Causality and Explanations in Data Management and Machine Learning

Leopoldo Bertossi

bertossi@scs.carleton.ca

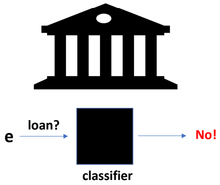
Mindbridge, Ottawa, March 2025

Explanations in Machine Learning

- Bank client $\mathbf{e} = \langle \text{john}, 18, \text{plumber}, 70\text{K}, \text{harlem}, \ldots \rangle$

  As an entity represented as a record of values for features Name, Age, Activity, Income, ...

- $\mathbf{e}$ requests a loan from a bank, which uses a classifier



- The client asks *Why?*
- What kind of *explanation?*
  How?
  From what?

Explanations in AI

- A problem that is common in applications of AI systems
- Users and stakeholders affected by their results need explanations
- Whole new area of AI: *Explainable AI* (XAI)
- Part of AI:
  - AI systems should be extended with explanation capabilities
  - AI researchers and professionals understand those systems
  - Humans explanations are part of intelligent behaviour

    Hence, explanation building should be a capability of AI agents
  - Explanations have to be understood, modeled, implemented, ... as part of AI

- XAI is of interest to many other people
- Stakeholders are being affected by *outcomes from AI systems*
  Assessments (e.g. a credit score), classifications (good/bad client), decisions (approve/reject loan), etc.
- There is a need for more *transparent, trustable, fair, unbiased, responsible* AI systems
- A whole discipline has emerged: *Ethical AI*
- It touches many others, including AI itself, but beyond: Law, Sociology, Philosophy, ..., Business, ...
- Also, *interpretable* AI systems
- New legislation forces (owners of) AI systems affecting users to provide explanations and guarantee all of the above

It may really be a "black box"!

Explanations (in AI)

- Search for explanations belongs to the nature of human beings
  The quest has been around since the inception of humans
- Ancient Greeks already concerned with *causes* (and effects)
- Are explanations a new subject in AI?
- Yes and No
- Explanations have been studied in AI for some decades by now
  And in related disciplines: Logic, Statistics, Philosophy, Physics, ...
- Some forms of explanations are new in AI
  Others have roots in already existing ones

Explanations in Databases

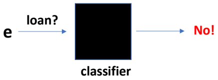
Receives	*R*.1	*R*.2
	s_2	s_1
	s_3	s_3
	s_4	s_3

Store	*S*.1
	s_2
	s_3
	s_4

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

  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 query answering (QA) (could be violation of ICs, etc.)

Explanations in Machine Learning (back)



- Client requesting a loan from a bank using a black-box classifier
- It may have been learned from data, and became a very complicated model (and implementation)
- $$\mathbf{e} = \langle \text{john}, 18, \text{plumber}, 70\text{K}, \text{harlem}, \ldots \rangle$$

  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

Causality and Responsibility

- Causality has been developed in AI for 3 decades or so
  In particular, Actual Causality (Halpern & Pearl, 2001)
- Also the quantitative notion of Responsibility: A measure of causal contribution (Chockler & Halpern, 2004)
- Both based on Counterfactual Interventions
  Hypothetical changes of values in a (causal) model to detect other changes To identify actual causes
- Do deletions of certain database tuples make the query false?
  Do changes of feature values change the label to "Yes"?
- We have investigated causality, counterfactual explanations, and responsibility in data management and classification
  Semantics, computational mechanisms, intrinsic complexity, logic-based specifications, reasoning, etc.

$\mathcal{Q}:\ \exists x \exists y (Store(x) \wedge Receives(x,y) \wedge Store(y))$

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~~

$D' \not\models \mathcal{Q}$

- $Receives(s_3, s_3)$ is actual cause

  With $\{Store(s_4)\}$ as minimum-size contingency set

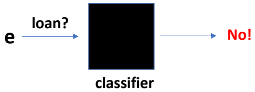
  It needs company to invalidate the query, extra deletions

- $Resp(Receives(s_3, s_3)) := \frac{1}{1 + |\{Store(s_4)\}|} = \frac{1}{2}$

- $Resp(Store(s_3)) := \frac{1}{1+0} = 1$ a counterfactual cause

  It has the highest possible responsibility (Meliou et al., 2010; B. & Salimi, TOCS 2017)

- Also explored in QA the causal-effect (score) of causality in observational studies



$$\mathbf{e} \;=\; \langle \text{john}, 18, \text{plumber}, 70\text{K}, \text{harlem}, \ldots \rangle \qquad \text{No}$$

- Counterfactual versions:

$$\mathbf{e}' \;=\; \langle \text{john}, 25, \text{plumber}, 70\text{K}, \text{harlem}, \ldots \rangle \qquad \text{Yes}$$

$$\mathbf{e}'' = \langle \text{john}, 18, \text{plumber}, 80\text{K}, \text{brooklyn}, \ldots \rangle \qquad \text{Yes}$$

- For the gist:

  1. Value for feature *Age* is counterfactual cause with explanatory responsibility $\mathit{Resp}(\mathbf{e}, \mathit{Age}) = 1$
  2. Value for *Income* is actual cause with $\mathit{Resp}(\mathbf{e}, \mathit{Income}) = \frac{1}{2}$
     This one needs additional (contingent) changes ...

- For binary features this form of responsibility works fine
  So as that for DBs

- For a multi-valued feature, possibly many new values for it do not change the label, and few of them do
- Then, the original value is not great explanation
- Responsibility score has to be generalized (B. et al., Deem@SIGMOD20)
- Better consider contingent features and values for them, and average labels!
- We are considering binary classifiers, with labels 1 or 0

  Assume label 1 is the one we want to explain
- *Resp* is a "local" explanation score: for a feature value in a particular entity
- It belongs to a family of Local and Model-Agnostic Attribution Scores

Generalized Responsibility

- $\mathbf{e}$ classified entity, $L(\mathbf{e}) = 1$, $F^\star \in \mathcal{F}$ (set of features)
- "Local" *Resp*-score: for fixed contingent assignment $\Gamma := \bar{w}$
  $\Gamma \subseteq \mathcal{F} \smallsetminus \{F^\star\}$ (potential contingent set of features)
- $\mathbf{e}' := \mathbf{e}[\Gamma := \bar{w}]$ (potential contingent values), with $L(\mathbf{e}') = L(\mathbf{e})$

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

  - $\mathbf{e}'' := \mathbf{e}[\Gamma := \bar{w}, F^\star := v]$, with $v \in dom(F^\star)$
  - $\mathbf{e}_S$ is projection of $\mathbf{e}$ on $S \subseteq \mathcal{F}$
  - When $(*) > 0$, $F^\star(\mathbf{e})$ is *actual causal explanation* for $L(\mathbf{e}) = 1$ with contingency $\langle \Gamma, \mathbf{e}_\Gamma \rangle$
- Global score: $Resp(\mathbf{e}, F^\star) := \max_{\langle \Gamma, \bar{w} \rangle,\, |\Gamma|\ min.,\ (*) > 0} Resp(\mathbf{e}, F^\star, \Gamma, \bar{w})$

- $(*)$ requires multiple "passes" through the classifier ...
- *Resp* requires (assumes) a probability distribution on the entity population $\mathcal{E}$

  Several probability distributions can be used (B. et al., Deem@SIGMOD20)

  Among them, two coming from sample $T \subseteq \mathcal{E}$

  - Empirical distribution: $P(\mathbf{e}) := \begin{cases} \frac{1}{|T|} & \text{if } \mathbf{e} \in T \\ 0 & \text{if } \omega \notin T \end{cases} \quad \mathbf{e} \in \mathcal{E}$
  - Product probability space over $\mathcal{E}$: (say, for binary features)

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

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

- In our experiments, *Resp* score computed on product space

  Not very good at capturing feature correlations

  Empirical distribution not suitable for *Resp* score

Shapley Values: *Shap*

- Based on the general Shapley value of coalition game theory
- For each application of Shapley one needs an appropriate game function that maps (sub)sets of players to real numbers
- Our case: Set of players $\mathcal{F}$ contain features, but relative to $\mathbf{e}$
- Game function: For $S \subseteq \mathcal{F}$, and $\mathbf{e}_S$ the projection of $\mathbf{e}$ on S

$$\mathcal{G}_{\mathbf{e}}(S) := \mathbb{E}(L(\mathbf{e}') \mid \mathbf{e}' \in \mathcal{E} \ \& \ \mathbf{e}'_S = \mathbf{e}_S)$$

- For a feature $F^\star \in \mathcal{F}$, compute: $\mathit{Shap}(\mathcal{F}, \mathcal{G}_{\mathbf{e}}, F^\star)$

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

- *Shap* score has become popular (Lee & Lundberg, 2017)
- Assumes a probability distribution on entity population

Experimenting with Scores

- In general, *Resp* and *Shap* consider exponentially many value combinations

  Still, *Resp* is in general simpler to compute

- We experimented with *Resp* and *Shap* (B. et al., Deem@SIGMOD20)

- 13 features of the Kaggle dataset for fraudulent card transactions

  0. credit.policy
  1. purpose
  2. int.rate
  3. installment
  4. log.annual.inc
  5. dti
  6. fico
  7. days.with.cr.line
  8. revol.bal
  9. revol.util
  10. inq.last.6mths
  11. delinq.2yrs
  12. pub.rec

  Classification about "fraudulent" (1) or not (0)

- XGBoost classifier using Python library (rather opaque model, basically black-box)

- Also experimented with FICO dataset for loan assignment
  ("Fair, Isaac and Company", https://www.fico.com)

  Computed *Resp*, *Shap*, *Banzhaf*, and FICO-Rudin scores

- C. Rudin uses internals of open-box model

  Coefficients of two coupled logistic regressions

- 23 features plus bucketization

  Requires approximate and optimized computations of black-box score computation

- *Resp* gave quite reasonable results

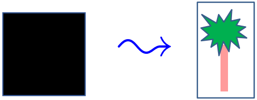
Shap: **Tractability**

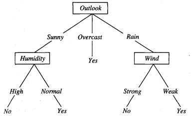
- Both *Resp* and *Shap* may end up considering exponentially many combinations

  And multiple passes through the black-box classifier

- Both provably intractable in the general case

- Can we do better with an open-box classifier?





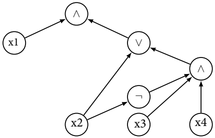
  Exploiting its elements and internal structure?

- 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 (Arenas, Barcelo, B., Monet; AAAI21)
- Tractable and intractable cases, with algorithms for the former

  Investigated existence (or not) of good approximation algorithms
- Choosing the right abstraction (model) is crucial
- We used Boolean classifiers (BCs), i.e. propositional formulas with (binary) output gate
- We established early on that computing *Shap* is at least as hard as counting the satisfying truth assignments of the BC (intractable in general)

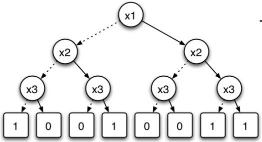


- So, it has to be a broad and interesting class of BCs for which the latter problem is not intractable

- We concentrated on the class of deterministic and decomposable Boolean circuits (dDBCs) (example above)
  - Input gates are variables (features) or constants
  - An $\vee$-gate never has both inputs true (determinism)
  - An $\wedge$-gate do not has inputs sharing variables (decomposability)
- A class of BCs that includes -possibly via efficient compilation- many interesting ones, syntactic and not ...
  - Decision trees (and random forests)
  - Ordered binary decision diagrams (OBDDs)
  - Sentential decision diagrams (SDDs)
  - Deterministic-decomposable negation normal-form (dDNNFs)
- Theorem: For dDBCs, under the uniform or product distribution, *Shap* can be computed in polynomial time

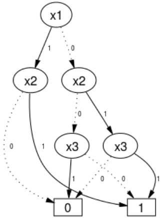
- Binary decision trees can be inductively compiled into dDBCs
- Non-binary ones can be binarized first
- OBDDs can also be compiled into dDBCs

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



Binary Decision Tree

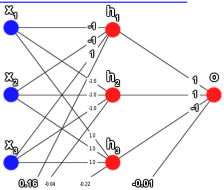
x1	x2	x3	*f*
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



OBDD (same variable order along paths)

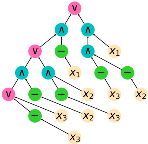
- Etc.
- We obtain tractability of *Shap* for all these classes of classifiers

SHAP on Binary Neural Networks



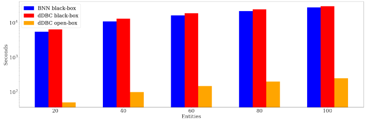
- NNs considered as black-boxes
- We experimented with SHAP computation on a BNN via compilation into a dDBC
  1. BNN $\mapsto$ CNF (parsimonious and optimized)
  2. CNF $\mapsto$ SDD (non-polynomial, but FPT)
  3. SDD $\mapsto$ dDBC (straightforward)

  Still worth this one-time computation
  (target dDBC may be used multiple times)



- Experiments: BNN with 14 gates, dDBC with 18,670 nodes

  Compared SHAP computation for:
  black-box BNN, open-box dDBC,
  and black-box dDBC



(logarithmic scale)

- All SHAP scores for all entities,
  with increasing numbers of them (JELIA'23)