The **next** circuits for a better life

Discriminative Bias for Learning Probabilistic Sentential Decision Diagrams

Laura I. Galindez Olascoaga[*], Wannes Meert [*], Nimish Shah [*], Guy Van den Broeck[†], Marian Verhelst [*]

[*] KU LEUVEN micas KU LEUVEN

[†] UCLA STAR AI RESEARCH LAB UCLA

Outline

- **Motivation and objective**
- Background
- Discriminative bias for learning PSDDs
- Experimental results
- Conclusions

Motivation

Probabilistic inference has proven to be well suited for **resource-constrained embedded applications**.

(Galindez et al. 2019)

Probabilistic circuits successfully **balance efficiency vs. expressiveness** trade-offs while remaining robust.

Some of these models' robustness (from generative learning) is **at odds with discriminative performance.**

Objective

Keep robustness provided by generative learning strategies.

Improve discriminative performance by exploiting knowledge encoding capabilities.

Outline

- Motivation and objective
- **Background**
- Discriminative bias for learning PSDDs
- Experimental results
- Conclusions

Background: probabilistic inference

Given a probabilistic model **m** of the world

Answer probabilistic queries

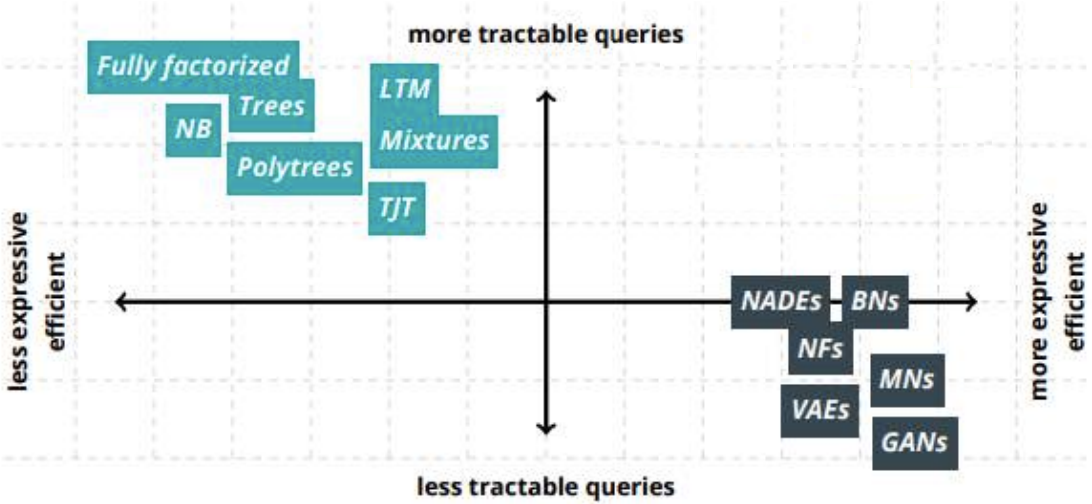
Evidence $q_1(\mathbf{m})=Pr_{\mathbf{m}}(\quad)$

Conditional $q_2(\mathbf{m})=Pr_{\mathbf{m}}(\quad|\quad)$

MAP $q_3(\mathbf{m})=Argmax_{time}Pr_{\mathbf{m}}(\quad)$

Background: tractable probabilistic inference

A query q(**m**) is **tractable** iff **exactly** computing it runs in time **O(poly(|m|).**

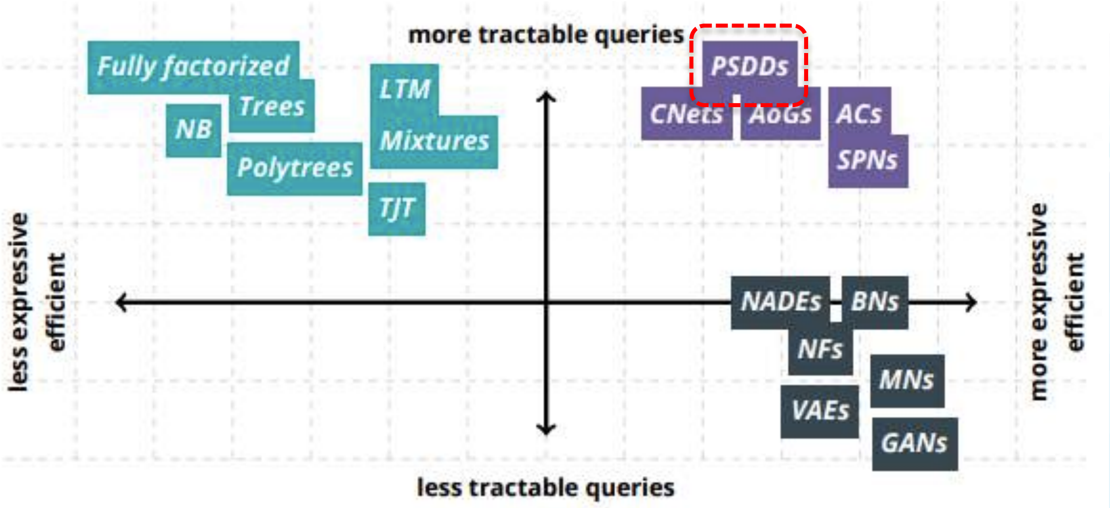


There is an inherent trade-off between tractability and expressiveness

(From UAI 2019 tutorial on Tractable Probabilistic Models by Vergari, Di Mauro and Van den Broeck and AAAI 2020 tutorial on Probabilistic Circuits by Vergari, Choi, Peharz and Van den Broeck)

Background: probabilistic circuits

A **probabilistic circuit** is a computational graph that encodes a probability distribution p(X).

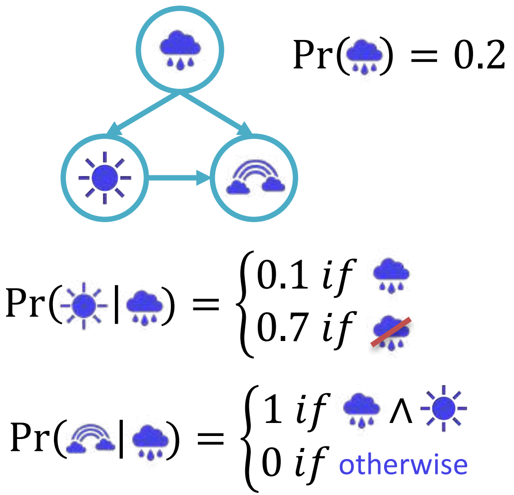


(From UAI 2019 tutorial on Tractable Probabilistic Models by Vergari, Di Mauro and Van den Broeck and AAAI 2020 tutorial on Probabilistic Circuits by Vergari, Choi, Peharz and Van den Broeck)

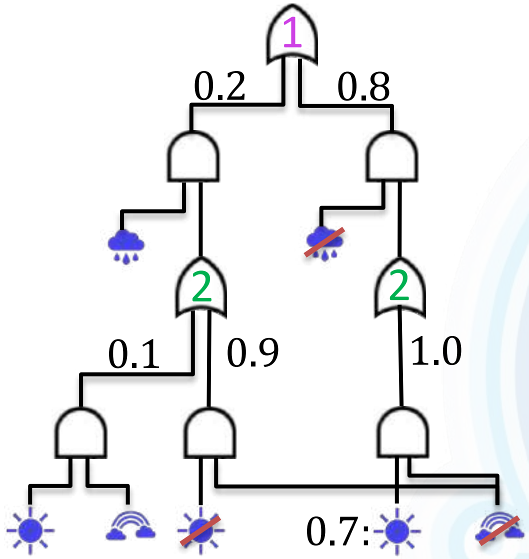
Background: what is a PSDD?

- PSDDs are probabilistic extensions to SDDs, which represent Boolean functions as logical circuits (Kisa et al., 2014).

Bayesian Network



PSDD

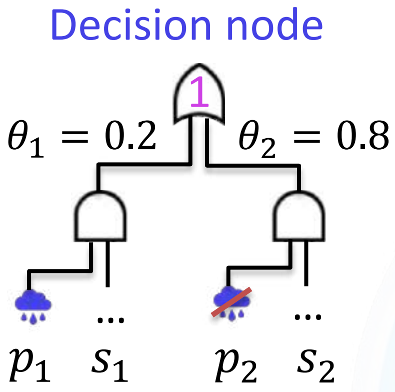


(Example from Liang et al., 2017)

Background: PSDDs' properties

The left variable of the AND gate is the prime (p) and the right is the sub (s).

Edges of decision nodes are annotated with a normalized probability distribution.



Background: PSDDs' properties

Syntactic restrictions: See (Kisa et al., 2014).

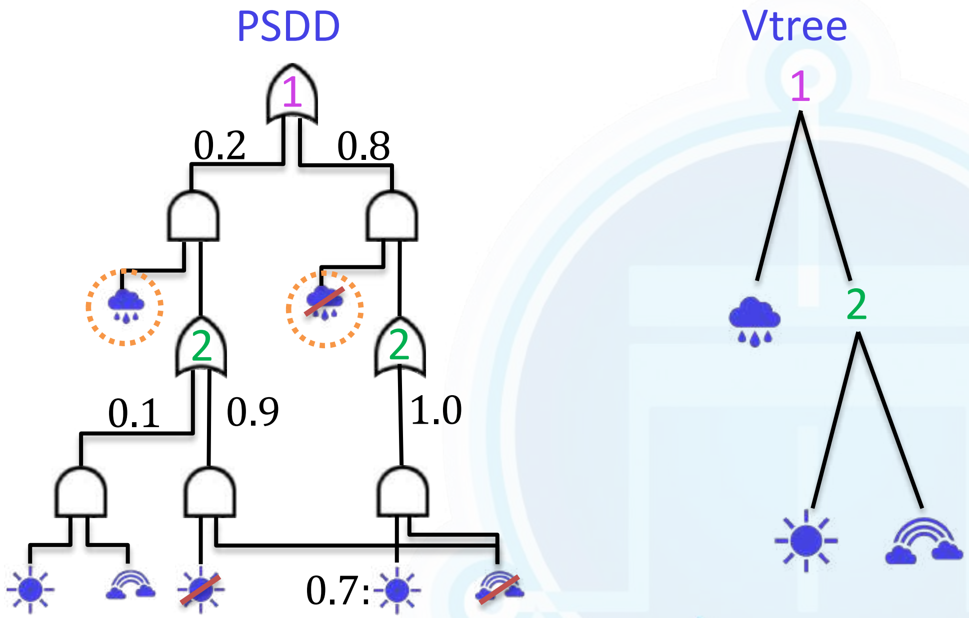
1) *Decomposability*: inputs of AND node must be disjoint.

For example at 1:

Prime variables $\boldsymbol{X} = \{Rain\}$

Sub variables $\boldsymbol{Y} = \{Sun, Rbow\}$

2) *Determinism*: only one of the decision node's inputs can be true.



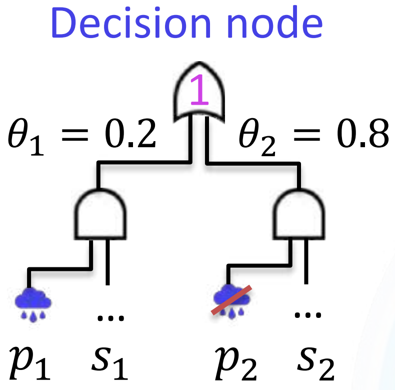
Background: PSDDs' properties

Decision nodes q encode the distribution:

$$Pr_q(\boldsymbol{XY}) = \sum_i \theta_i \boxed{Pr_{pi}(\boldsymbol{X}) Pr_{si}(\boldsymbol{Y})}$$

$$Pr_n(\boldsymbol{XY}|[p_i]) = Pr_{p_i}(\boldsymbol{X}|[p_i]) Pr_{s_i}(\boldsymbol{Y}|[p_i])$$
$$= Pr_{p_i}(\boldsymbol{X}) Pr_{s_i}(\boldsymbol{Y})$$

A logical sentence that defines the support of node distribution

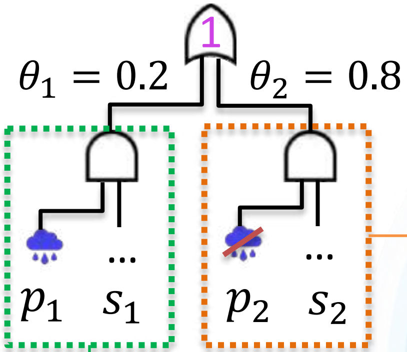


Background: PSDDs' properties

Decision nodes q encode the distribution:

$$Pr_q(\boldsymbol{X}\boldsymbol{Y}) = \sum_i \theta_i Pr_{pi}(\boldsymbol{X}) Pr_{si}(\boldsymbol{Y})$$

Decision node



$$Pr_1(\boldsymbol{X}\boldsymbol{Y}) = \boxed{0.2 \cdot Pr_{p_1}(\boldsymbol{X}) Pr_{s_1}(\boldsymbol{Y})} + \boxed{0.8 \cdot Pr_{p_2}(\boldsymbol{X}) Pr_{s_2}(\boldsymbol{Y})}$$

$$\boxed{0.8 \cdot Pr_{p_1}(\boldsymbol{X}|[\;]) Pr_{s_1}(\boldsymbol{Y}|[\;])}$$

$$\boxed{0.2 \cdot Pr_{p_1}(\boldsymbol{X}|[\;]) Pr_{s_1}(\boldsymbol{Y}|[\;])}$$

For example at 1:

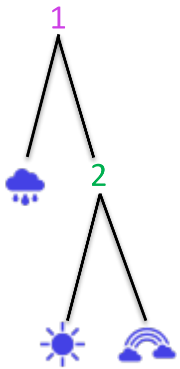
Prime variables $\boldsymbol{X} = \{Rain\}$

Sub variables $\boldsymbol{Y} = \{Sun, Rbow\}$

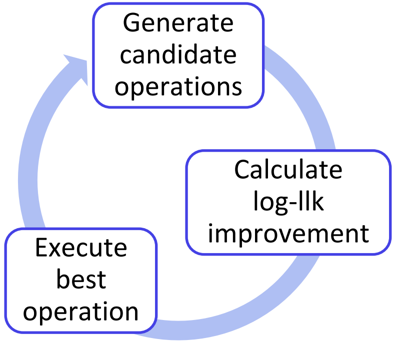
Background: learning PSDDs

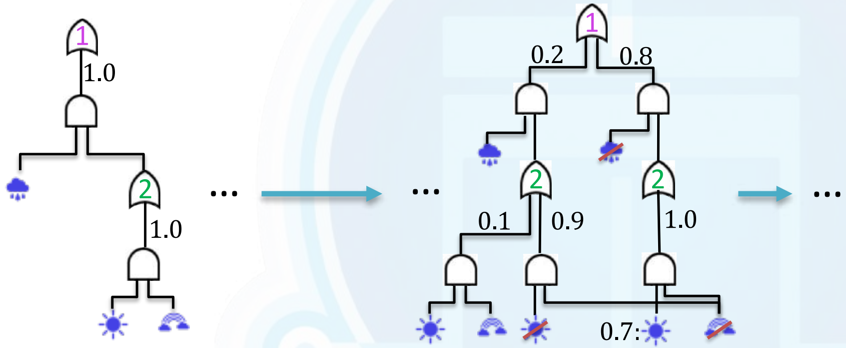
- The LearnPSDD algorithm (Liang et al., 2017) learns the PSDD structure incrementally from data.

Learn vtree from data (Minimize mutual information)



Iteratively apply *split* and *clone* operations





Outline

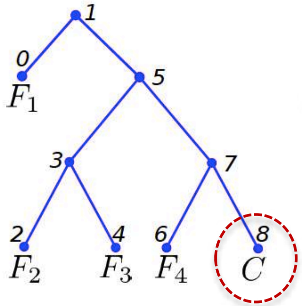
- Motivation and objective
- Background
- **Discriminative bias for learning PSDDs**
- Experimental results
- Conclusions

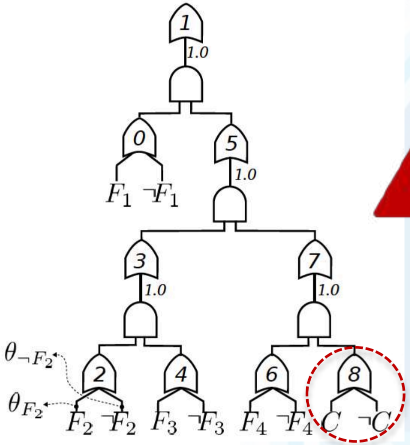
Classification with PSDDs

- Given a feature variable set $\boldsymbol{F}$ and a class variable C.
- The classification task can be stated as a probabilistic query:

$$\Pr(C|\boldsymbol{F}) \sim \Pr(\boldsymbol{F}|C) \cdot \Pr(C)$$

LearnPSDD remains agnostic to the classification task



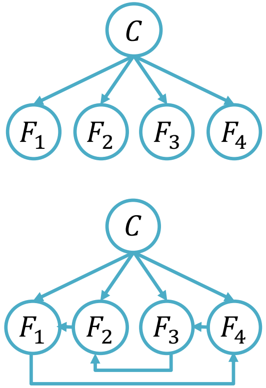


With **LearnPSDD** features might never be conditioned on the class

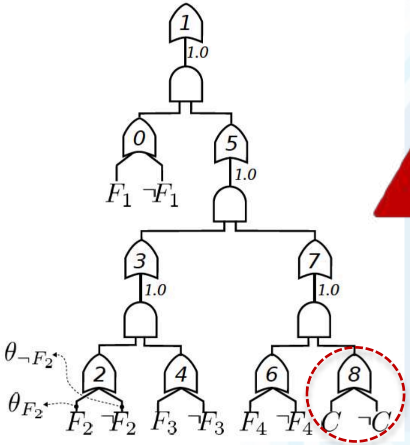
Bayesian Network classifiers

- Effects of explicitly conditioning $\boldsymbol{F}$ on C.

$$\Pr(C|\boldsymbol{F}) \sim \Pr(\boldsymbol{F}|C) \cdot \Pr(C)$$



With **Bayesian Network classifiers** features are always conditioned on the class.

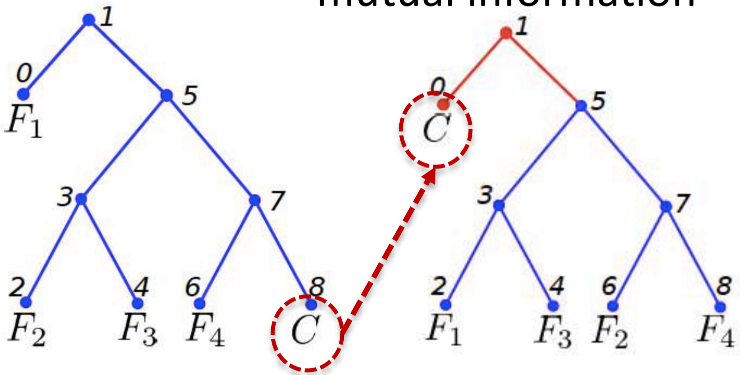


With **LearnPSDD** features might never be conditioned on the class.

Enforcing the discriminative bias: D-LearnPSDD

- Make sure that feature variables $\boldsymbol{F}$ can be conditioned on the class variable C.

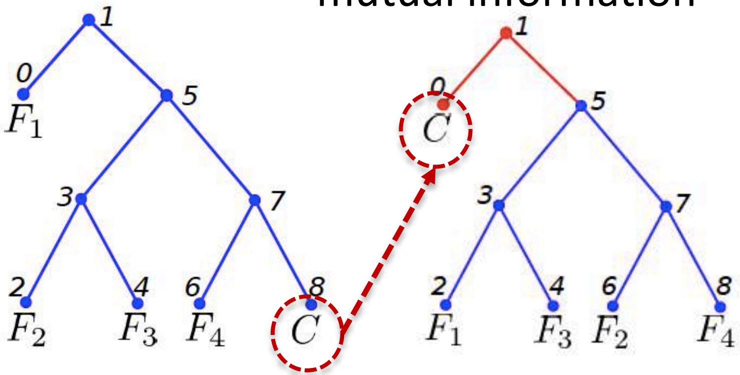
Minimize conditional mutual information



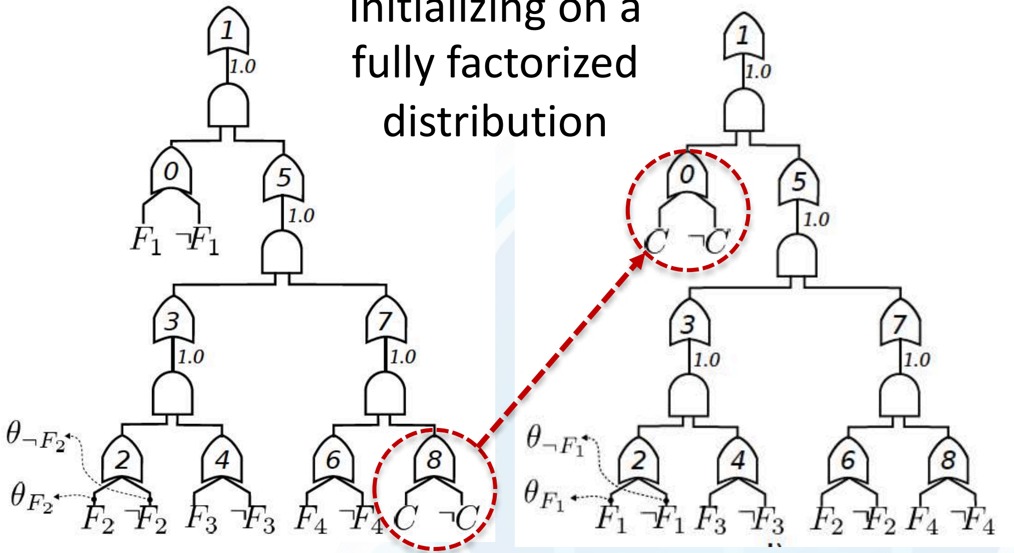
Enforcing the discriminative bias: D-LearnPSDD

- Make sure that feature variables $\boldsymbol{F}$ can be conditioned on the class variable C.

Minimize conditional mutual information

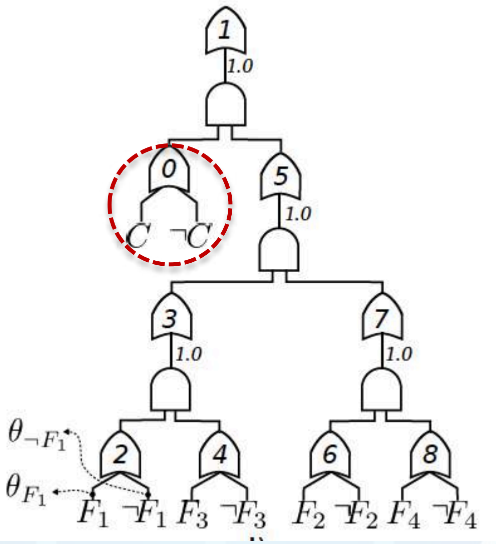


Initializing on a fully factorized distribution



Enforcing the discriminative bias: D-LearnPSDD

- Make sure that feature variables $\boldsymbol{F}$ can be conditioned on the class variable C.
  - However, only setting the vtree is not enough.

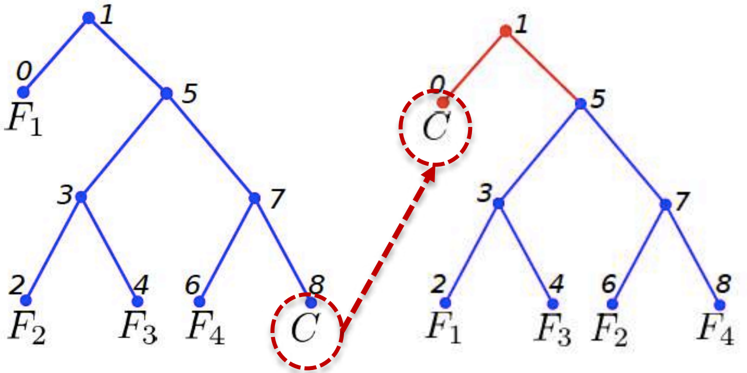


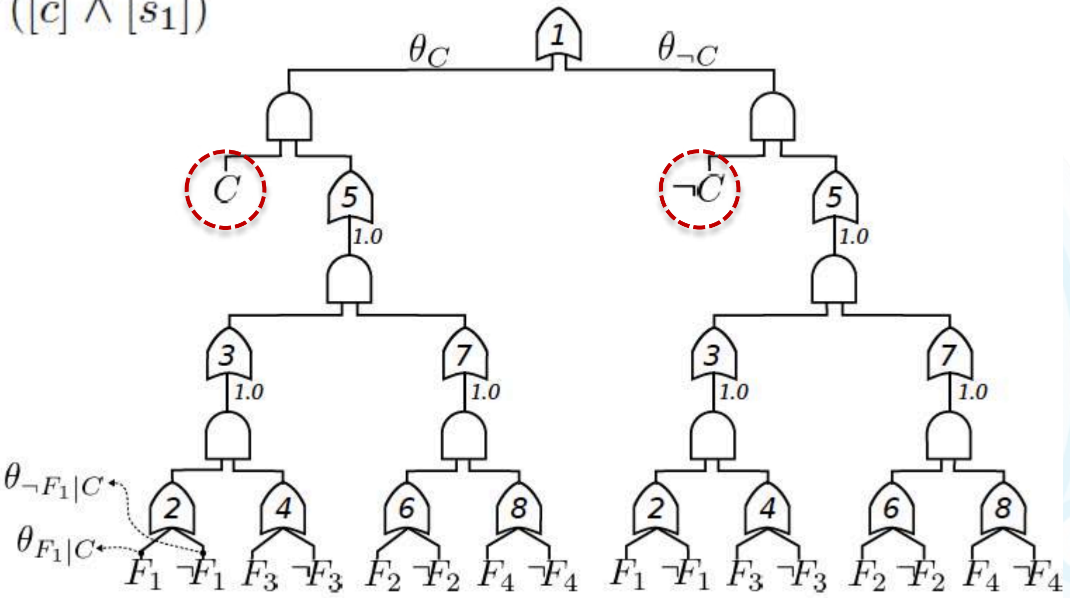
$$
\begin{aligned}
\Pr_q(C\mathbf{F}) &= \Pr_{p_0}(C|[c \lor \neg c]) \cdot \Pr_{s_0}(\mathbf{F}|[c \lor \neg c]) \\
&= (\Pr_{p_1}(C|[c]) + \Pr_{p_2}(C|[\neg c])) \cdot \Pr_{s_0}(\mathbf{F}|[c \lor \neg c]) \\
&= \boxed{(\Pr_{p_1}(C=1) + \Pr_{p_2}(C=0)) \cdot \Pr_{s_0}(\mathbf{F})}
\end{aligned}
$$

⚠ $\boldsymbol{F}$ still independent from C

Enforcing the discriminative bias: D-LearnPSDD

- Make sure that feature variables $\boldsymbol{F}$ can be conditioned on the class variable C.
  - Set $[root] = ([\neg c] \wedge [s_0]) \vee ([c] \wedge [s_1])$





Enforcing the discriminative bias: D-LearnPSDD

- Make sure that feature variables $\boldsymbol{F}$ can be conditioned on the class variable C.
  - Set $[root] = ([\neg c] \wedge [s_0]) \vee ([c] \wedge [s_1])$
  - LearnPSDD ensures that the base of the root node remains unchanged.

$$\begin{aligned}
\Pr_q(C\mathbf{F}) &= \Pr_{\neg c}(C)\Pr_{s_0}(\mathbf{F}) + \Pr_c(C)\Pr_{s_1}(\mathbf{F}) \\
&= \Pr_{\neg c}(C|[\neg c]) \cdot \Pr_{s_0}(\mathbf{F}|[\neg c]) + \Pr_c(C|[c]) \cdot \Pr_{s_1}(\mathbf{F}|[c]) \\
&\boxed{= \Pr_{\neg c}(C=0) \cdot \Pr_{s_0}(\mathbf{F}|C=0) + \Pr_c(C=1) \cdot \Pr_{s_1}(\mathbf{F}|C=1)}
\end{aligned}$$

Encodes a naive Bayes structure

Outline

- Motivation and objective
- Background
- Discriminative bias for learning PSDDs
- **Experimental results**
- Conclusions

Experimental results

Dataset	D-LearnPSDD		LearnPSDD		NB		TANB		LogReg
	Accuracy	Size	Accuracy	Size	Accuracy	Size	Accuracy	Size	Accuracy
Australian									
Breast									
Chess									
Cleve									
Corral 6									
Credit									
Diabetes									
German									
Glass									
Heart									
Iris									
Mofn									
Pima									
Vehicle									
Waveform									

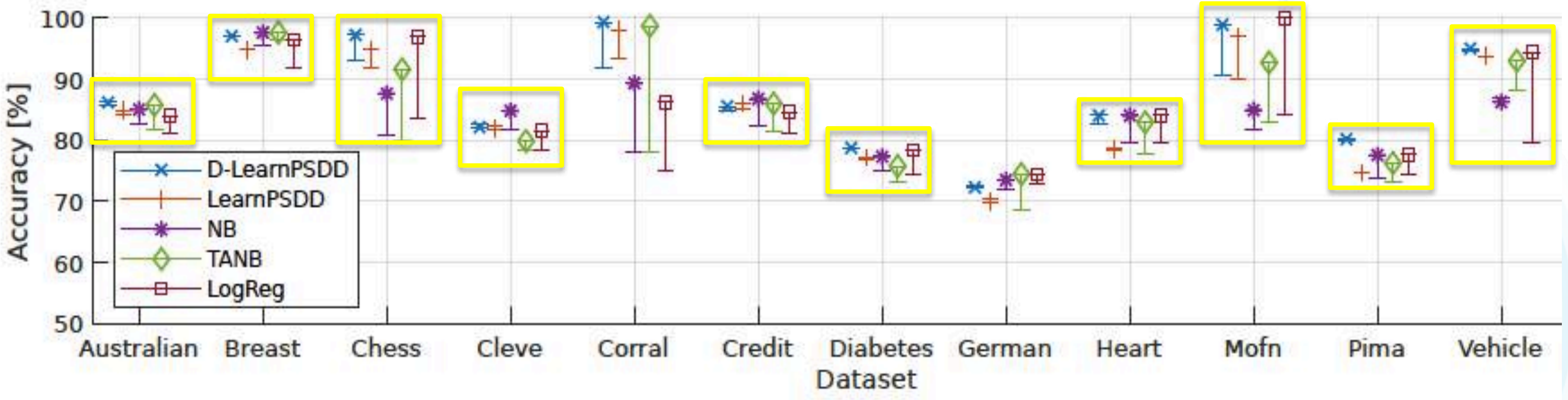
- 15 UCI datasets
- 5-fold cross validation
- Average accuracy over a range of model size
- Model size is number of parameters

Experimental results

Dataset	D-LearnPSDD		LearnPSDD		NB		TANB		LogReg
	Accuracy	Size	Accuracy	Size	Accuracy	Size	Accuracy	Size	Accuracy
Australian	**86.2 ± 3.6**	367	84.9 ± 2.7	386	85.1 ± 3.1	161	85.8 ± 3.4	312	84.1 ± 3.4
Breast	97.1 ± 0.9	291	94.9 ± 0.5	491	**97.7 ± 1.2**	114	97.7 ± 1.2	219	96.5 ± 1.6
Chess	**97.3 ± 1.4**	2178	94.9 ± 1.6	2186	87.7 ± 1.4	158	91.7 ± 2.2	309	96.9 ± 0.7
Cleve	82.2 ± 2.5	292	81.9 ± 3.2	184	**84.9 ± 3.3**	102	79.9 ± 2.2	196	81.5 ± 2.9
Corral 6	**99.4 ± 1.4**	39	98.1 ± 2.8	58	89.4 ± 5.2	26	98.8 ± 1.7	45	86.3 ± 6.7
Credit	85.6 ± 3.1	693	86.1 ± 3.6	611	**86.8 ± 4.4**	170	86.1 ± 3.9	326	84.7 ± 4.9
Diabetes	**78.7 ± 2.9**	124	77.2 ± 3.3	144	77.4 ± 2.56	46	75.8 ± 3.5	86	78.4 ± 2.6
German	72.3 ± 3.2	1185	69.9 ± 2.3	645	73.5 ± 2.7	218	**74.5 ± 1.9**	429	74.4 ± 2.3
Glass	**79.1 ± 1.9**	214	72.4 ± 6.2	321	70.0 ± 4.9	203	69.5 ± 5.2	318	73.0 ± 5.7
Heart	**84.1 ± 4.3**	51	78.5 ± 5.3	75	84.0 ± 3.8	38	83.0 ± 5.1	70	84.0 ± 4.7
Iris	90.0 ± 0.1	76	94.0 ± 3.7	158	**94.7 ± 1.8**	75	94.7 ± 1.8	131	94.7 ± 2.9
Mofn	98.9 ± 0.9	260	97.1 ± 2.4	260	85.0 ± 5.7	42	92.8 ± 2.6	78	**100.0 ± 0**
Pima	**80.2 ± 0.3**	108	74.7 ± 3.2	110	77.6 ± 3.0	46	76.3 ± 2.9	86	77.7 ± 2.9
Vehicle	**95.0 ± 1.7**	1186	93.9 ± 1.69	1560	86.3 ± 2.00	228	93.0 ± 0.8	442	94.5 ± 2.4
Waveform	85.0 ± 1.0	3441	78.7 ± 5.6	2585	80.7 ± 1.9	657	83.1 ± 1.1	1296	**85.5 ± 0.7**

Experimental results

- D-LearnPSDD remains robust against missing features.



Outline

- Motivation and objective
- Background
- Discriminative bias for learning PSDDs
- Experimental results
- **Conclusions**

Conclusions

- We introduced a PSDD learning technique that improves classification performance by introducing a discriminative bias.
- Robustness is ensured by exploiting the generative learning strategy.
- The proposed technique outperforms purely generative PSDDs in terms of classification accuracy and the other baseline classifiers in terms of robustness.

References

- Laura I. Galindez Olascoaga, Wannes Meert, Nimish Shah, Marian Verhelst and Guy Van den Broeck. Towards Hardware-Aware Tractable Learning of Probabilistic Models, *In Advances in Neural Information Processing Systems 32 (NeurIPS)*, 2019.
- YooJung Choi, Antonio Vergari, Robert Peharz and Guy Van den Broeck. Probabilistic Circuits: Representation and Inference, AAAI tutorial, 2020.
- 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.
- 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.

Thank you!

Contact:

laura.galindez@esat.kuleuven.be