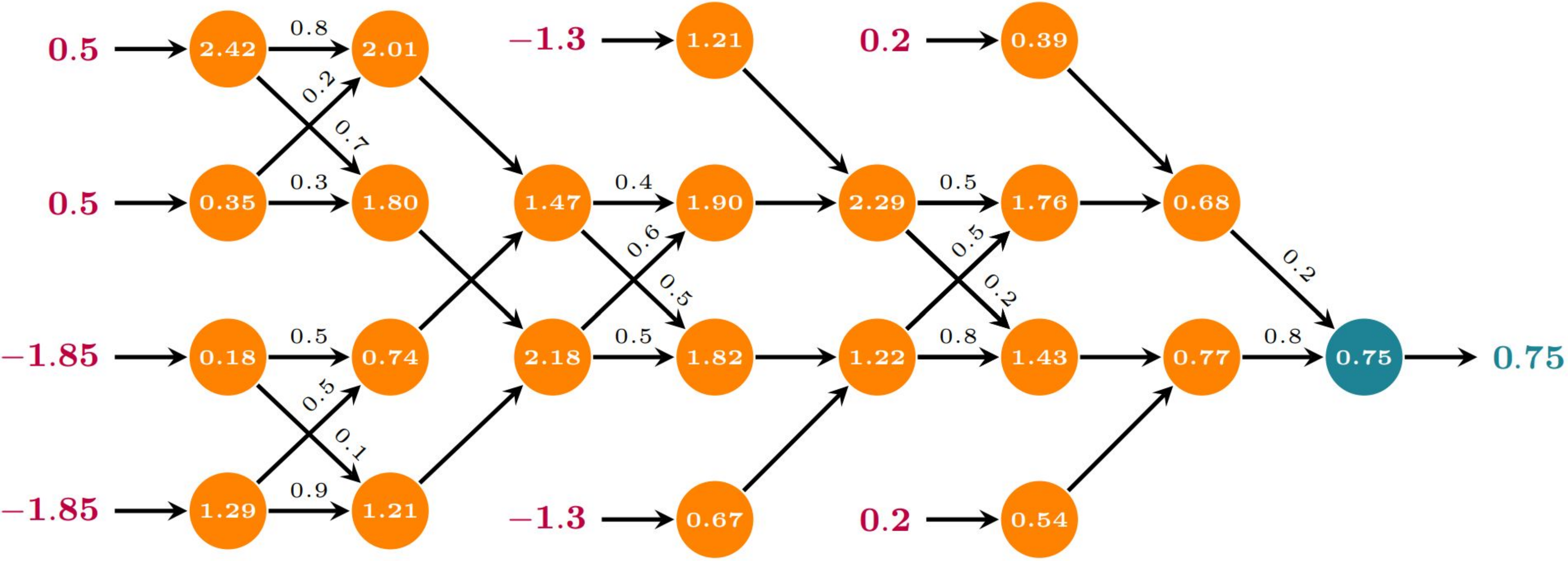


Feedforward $p(X_1 = \mathbf{-1.85}, X_2 = \mathbf{0.5}, X_3 = \mathbf{-1.3}, X_4 = \mathbf{0.2})$



Feedforward

$p(X_1 = \mathbf{-1.85}, X_2 = \mathbf{0.5}, X_3 = \mathbf{-1.3}, X_4 = \mathbf{0.2})$



Why are these tractable models?

Let's compute a marginal probability $\int \boldsymbol{p}(\mathbf{x})d\mathbf{x}$.

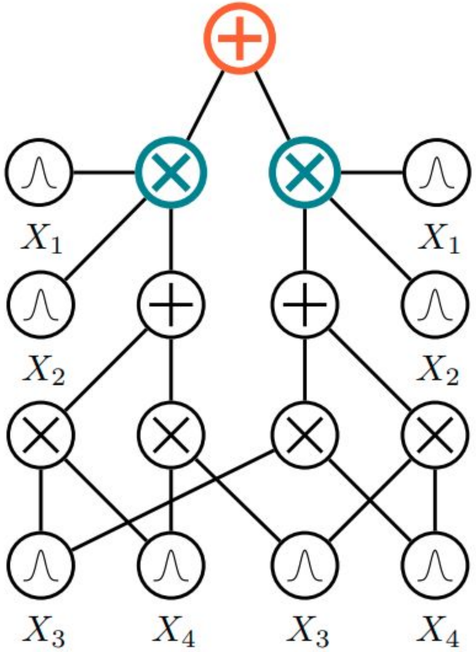
Smoothness + decomposability = tractable MAR

If $p(\mathbf{x}) = \sum_i w_i p_i(\mathbf{x})$, (***smoothness***):

$$\int p(\mathbf{x}) d\mathbf{x} = \int \sum_i w_i p_i(\mathbf{x}) d\mathbf{x} =$$

$$= \sum_i w_i \int p_i(\mathbf{x}) d\mathbf{x}$$

$\Rightarrow$ *integrals are "pushed down" to children*



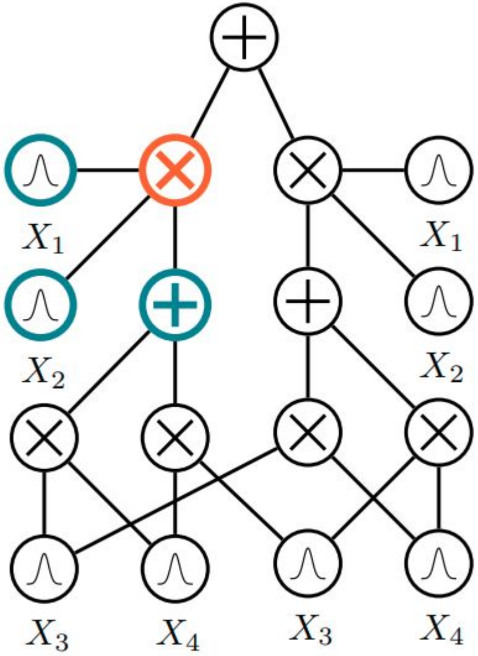
[Darwiche & Marquis JAIR 2001, Poon & Domingos UAI11]

Smoothness + decomposability = tractable MAR

If $p(\mathbf{x}, \mathbf{y}, \mathbf{z}) = p(\mathbf{x})p(\mathbf{y})p(\mathbf{z})$, (***decomposability***):

$$\int\int\int p(\mathbf{x}, \mathbf{y}, \mathbf{z})d\mathbf{x}d\mathbf{y}d\mathbf{z} =$$
$$= \int\int\int p(\mathbf{x})p(\mathbf{y})p(\mathbf{z})d\mathbf{x}d\mathbf{y}d\mathbf{z} =$$
$$= \int p(\mathbf{x})d\mathbf{x} \int p(\mathbf{y})d\mathbf{y} \int p(\mathbf{z})d\mathbf{z}$$

$\Rightarrow$ *integrals decompose into easier ones*



Smoothness + decomposability = tractable MAR

Forward pass evaluation for MAR

$\Rightarrow$ *linear in circuit size!*

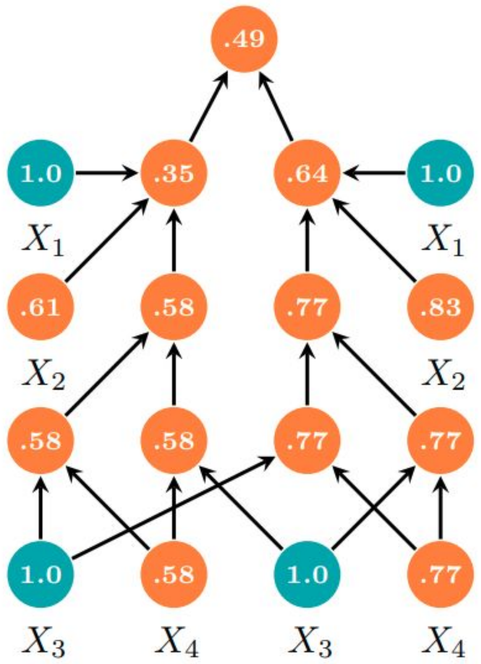
E.g. to compute $p(x_2, x_4)$:

- leafs over X_1 and X_3 output $Z_i = \int p(x_i)dx_i$

  $\Rightarrow$ *for normalized leaf distributions:* 1.0

- leafs over X_2 and X_4 output ***EVI***

- feedforward evaluation (bottom-up)



Fully factorized

NaiveBayes *AndOrGraphs* *PDGs*

Trees *PSDDs* *CNets* *LTMs* *SPNs* *NADEs*

Thin Junction Trees *ACs* *MADEs* *MAFs* *VAEs*

DPPs *FVSBNs* *TACs* *IAFs* *NAFs* *RAEs*

Mixtures *BNs* *NICE* *FGs* *GANs*

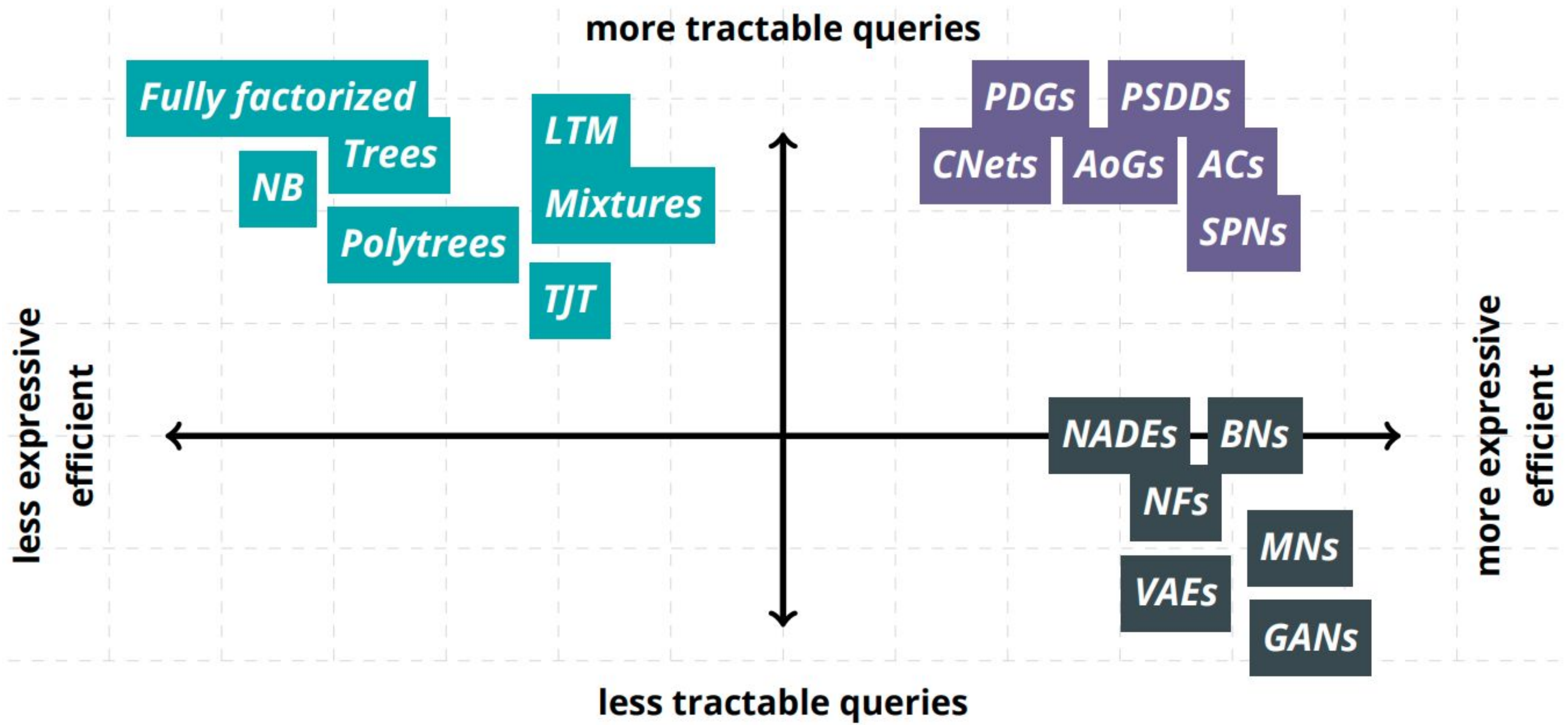
RealNVP *MNs*

Expressive **models without** ***compromises***

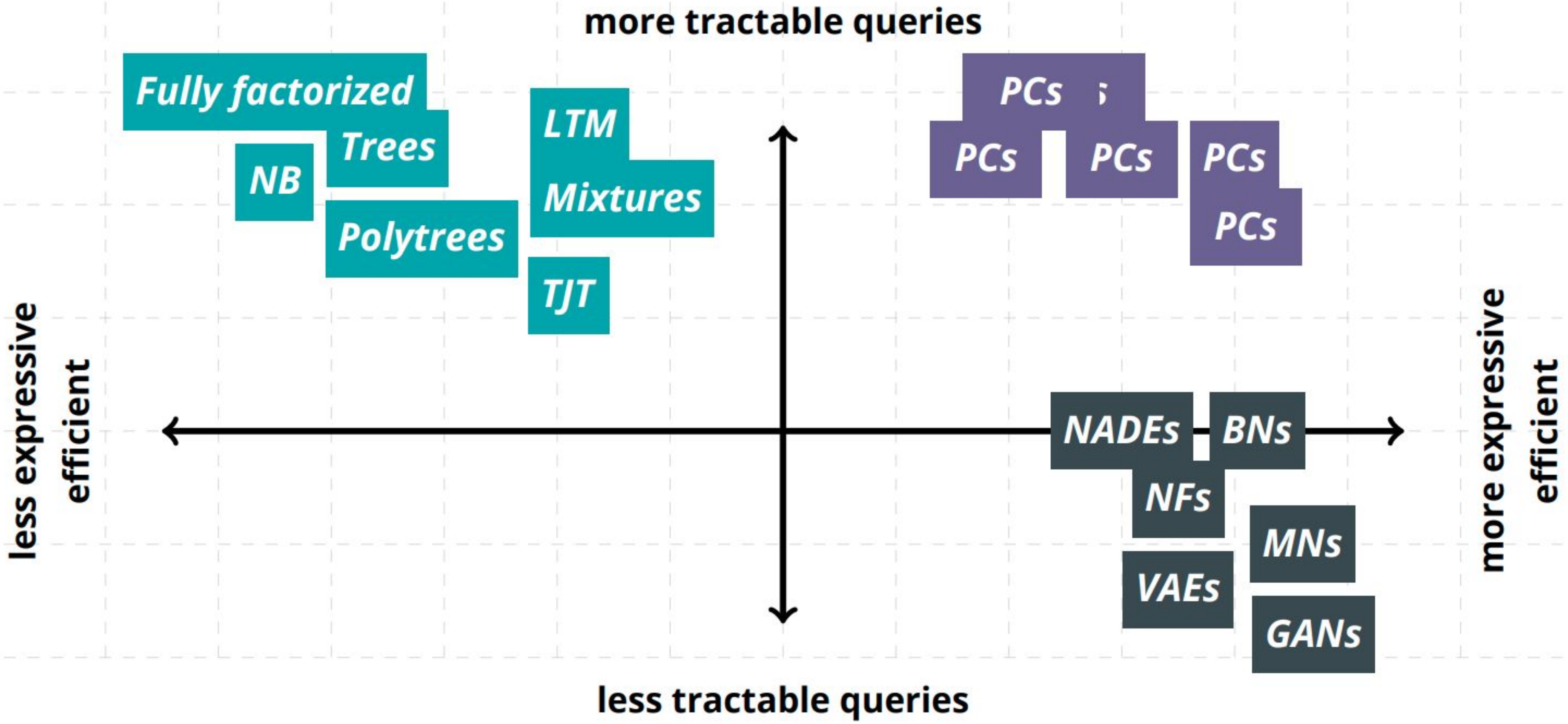
How expressive are probabilistic circuits?

density estimation benchmarks

dataset	best circuit	BN	MADE	VAE	dataset	best circuit	BN	MADE	VAE
nltcs	**-5.99**	-6.02	-6.04	**-5.99**	***dna***	**-79.88**	-80.65	-82.77	-94.56
msnbc	**-6.04**	**-6.04**	-6.06	-6.09	***kosarek***	**-10.52**	-10.83	-	-10.64
kdd	-2.12	-2.19	**-2.07**	-2.12	***msweb***	-9.62	-9.70	**-9.59**	-9.73
plants	**-11.84**	-12.65	-12.32	-12.34	***book***	-33.82	-36.41	-33.95	**-33.19**
audio	-39.39	-40.50	-38.95	**-38.67**	***movie***	-50.34	-54.37	-48.7	**-47.43**
jester	-51.29	**-51.07**	-52.23	-51.54	***webkb***	-149.20	-157.43	-149.59	**-146.9**
netflix	-55.71	-57.02	-55.16	**-54.73**	***cr52***	-81.87	-87.56	-82.80	**-81.33**
accidents	-26.89	**-26.32**	-26.42	-29.11	***c20ng***	-151.02	-158.95	-153.18	**-146.9**
retail	**-10.72**	-10.87	-10.81	-10.83	***bbc***	**-229.21**	-257.86	-242.40	-240.94
pumbs*	-22.15	**-21.72**	-22.3	-25.16	***ad***	-14.00	-18.35	**-13.65**	-18.81



more tractable queries
Fully factorized
Trees
NB
LTM
Mixtures
Polytrees
TJT
PDGs
PSDDs
CNets
AoGs
ACs
SPNs
less expressive
efficient
more expressive
efficient
NADEs
BNs
NFs
MNs
VAEs
GANs
less tractable queries



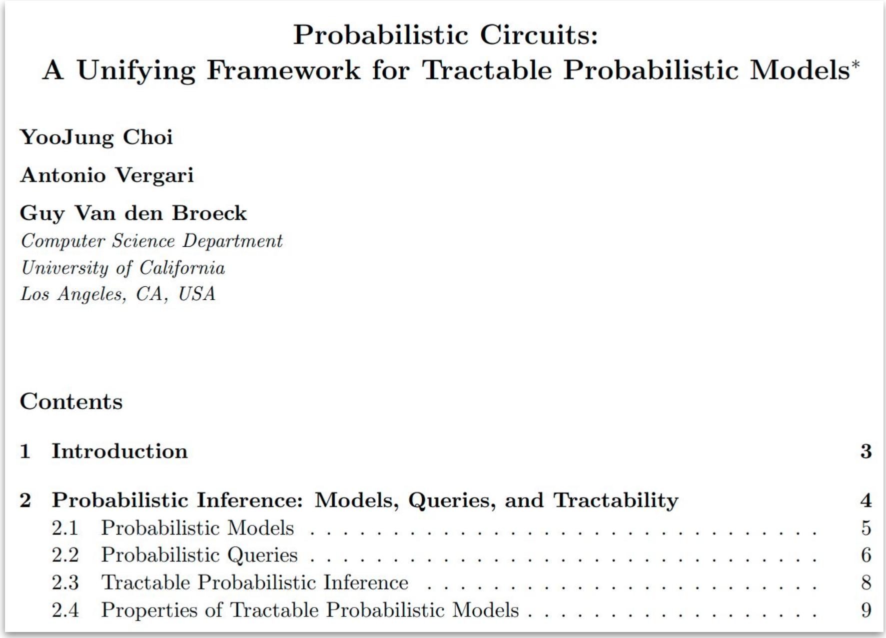
more tractable queries
Fully factorized
Trees
NB
LTM
Mixtures
Polytrees
TJT
PCs
PCs
PCs
PCs
PCs
less expressive
efficient
more expressive
efficient
NADEs
BNs
NFs
MNs
VAEs
GANs
less tractable queries

Want to learn more?

Tutorial (3h)

https://youtu.be/2RAG5-L9R70

Overview Paper (80p)



Probabilistic Circuits:
A Unifying Framework for Tractable Probabilistic Models*

YooJung Choi

Antonio Vergari

Guy Van den Broeck

Computer Science Department
University of California
Los Angeles, CA, USA

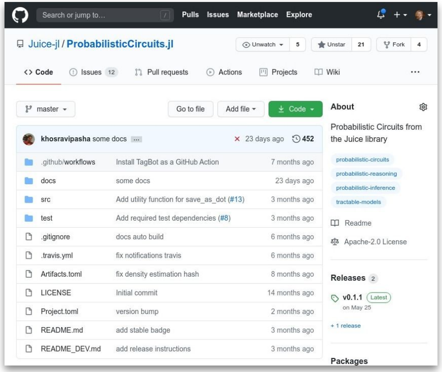
Contents

http://starai.cs.ucla.edu/papers/ProbCirc20.pdf

Training PCs in Julia with Juice.jl

Training maximum likelihood parameters of probabilistic circuits

```
julia> using ProbabilisticCircuits;
julia> data, structure = load(...);
julia> num_examples(data)
17,412
julia> num_edges(structure)
270,448
julia> @btime estimate_parameters(structure , data);
63 ms
```



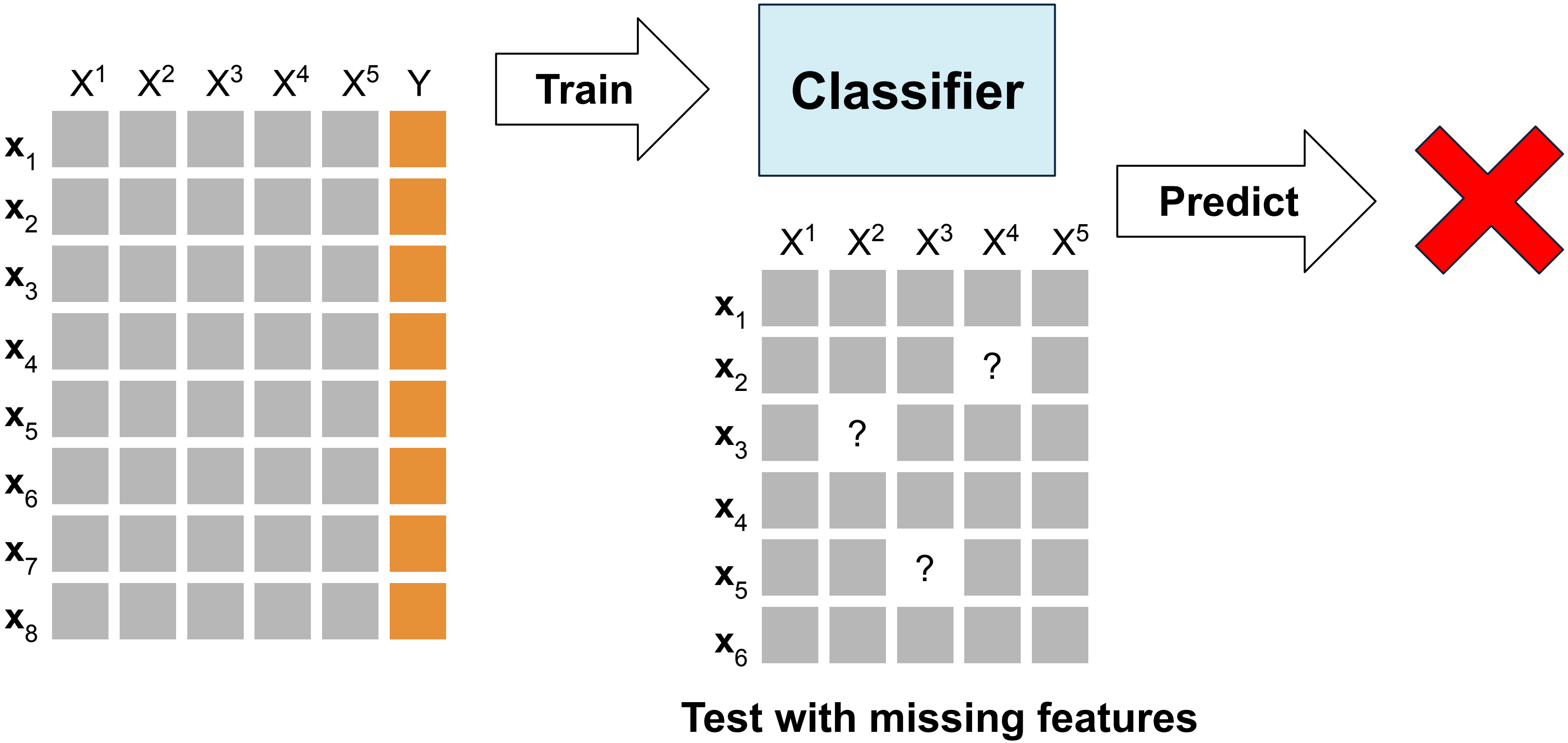
Custom SIMD and CUDA kernels to parallelize over layers and training examples.

[https://github.com/Juice-jl/]

Outline

1. Theoretical motivation
   Tractability of SHAP explanations [AAAI'21]
2. Tractable reasoning about classifier behavior
   Reasoning about missing features [NeurIPS'19]
   with Pasha Khosravi, YooJung Choi, Yitao Liang, Antonio Vergari

Prediction with Missing Features



Expected Predictions

Consider **all possible complete inputs** and **reason** about the *expected* behavior of the classifier

$$\mathbb{E}_{\mathbf{x}^m \sim p(\mathbf{x}^m|\mathbf{x}^o)}\left[f\left(\mathbf{x}^m\mathbf{x}^o\right)\right]$$

$\boldsymbol{x}^{\mathrm{o}}$ = observed features
$\boldsymbol{x}^{\mathrm{m}}$ = missing features

How can this be tractable for a complex feature distribution?

- feature distribution is a probabilistic circuits
- classifier is a compatible regression circuit

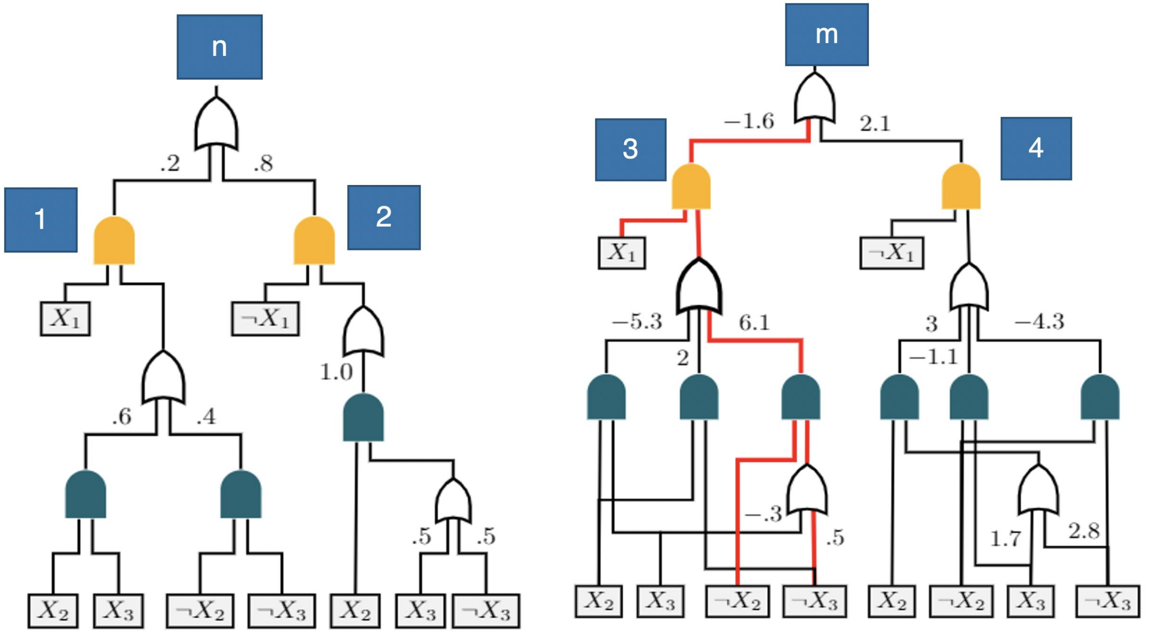
☺

[Khosravi et al. IJCAI19, NeurIPS19, Artemiss20]

Recursion that “breaks down” the computation

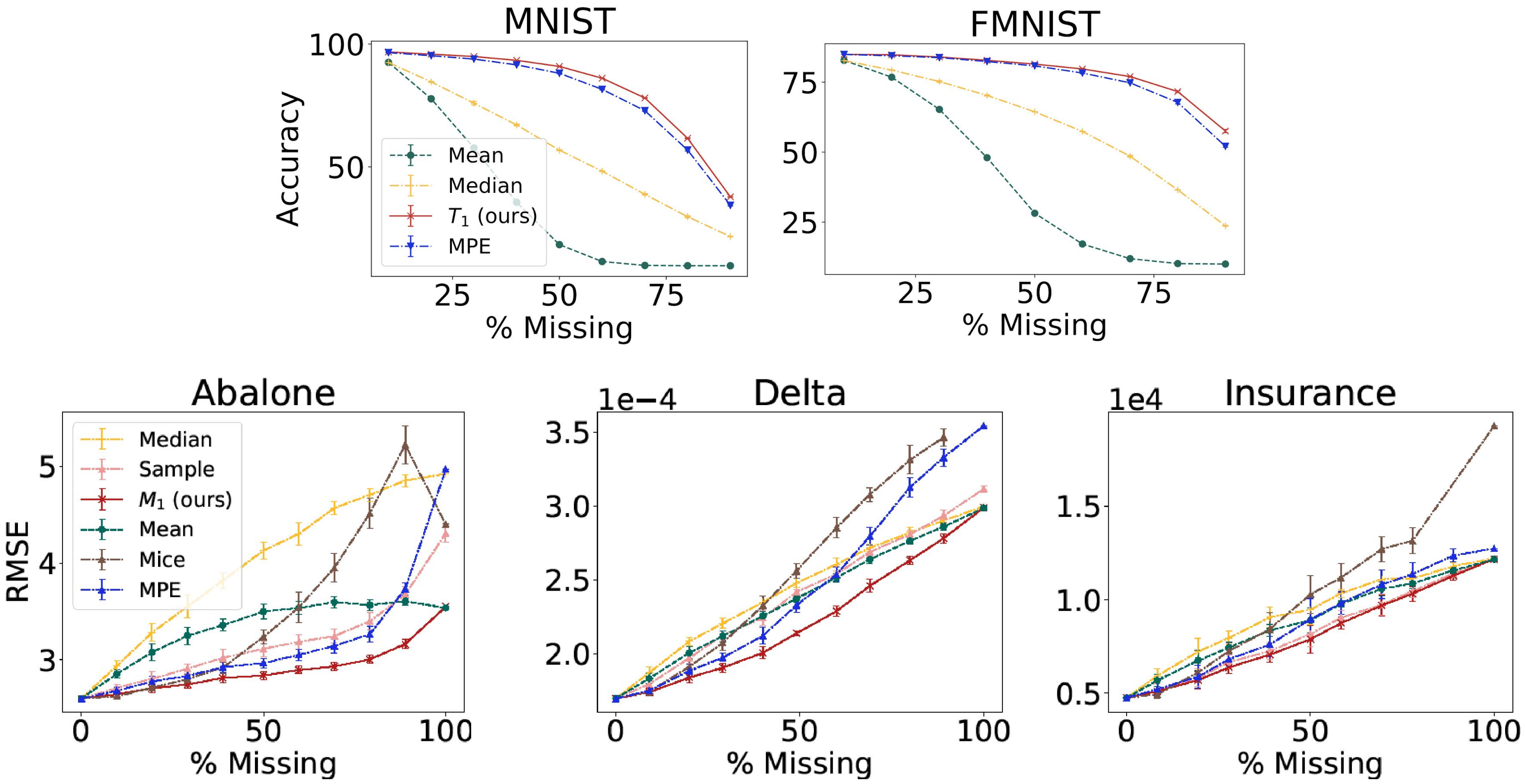
Expectation of
function m w.r.t. dist. n ?

Solve subproblems:
(1,3), (1,4), (2,3), (2,4)



[Khosravi et al. IJCAI19, NeurIPS20, Artemiss20]

Probabilistic Circuits for Missing Data



[Khosravi et al. IJCAI19, NeurIPS20, Artemiss20]

ADV inference in Julia with Juice.jl

```julia
using ProbabilisticCircuits
pc = load_prob_circuit(zoo_psdd_file("insurance.psdd"));
rc = load_logistic_circuit(zoo_lc_file("insurance.circuit"), 1);
```

Is the predictive model biased by gender?

```julia
groups = make_observations([["male"], ["female"]])
exps, _ = Expectation(pc, rc, groups);
println("Female  : \$ $(exps[2])");
println("Male    : \$ $(exps[1])");
println("Diff    : \$ $(exps[2] - exps[1])");
Female  : $ 14170.125469335406
Male    : $ 13196.548926381849
Diff    : $ 973.5765429535568
```

Outline

1. Theoretical motivation
   Tractability of SHAP explanations [AAAI'21]
2. Tractable reasoning about classifier behavior
   Reasoning about missing features [NeurIPS'19]
   Latent fair decisions [AAAI'21]
   with YooJung Choi, Meihua Dang

Model-Based Algorithmic Fairness: FairPC

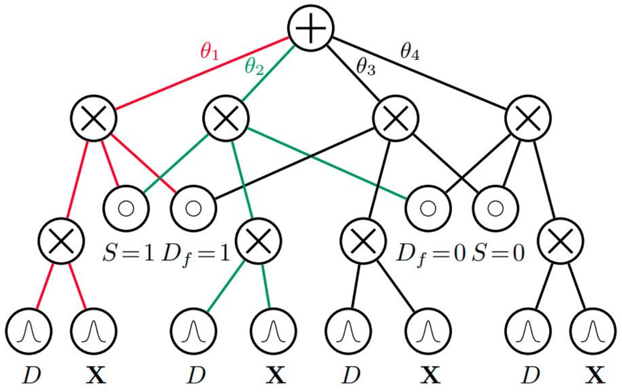
Learn classifier given

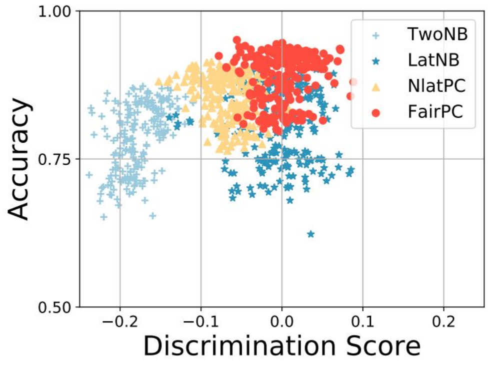
- features S and X
- training labels/decisions D

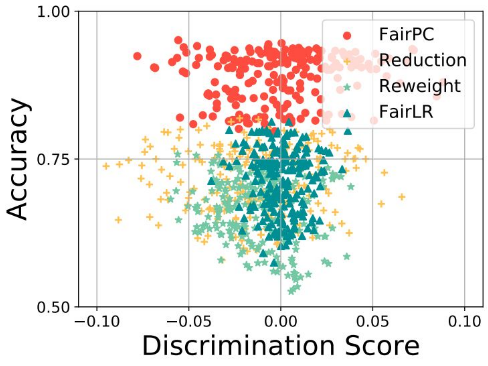
Group fairness by demographic parity:

Fair decision D_f should be independent of the sensitive attribute S

Discover the **latent fair decision** D_f by learning a PC.







[Choi et al. AAAI21]

Outline

1. Theoretical motivation
   Tractability of SHAP explanations [AAAI’21]
2. Tractable reasoning about classifier behavior
   Reasoning about missing features [NeurIPS’19]
   Latent fair decisions [AAAI’21]
   Probabilistic sufficient explanations [IJCAI’21]
   with Eric Wang, Pasha Khosravi

Local Explanation Approaches

Model agnostic: (e.g. LIME, SHAP, Anchors)

- Treat classifier as black box
- Evaluate on sampled perturbations
  - Often ignores feature distribution (in favor of perturbation distribution)
  - Evaluate on impossible or low likelihood instances
- Can be fooled! [Slack et al., 2020; Dimanov et al., 2020]
- Might produce over-confident results [Ignatiev et al.]
- Very generally applicable
- No guarantees

Local Explanation Approaches

Logical explanations:
(e.g, sufficient reasons, abduction, prime implicants)

- Give formal guarantees with 100% certainty
- Ensure minimality
- Hard to compute
  (e.g., reduce MNIST from 784 to 64 pixels)
- Ignores feature distribution (it is irrelevant!)
- Lead to complex explanations
  To give a guarantee with 100% certainty,
  one needs to know almost all of the pixels...

Probabilistic Sufficient Explanations

Explanation is a subset of features, s.t.

1. The explanation is “probabilistically sufficient”

   Under the feature distribution, given the explanation, the classifier is likely to make the observed prediction.

   => Strong probabilistic guarantees

2. It is minimal and “simple”

[Khosravi et al. IJCAI19, Wang et al. IJCAI’21]

Probabilistic Sufficiency Criteria

Same Decision Probability

$$\mathsf{SDP}_{\mathcal{C},\mathbf{x}}(\mathbf{z}) = \mathop{\mathbb{E}}_{\mathbf{m}\sim\Pr(\mathbf{M}|\mathbf{z})} [\![\mathcal{C}(\mathbf{zm}) = \mathcal{C}(\mathbf{x})]\!]$$

C(.) is a threshold-based classifier, output is + or -

Similar criteria used in **Anchors**.

Logical Reasoning approaches require SDP=1.

Hard to calculate (PP^PP-hard on Bayesian networks, NP-hard on Naive Bayes)

Probabilistic Sufficiency Criteria

Expected Prediction

$$\mathsf{EP}(\mathbf{z}) = \mathop{\mathbb{E}}_{\mathbf{m} \sim \Pr(\mathbf{M}|\mathbf{z})} f(\mathbf{zm})$$

f(.) is a probabilistic classifier, output is a class probability or log-probability

Expected Prediction is tractable to compute for

1. Logistic regression with conformant Naive Bayes
2. Decision trees w.r.t. PC
3. Discriminative circuits w.r.t. PC
4. Feature distribution and classifier defined by same PC

Probabilistic Sufficient Explanation

Want to maximize the **expected prediction** while keeping explanations **simple**.

$$\mathsf{SE}_k(\mathbf{x}) = \operatorname*{argmax}_{\mathbf{z} \subseteq \mathbf{x}} \mathsf{EP}(\mathbf{z}) \quad \text{s.t. } |\mathbf{z}| \leq k$$

To also achieve **minimality**, we can choose the most likely ones.

$$\mathsf{MLSE}_k(\mathbf{x}) = \operatorname*{argmax}_{\mathbf{z} \in \mathsf{SE}_k(\mathbf{x})} \Pr(\mathbf{z})$$

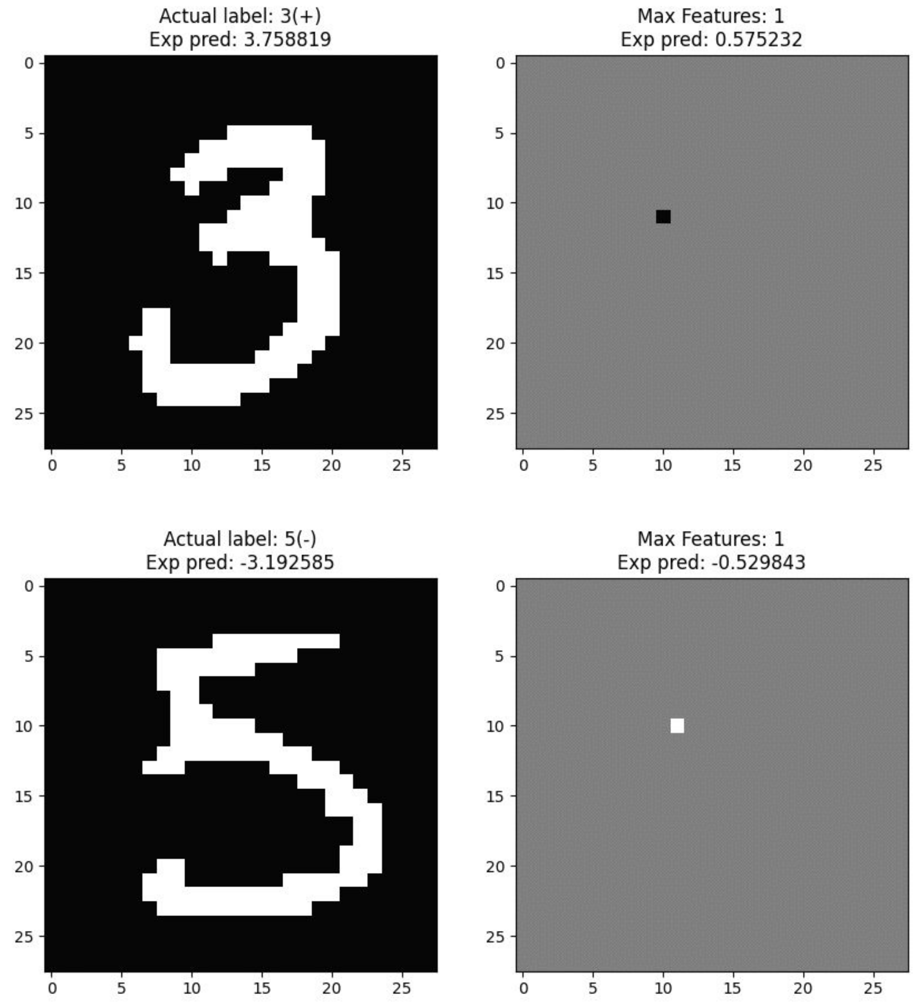
Correctly Classified Examples

Binary classification: 3 vs 5

Used decision forest classifier and probabilistic circuit feature distribution

Beam search - keep top b explanation candidates for each size

Sort by expected prediction, break ties by feature probability



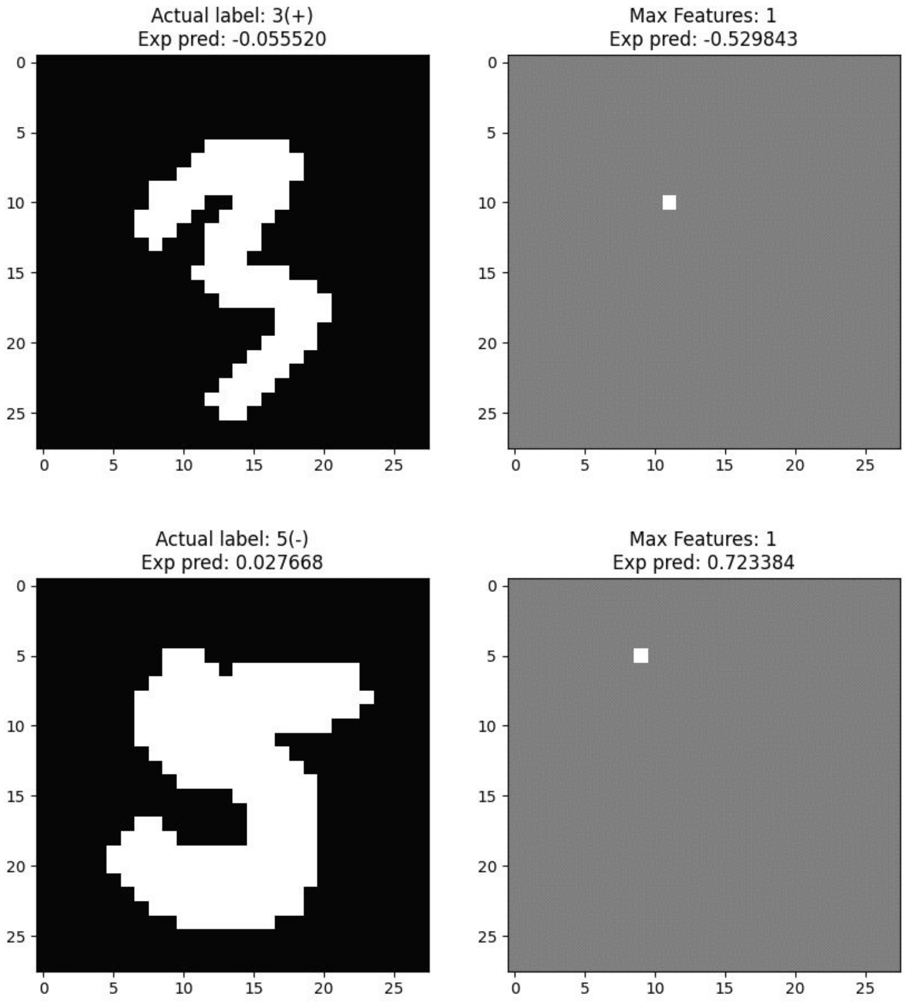
Misclassified Examples

Binary classification: 3 vs 5

Used decision forest classifier and probabilistic circuit feature distribution

Beam search - keep top b explanation candidates for each size

Sort by expected prediction, break ties by feature probability

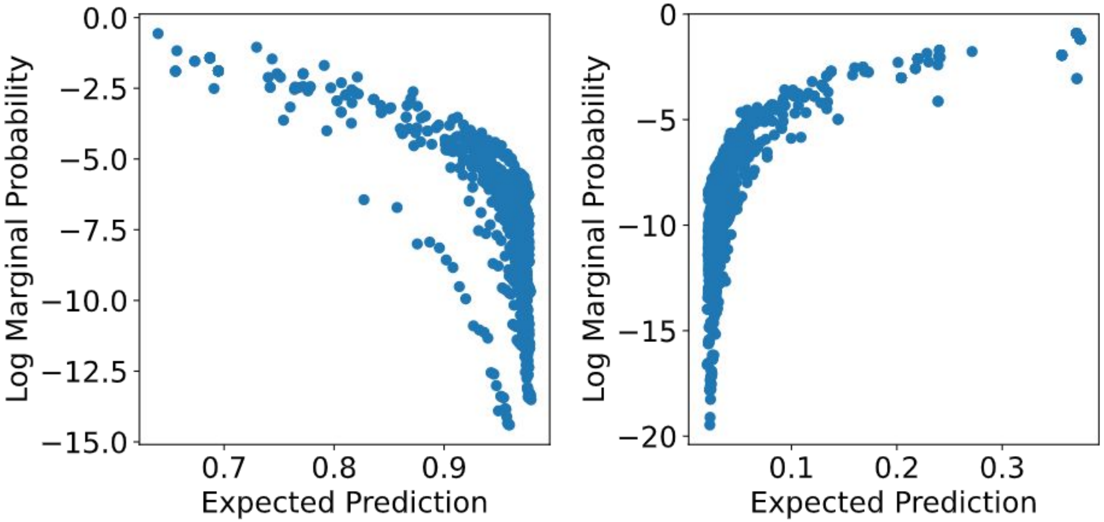


Comparison with Anchors on MNIST (784 pixels)

Method	$\lvert \mathrm{EP}_{\mathcal{O}}(\mathbf{z}) \rvert$	$\mathrm{SDP}_{\mathcal{C},\mathbf{x}}(\mathbf{z})$	$\log P(\mathbf{z})$
Anchors	0.75 ± 0.37	0.66 ± 0.08	-3.29 ± 0.88
MLSE_s	1.57 ± 0.29	0.86 ± 0.05	-3.05 ± 0.65
MLSE_{10}	3.11 ± 0.23	0.99 ± 0.01	-6.98 ± 1.37
MLSE_{20}	3.60 ± 0.15	1.00 ± 0.00	-9.90 ± 2.14
MLSE_{30}	3.75 ± 0.13	1.00 ± 0.00	-11.77 ± 2.88

- For same size of explanation
  - SE has more realistic explanations (higher marginal likelihood P(z))
  - SE has stronger guarantees (higher expected log-odds and SDP)
- For >10 pixels
  - SE are almost logical explanations (around 100% SDP)
  - Yet they remain \`simple\`: small and with high marginal likelihood

Explanation Complexity vs Sufficiency Constraints

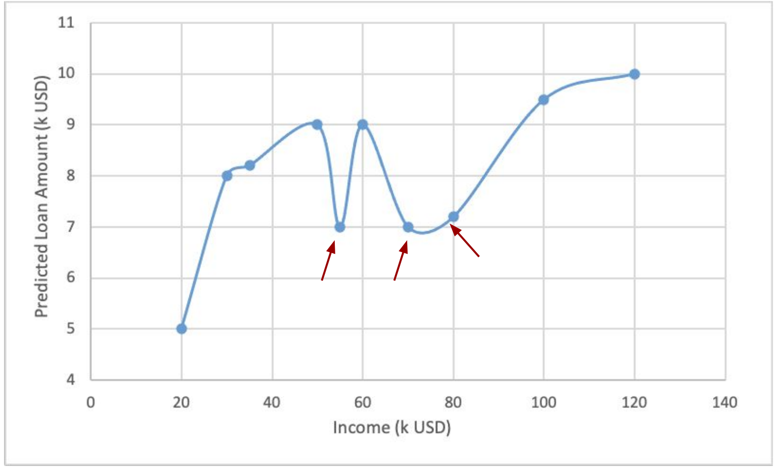


- Simple explanations (high likelihood) give strong probabilistic guarantees (EP)
- Steep around 0 and 1: making guarantees even slightly probabilistic will lead to significant simplification of the explanations

Outline

1. Theoretical motivation
   Tractability of SHAP explanations [AAAI'21]
2. Tractable reasoning about classifier behavior
   Reasoning about missing features [NeurIPS'19]
   Latent fair decisions [AAAI'21]
   Probabilistic sufficient explanations [IJCAI'21]
3. Practical neuro-symbolic verification
   Learning monotonic neural networks [NeurIPS'20]
   with Aishwarya Sivaraman, Golnoosh Farnadi, Todd Millstein

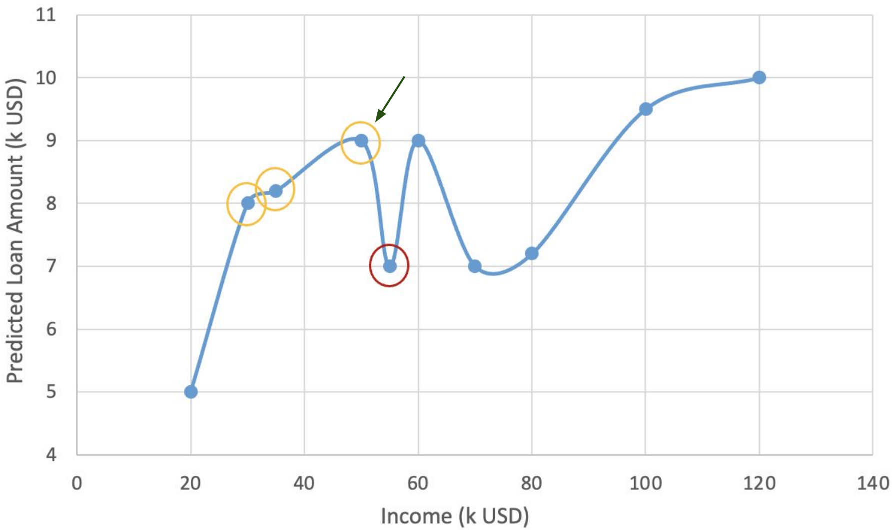
Predict Loan Amount



Neural Network Model: Increasing income can decrease the approved loan amount

Monotonicity (Prior Knowledge):
Increasing income should increase the approved loan amount

Counterexamples

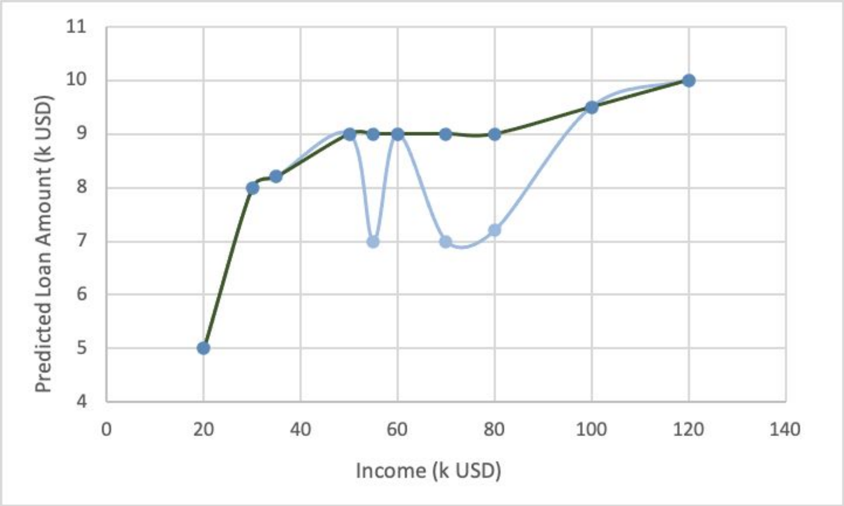


$$\exists x, y \; x \leq y \implies f(x) > f(y)$$

Computed using SMT(LRA)

Maximal counterexamples (largest violation) using OMT

Counterexample-Guided Predictions



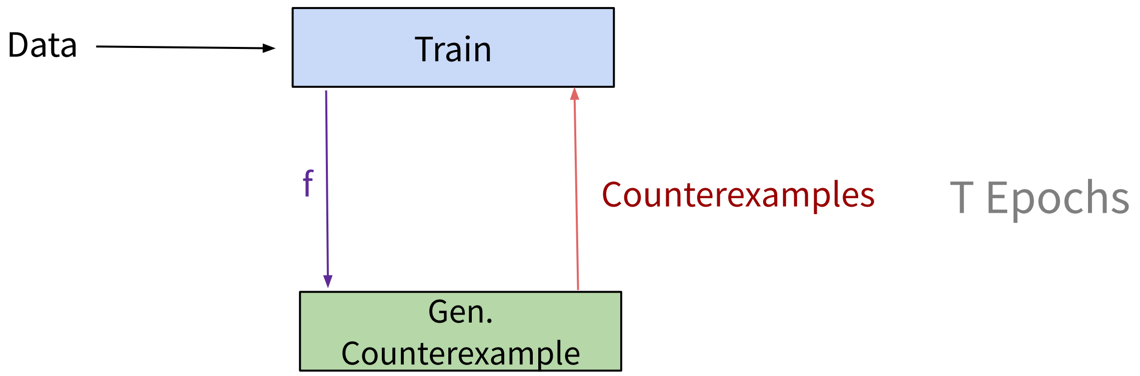
Monotonic Envelope:

- Replace each prediction by its maximal counterexample
- Envelope construction is online (during prediction)
- Guarantees monotonic predictions for any ReLU neural net
- Works for high-dimensional input
- Works for multiple monotonic features

Counterexample-Guided Learning

How to use monotonicity to improve model quality?

“Monotonicity as inductive bias”



Counterexample-Guided Monotonicity Enforced Training (COMET)

Table 4: Monotonicity is an effective inductive bias. COMET outperforms Min-Max networks on all datasets. COMET outperforms DLN in regression datasets and achieves similar results in classification datasets.

Dataset	Features	Min-Max	DLN	COMET
Auto-MPG	Weight	9.91±1.20	16.77±2.57	**8.92±2.93**
	Displ.	11.78±2.20	16.67±2.25	**9.11±2.25**
	W,D	11.60±0.54	16.56±2.27	**8.89±2.29**
	W,D,HP	10.14±1.54	13.34±2.42	**8.81±1.81**
Boston	Rooms	30.88±13.78	15.93±1.40	**11.54±2.55**
	Crime	25.89±2.47	12.06±1.44	**11.07±2.99**

Dataset	Features	Min-Max	DLN	COMET
Heart	Trestbps	0.75±0.04	0.85±0.02	**0.86±0.03**
	Chol.	0.75±0.04	0.85±0.04	**0.87±0.03**
	T,C	0.75±0.04	**0.86±0.02**	**0.86±0.03**
Adult	Cap. Gain	0.77	**0.84**	**0.84**
	Hours	0.73	**0.85**	0.84

Our Contributions

- Counterexample-guided algorithm that guarantees monotonicity at prediction time for an arbitrary ReLU neural network
- Counterexample-guided algorithm to incorporate monotonicity as an inductive bias during training
- Outperforms state-of-the-art monotonic learners in regression and classification tasks
- Counterexample-guided learning when used in conjunction with envelope **improves accuracy and provides provable guarantees**

Pure Logic **Probabilistic World Models** **Pure Learning**

A New Synthesis of Learning and Reasoning

"Pure learning is brittle"

bias, **algorithmic fairness**, interpretability, **explainability**, adversarial attacks, unknown unknowns, calibration, **verification**, **missing features**, missing labels, data efficiency, shift in distribution, general robustness and safety

We need to incorporate a sensible probabilistic/logic model of the world

Thanks

This was the work of many wonderful students/postdoc/collaborators!

References: http://starai.cs.ucla.edu/publications/