FACULTAD DE INGENIERÍA Y CIENCIAS

UNIVERSIDAD ADOLFO IBÁÑEZ

SMART + SUSTAINABLE

Score-Based Explanations for Classification Results

Leopoldo Bertossi

leopoldo.bertossi@uai.cl

Explanations in Databases

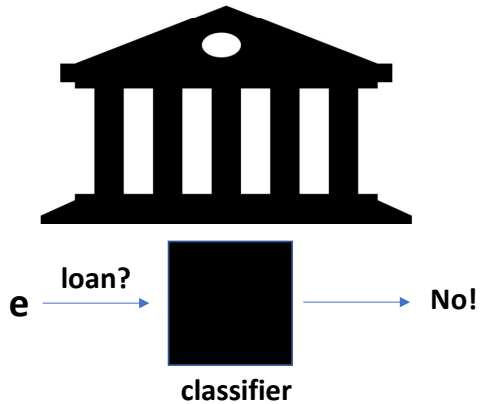
Receives	$R.1$	$R.2$
	s_2	s_1
	s_3	s_3
	s_4	s_3

Store	$S.1$
	s_2
	s_3
	s_4

- Query: Are there pairs of official stores in a receiving relationship?
- $\mathcal{Q}$: $\exists x \exists y (Store(x) \wedge Receives(x, y) \wedge Store(y))$

  The query is true in D: $D \models \mathcal{Q}$
- What tuples cause the query to be true?
- How strong are they as causes?
- We would expect tuples $Receives(s_3, s_3)$ and $Receives(s_4, s_3)$ to be causes
- Explanations for a query result ...

Explanations in Machine Learning -1



- Client requesting a loan from a bank using a black-box classifier
- $\mathbf{e} = \langle \text{john}, 18, \text{plumber}, 70\text{K}, \text{harlem} \rangle$

  Record of values for features Name, Age, Income, ...
- Which are the feature values most relevant for the classification outcome, i.e. the label "No"?
- What is the contribution of each feature value to the outcome?
- Questions like these are at the core of Explainable AI

A Score-Based Approach: Responsibility -1

- Causality has been developed in AI for 3 decades or so
- In particular, Actual Causality
- Also the quantitative notion of Responsibility: a measure of causal contribution
- Both based on Counterfactual Interventions
- Hypothetical changes of values in a causal model to detect other changes
  By so doing identify actual causes
- Do deletions of certain database tuples make the query false?
- Do changes of feature values make the label change to "Yes"?
- We have investigated causality and responsibility in data management and classification
- Semantics, computational mechanisms, intrinsic complexity, logic-based specifications, reasoning, etc.

A Score-Based Approach: Responsibility -2

For the gist: $\exists x \exists y (Store(x) \wedge Receives(x, y) \wedge Store(y))$

Receives	*R*.1	*R*.2
	s_2	s_1
	~~s_3~~	~~s_3~~
	s_4	s_3

Store	*S*.1
	s_2
	s_3
	~~s_4~~

$D' \not\models Q$

- $Receives(s_3, s_3)$ is actual cause, with $\{Store(s_4)\}$ as minimum-size contingency set
- $Resp(Receives(s_3, s_3)) := \frac{1}{1 + |\{Store(s_4)\}|} = \frac{1}{2}$
- $Resp(Store(s_3)) := \frac{1}{1 + 0} = 1$ a counterfactual cause

- $\mathbf{e} = \langle \text{john}, 18, \text{plumber}, 70\text{K}, \text{harlem}, \ldots \rangle$ No
  $\mathbf{e}' = \langle \text{john}, 25, \text{plumber}, 70\text{K}, \text{harlem}, \ldots \rangle$ Yes
  $\mathbf{e}'' = \langle \text{john}, 18, \text{plumber}, 80\text{K}, \text{brooklyn}, \ldots \rangle$ Yes
- Value for Age is counterfactual cause with $x\text{-}Resp(\text{Age}) = 1$
  Value for Income is actual cause with $x\text{-}Resp(\text{Income}) = \frac{1}{2}$
- Second may be actionable, but not the first

A Score-Based Approach: Shapley Values -1

- Database tuples and feature values can be seen as players in a coalition game
  Each of them contributing to a shared wealth function
- The Shapley value is a established measure of contribution by players to the wealth function
- It emerges as the only measure that enjoys certain desired properties
- For each game one defines an appropriate wealth or game function
- In the case of $\mathcal{Q}$: $\exists x \exists y (Store(x) \wedge Receives(x, y) \wedge Store(y))$, the game function can be the value of the query, i.e. 1 or 0
- A set of tuples make it true or not, with some maybe contributing more than others to making it true

A Score-Based Approach: Shapley Values -2

$$Shapley(D, \mathcal{Q}, \tau) := \sum_{S \subseteq D \setminus \{\tau\}} \frac{|S|!(|D|-|S|-1)!}{|D|!}(\mathcal{Q}(S \cup \{\tau\}) - \mathcal{Q}(S))$$

- Quantifies the contribution of tuple τ to query result
- All possible permutations of subinstances of D
- Average of differences between having τ or not
- Counterfactuals implicitly involved and aggregated
- We investigated algorithmic, complexity and approximation problems
- Extended to aggregate queries
- It has been applied to measure contribution of tuples to inconsistency of a database
- In these areas there are underlying and useful connections to database repairs w.r.t. integrity constraints

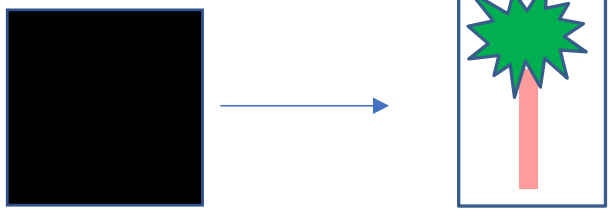
A Score-Based Approach: Shapley Values -3

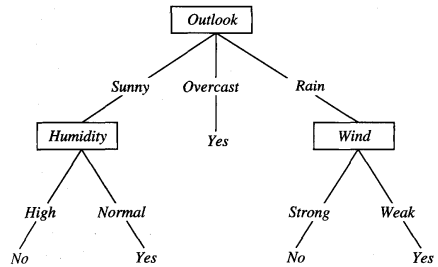
- Assume the classifier is binary, with labels 0 and 1
- Set of players $\mathcal{F}$ contain features All relative to $\mathbf{e}$
- Game function: $\mathcal{G}_{\mathbf{e}}(S) := \mathbb{E}(L(\mathbf{e}') \mid \mathbf{e}'_S = \mathbf{e}_S)$ ($\mathbf{e}_S$: projection on S)
- For a feature $F \in \mathcal{F}$, compute: $Shap(\mathcal{F}, \mathcal{G}_{\mathbf{e}}, F)$

  $\sum_{S \subseteq \mathcal{F} \setminus \{F\}} \frac{|S|!(|\mathcal{F}|-|S|-1)!}{|\mathcal{F}|!} [\mathbb{E}(L(\mathbf{e}')|\mathbf{e}'_{S\cup\{F\}} = \mathbf{e}_{S\cup\{F\}}) - \mathbb{E}(L(\mathbf{e}')|\mathbf{e}'_S = \mathbf{e}_S)]$

- *Shap* score has become popular (Lee & Lundberg; 2017)
- Assuming a probability distribution on entity population
- *x-Resp* can be generalized as expected value for multi-valued features
- Both *Resp* and *Shap* may end up considering exponentially many combinations
- We have experimentally compared *Resp* and *Shap*

A Score-Based Approach: Shapley Values -4





- Can we do better when we have the classification model?
- What if we have a decision tree, or a random forest, or a Boolean circuit?
- Can we compute *Shap* in polynomial time?
- We investigated this problem in detail in a AAAI'21 paper
- Tractable and intractable cases
- Provided algorithms for the former
- In particular, tractable for decision trees and random forests
- Investigated approximation algorithms

Score-Based Approaches: General Remarks

- There are many interesting open problems to investigate
- Addition of semantic and domain knowledge being an important one
- Reasoning about counterfactuals
- Connections to model-based diagnosis?
- Explanations are at the basis of fairness and bias analysis
- Identifying unexpected or undesirable high-score features becomes relevant
- Another promising problem: higher-order analytics on explanations
- What else can be learnt about the population or our classification mechanism?
- These extensions belong to our ongoing research

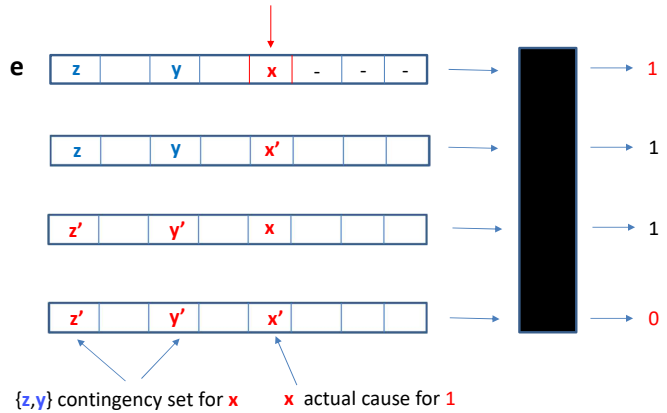
The *Resp* Score: Classification -1

- $\mathbf{e} = \langle \ldots, \mathbf{e}_F, \ldots \rangle, \quad F \in \mathcal{F}$ (B, Li, Schleich, Suciu, Vagena; DEEM@SIGMOD'20)
- $\mathit{Counter}(\mathbf{e}, F) := L(\mathbf{e}) - \mathbb{E}(L(\mathbf{e}') \mid \mathbf{e}'_{\mathcal{F} \setminus \{F\}} = \mathbf{e}_{\mathcal{F} \setminus \{F\}})$
- Easy to compute, and gives reasonable results
- So as with SHAP, requires underlying probability space
- No need to access the internals of the classification model
- Changing one value may not switch the label

  No explanations are obtained
- Extend it bringing in contingency sets of feature values!
- The *Resp*-score
- First a simplified version

The *Resp* Score: Classification -2

- Want explanation for label "1"
- Through changes of feature values, try to get "0"
- Fix a feature value $\mathbf{x} = \mathbf{e}_F$



- $\mathbf{x}$ counterfactual explanation for $L(\mathbf{e}) = 1$ if $L(\mathbf{e}\frac{\mathbf{x}}{\mathbf{x}'}) = 0$, for $\mathbf{x}' \in Dom(F)$
- $\mathbf{x}$ actual explanation for $L(\mathbf{e}) = 1$ if there are values $\mathbf{Y}$ in $\mathbf{e}$, $\mathbf{x} \notin \mathbf{Y}$, and new values $\mathbf{Y}' \cup \{\mathbf{x}'\}$:

  (a) $L(\mathbf{e}\frac{\mathbf{Y}}{\mathbf{Y}'}) = 1$ (b) $L(\mathbf{e}\frac{\mathbf{x}\mathbf{Y}}{\mathbf{x}'\mathbf{Y}'}) = 0$

- If $\mathbf{Y}$ is minimum in size: $Resp(\mathbf{x}) := \frac{1}{1+|\mathbf{Y}|}$

The *Resp* Score: Example

$\mathcal{C}$

entity (id)	F_1	F_2	F_3	L
$\mathbf{e}_1$	0	1	1	1
$\mathbf{e}_2$	1	1	1	1
$\mathbf{e}_3$	1	1	0	1
$\mathbf{e}_4$	**1**	**0**	1	0
$\mathbf{e}_5$	1	0	0	1
$\mathbf{e}_6$	0	1	0	1
$\mathbf{e}_7$	0	**0**	1	0
$\mathbf{e}_8$	0	0	0	0

- Due to $\mathbf{e}_7$, $F_2(\mathbf{e}_1)$ is counterfactual explanation, with $\mathit{Resp}(\mathbf{e}_1, F_2) = 1$

  Due to $\mathbf{e}_4$, $F_1(\mathbf{e}_1)$ is actual explanation; with $\{F_2(\mathbf{e}_1)\}$ as contingency set:

  $\mathit{Resp}(\mathbf{e}_1, F_1) = \frac{1}{2}$

- For non-binary features, *Resp* can be expressed as an expected value

The *Resp* Score: General Definition -1

- $\mathbf{e}$ entity under classification, with $L(\mathbf{e}) = 1$, and $F^\star \in \mathcal{F}$
  1. $\Gamma \subseteq \mathcal{F} \setminus \{F^\star\}$, set of features that may accompany $F^\star$
  2. $\bar{w} = (w_F)_{F \in \Gamma}$, $w_F \in dom(F)$, $w_F \neq \mathbf{e}_F$, new values for Γ
  3. $\mathbf{e}' := \mathbf{e}[\Gamma := \bar{w}]$, i.e. reset $\mathbf{e}$'s values for Γ as in $\bar{w}$
  4. $L(\mathbf{e}') = L(\mathbf{e}) = 1$, same label, but maybe extra change ...
  5. Pick $v \in dom(F^\star)$, $\mathbf{e}'' := \mathbf{e}[\Gamma := \bar{w}, F^\star := v]$
- When $F^\star(\mathbf{e}) \neq v$ and $L(\mathbf{e}) \neq L(\mathbf{e}'') = 0$, $F^\star(\mathbf{e})$ is an *actual causal explanation* for $L(\mathbf{e}) = 1$ with contingency $\langle \Gamma, \mathbf{e}_\Gamma \rangle$
- For "local" *Resp*-score make v vary randomly under 1.-5.

$$Resp(\mathbf{e}, F^\star, \mathcal{F}, \Gamma, \bar{w}) := \frac{L(\mathbf{e}') - \mathbb{E}[L(\mathbf{e}'') \mid \mathbf{e}''_{\mathcal{F} \setminus \{F^\star\}} = \mathbf{e}'_{\mathcal{F} \setminus \{F^\star\}}]}{1 + |\Gamma|} \quad (*)$$

- Globally: $Resp(\mathbf{e}, F^\star) := \max_{\substack{|\Gamma| \text{ min.}, (*) > 0 \\ \langle \Gamma, \bar{w} \rangle \models 1.-4.}} Resp(\mathbf{e}, F^\star, \mathcal{F}, \Gamma, \bar{w})$

Resp and *Shap* Scores: Experiments -1

- Several probability distributions can be used
- Among them, two coming from sample $T \subseteq \mathcal{E}$
- Empirical distribution: $P(\mathbf{e}) := \begin{cases} \frac{1}{|T|} & \text{if } \mathbf{e} \in T \\ 0 & \text{if } \omega \notin T \end{cases} \qquad \mathbf{e} \in \mathcal{E}$
- Product probability space over $\mathcal{E}$: (say, for binary features)

  $p_i = P(F_i = 1) \approx \frac{|\{\mathbf{e} \in T \mid \omega_i = 1\}|}{|T|} =: \hat{p}_i$ (empirical marginals)

  $$P(\mathbf{e}) := \Pi_{\mathbf{e}_i=1} \hat{p}_i \times \Pi_{\mathbf{e}_j=0} (1 - \hat{p}_j), \quad \text{for } \mathbf{e} \in \mathcal{E}$$

- *Resp* score computed on product space
- Not very good at capturing feature correlations
- Empirical distribution not suitable for *Resp*-score

Resp and *Shap* Scores: Experiments -2

- *Resp* score computation for $\mathbf{e} \in \mathcal{E}$:
  - Expectations relative to product probability space
  - Intervention values from feature domains determined by T
  - Call the classifier
  - Possibly restrict contingency sets to, say, two features
- *Shap* score applied with empirical distribution
- Already clear that computation on the product probability could be $\#P$-hard
- Use sample $T \subseteq \mathcal{E}$, test data

  Labels $L(\mathbf{e})$, $\mathbf{e} \in T$, computed with learned classifier
- *Shap* with expectations over this space, directly over data/labels in T

The *Shap* Score: Boolean-Circuit Classifiers[1] -1

- $Shap(\mathcal{F}, \mathcal{G}_{\mathbf{e}}, F) \;=$

  $\sum_{S \subseteq \mathcal{F} \setminus \{F\}} \frac{|S|!(|\mathcal{F}|-|S|-1)!}{|\mathcal{F}|!} [\mathbb{E}(L(\mathbf{e}')|\mathbf{e}'_{S \cup \{F\}} = \mathbf{e}_{S \cup \{F\}}) - \mathbb{E}(L(\mathbf{e}')|\mathbf{e}'_S = \mathbf{e}_S)]$

- Depends on $\mathbf{e}$ and (the classifier behind) L

- $Dom(F_i) = \{0,1\}, \quad F_i \in \mathcal{F},\ i = 1, \ldots, n, \quad \mathbf{e} \in \mathcal{E} := \{0,1\}^n$
  $L(\mathbf{e}) \in \{0,1\}$

- There is also a probability distribution $\mathcal{P}$ on $\mathcal{E}$

- We will identify the Boolean classifier with L

  $SAT(L) := \{\mathbf{e} \mid L(\mathbf{e}) = 1\} \qquad\qquad \#SAT(L) := |SAT(L)|$

  Counting the number of inputs that get label 1

- Proposition: For the uniform distribution $\mathcal{P}^u$, and $\mathbf{e} \in \mathcal{E}$

  $$\#SAT(L) \;=\; 2^{|\mathcal{F}|} \times (\; L(\mathbf{e}) - \textstyle\sum_{i=1}^{n} Shap(\mathcal{F}, G_{\mathbf{e}}, F_i) \;)$$

[1] Some slides borrowed from Pablo Barcelo

The *Shap* Score: Boolean-Circuit Classifiers -2

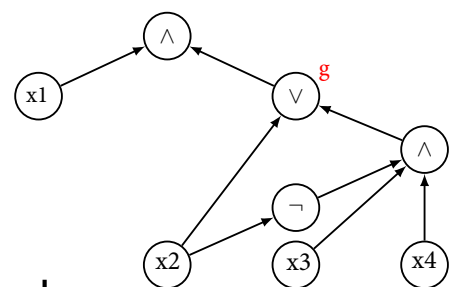
- $\#SAT \leq^{Turing}_{PTIME} Shap$
- When $\#SAT(L)$ is hard for a Boolean classifier L, computing Shap is also hard
- Negative Corollary: Computing *Shap* is $\#P$-hard for
  - Linear perceptron classifier
    By reduction from *#Knapsack* (with weights in binary)
  - Boolean classifiers defined by Monotone 2DNF or Monotone 2CNF [Provan & Ball, 1983]
- Can we do better for other classes of binary classifiers?

  Other classes of Boolean-circuit classifiers?

d-D Boolean-Circuits -1

- A Boolean circuit over set of variables X is a DAG $\mathcal{C}$ with:
  - Each node without incoming edges (input) is labeled with either a variable $x \in X$ or a constant in $\{0, 1\}$
  - Each other node is labeled with a gate in $\{\neg, \wedge, \vee\}$
  - There is a single sink node, O, called the output
- $\mathbf{e}: X \to \{0, 1\}$ (equivalently $\mathbf{e} \in \{0, 1\}^{|X|}$) is accepted by $\mathcal{C}$, written $\mathcal{C}(\mathbf{e}) = 1$, iff O takes value 1
- For a gate g of $\mathcal{C}$, $\mathcal{C}(g)$ is the induced subgraph containing gates on a path in $\mathcal{C}$ to g

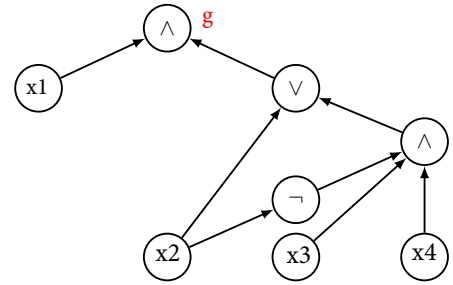
  $Var(g)$ is the set of variables of $\mathcal{C}(g)$
  $Var(g) = \{x2, x3, x4\}$



- $\mathcal{C}$ is deterministic if every $\vee$-gate g with input gates g_1, g_2: $\mathcal{C}(g_1)(\mathbf{e}) \neq \mathcal{C}(g_2)(\mathbf{e})$, for every $\mathbf{e}$

d-D Boolean-Circuits -2

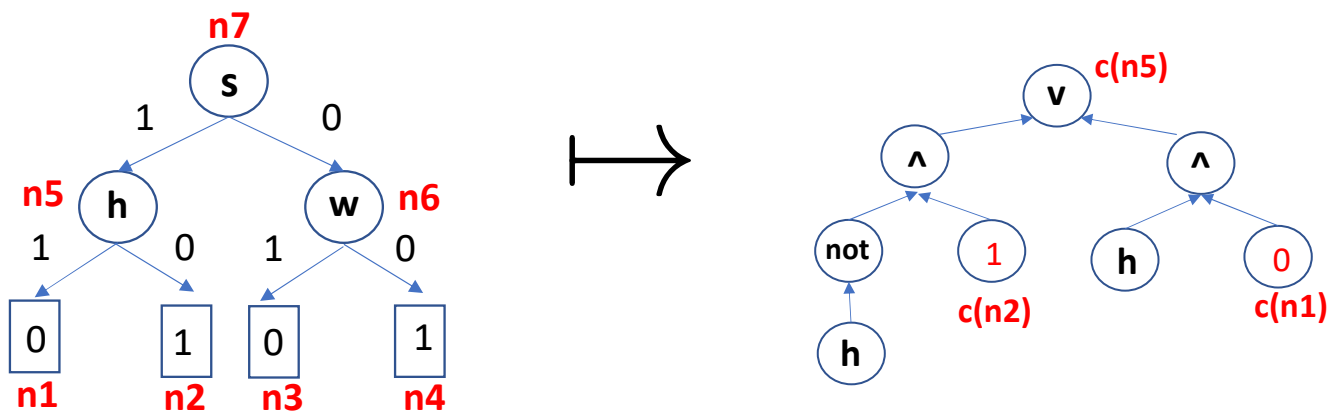
- $\mathcal{C}$ is decomposable if every $\wedge$-gate g with input gates g_1, g_2: $Var(g_1) \cap Var(g_2) = \emptyset$



- We will consider $\mathcal{C}$ to be deterministic and decomposable circuit (d-D circuit)
- Several classes of Boolean models can be translated in polynomial time into d-D Boolean circuits:
  - Decision trees
  - Ordered binary decision diagrams (OBDDs)
  - Free binary decision diagrams (FBDDs)
  - Deterministic-decomposable negation normal-form (d-D NNFs)

d-D Boolean-Circuits -3

- Compiling binary decision trees into d-D Boolean Circuits
- An inductive construction starting from the bottom of the DT
- Leaves of DT become constant binary gates in d-DC
- By induction one can prove the resulting circuit is d-D
- Final d-DC is the compilation $c(r)$ of root node r of DT



- Final equivalent d-DC: $c(n7)$
- Computable in linear time

The SHAP Score: d-D Boolean-Circuits -1

- *Shap* computation in polynomial time not precluded
- Proposition: For d-D circuits $\mathcal{C}$, $\#SAT(\mathcal{C})$ can be computed in polynomial time

  Idea: Bottom-up procedure that inductively computes $\#SAT(\mathcal{C}(g))$, for each gate g of $\mathcal{C}$
- To show that *Shap* can be computed efficiently for d-D circuits, we need a detailed analysis
- We assume the uniform distribution for the moment
- A related problem: "satisfiable circle of an entity"

$$SAT(\mathcal{C}, \mathbf{e}, \ell) := SAT(\mathcal{C}) \cap \{ \; e' \mid \underbrace{\|\mathbf{e} - \mathbf{e}'\|_1 = \ell}_{\ell \text{ value discrepancies}} \; \}$$

$$\#SAT(\mathcal{C}, \mathbf{e}, \ell) := |SAT(\mathcal{C}, \mathbf{e}, \ell)|$$

- Proposition: If computing $\#SAT(\mathcal{C}, \mathbf{e}, \ell)$ is tractable, so is $Shap(X, \mathcal{G}_{\mathbf{e}}, x)$

The SHAP Score: d-D Boolean-Circuits -2

- Main Result: $\#SAT(\mathcal{C}, \mathbf{e}, \ell)$ can be solved in polynomial time for d-D circuits $\mathcal{C}$, entities $\mathbf{e}$, and $1 \leq \ell \leq |X|$

  Idea: Inductively compute $\#SAT(\mathcal{C}(g), \mathbf{e}_{Var(g)}, \ell)$ for each gate $g \in \mathcal{C}$ and integer $\ell \leq |Var(g)|$

  - Input gate: immediate
  - $\neg$-gate:

    $\#SAT(\mathcal{C}(\neg g), \mathbf{e}_{Var(g)}, \ell) = \binom{Var(g)}{\ell} - \#SAT(\mathcal{C}(g), \mathbf{e}_{Var(g)}, \ell)$
  - $\vee$-gate: (uses determinism)

    $\#SAT(\mathcal{C}(g_1 \vee g_2), \mathbf{e}_{Var(g_1) \cup Var(g_2)}, \ell) =$

    $\#SAT(\mathcal{C}(g_1), \mathbf{e}_{Var(g_1)}, \ell) + \#SAT(\mathcal{C}(g_2), \mathbf{e}_{Var(g_2)}, \ell)$
  - $\wedge$-gate: (uses decomposition)

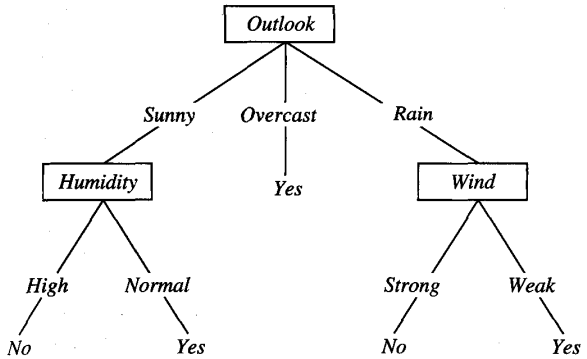
    $\#SAT(\mathcal{C}(g_1 \wedge g_2), \mathbf{e}_{Var(g_1) \cup Var(g_2)}, \ell) =$

    $\sum_{j+k=\ell} \#SAT(\mathcal{C}(g_1), \mathbf{e}_{Var(g_1)}, j) \times \#SAT(\mathcal{C}(g_2), \mathbf{e}_{Var(g_2)}, k)$

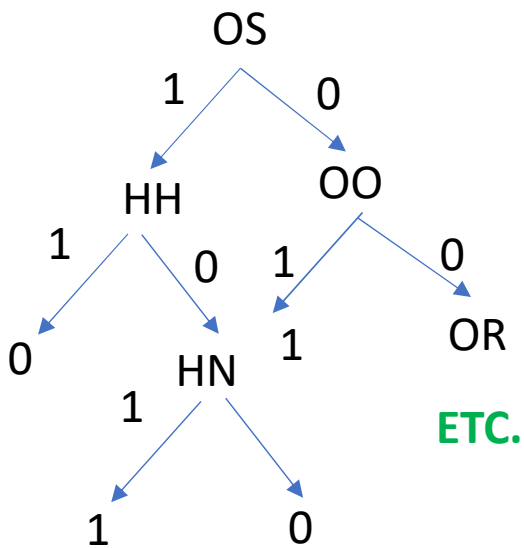
The SHAP Score: d-D Boolean-Circuits -3

- Theorem: *Shap* can be computed in polynomial time for d-D circuits under the uniform distribution
- Corollary: *Shap* can be computed in polynomial time for decision trees and random forests, OBDDs, etc., under the uniform distribution
- It can be extended to any product distribution on $\{0,1\}^{|X|}$ (uniform is a particular case)

The SHAP Score: Beyond Binary Features



- "Binarize" features
- *OutlookSunny* (OS)
  OutlookOvercast, *OutlookRain*, etc.
  become propositional features



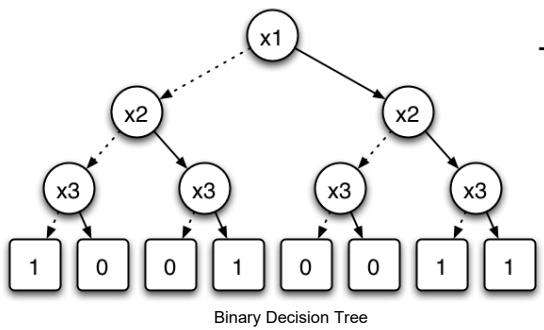
Certain entities become impossible (probability 0)

$\mathbf{e} = \langle \underbrace{0, 1, 1}_{\text{for OS, OO, OR}} , \ldots \rangle$ ×

Ordered Binary Decision Diagrams

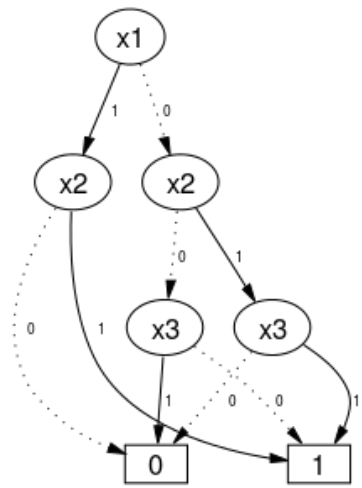
- Our polynomial time algorithm for *Shap* can be applied to *Ordered Binary Decision Diagrams* (OBDDs)
- They are relevant for several reasons in *Knowledge Compilation*
- In particular, to represent "opaque" classifiers as OBDDs, e.g. binary neural networks [Shi, Shih, Darwiche, Choi; KR20]
- Opening the ground for efficiently applying *Shap* to them

$f(x_1, x_2, x_3) = (\neg x_1 \wedge \neg x_2 \wedge \neg x_3) \vee (x_1, \wedge x_2) \vee (x_2 \wedge x_3)$



Binary Decision Tree

x1	x2	x3	f
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



OBDD

Same variable order along full paths

References (self-references for this presentation)

- Bertossi, L. and Salimi, B. "From Causes for Database Queries to Repairs and Model-Based Diagnosis and Back". Theory of Computing Systems, 2017, 61(1):191-232.

- Bertossi, L. and Salimi, B. "Causes for Query Answers from Databases: Datalog Abduction, View-Updates, and Integrity Constraints". International Journal of Approximate Reasoning, 2017, 90:226-252.

- E. Livshits, L. Bertossi, B. Kimelfeld and M. Sebag. "The Shapley Value of Tuples in Query Answering". In Proc. ICDT 2020.

- L. Bertossi, J. Li, M. Schleich, D. Suciu and Z. Vagena. "Causality-based Explanation of Classification Outcomes". Proc. 4th International Workshop on "Data Management for End-to-End Machine Learning" (DEEM) at ACM SIGMOD/PODS, 2020, pp. 6.1-6.10.

- L. Bertossi. "Score-Based Explanations in Data Management and Machine Learning". Proc. Int. Conf. Scalable Uncertainty Management (SUM 20), Springer LNCS 2322, pp. 17-31.

- Marcelo Arenas, Pablo Barcelo, Leopoldo Bertossi, Mikael Monet. "The Tractability of SHAP-scores over Deterministic and Decomposable Boolean Circuits". Proc. AAAI 2021.