Expected Shapley-Like Scores of Boolean Functions: Complexity and Applications to Probabilistic Databases

Pratik Karmakar Mikaël Monet
Pierre Senellart Stéphane Bressan

ACM PODS, June 2024

Motivation

Question

How to assess the responsibility of data items when they are both uncertain and involved in a complex task?

Motivation

Question

How to assess the responsibility of data items when they are both uncertain and involved in a complex task?

Practical motivation

- Shapley-Like scores (Shapley, Banzhaf, etc.) [Laruelle, 1999]: reasonable ways to quantify the responsibility of a data item for a complex task such as query evaluation

Motivation

Question

How to assess the responsibility of data items when they are both uncertain and involved in a complex task?

Practical motivation

- Shapley-Like scores (Shapley, Banzhaf, etc.) [Laruelle, 1999]: reasonable ways to quantify the responsibility of a data item for a complex task such as query evaluation
- Real data: marred with uncertainty, which may be represented by probability distributions

Motivation

Question

How to assess the responsibility of data items when they are both uncertain and involved in a complex task?

Practical motivation

- Shapley-Like scores (Shapley, Banzhaf, etc.) [Laruelle, 1999]: reasonable ways to quantify the responsibility of a data item for a complex task such as query evaluation
- Real data: marred with uncertainty, which may be represented by probability distributions

Theoretical motivation

The tractability landscapes of Shapley value computation and probabilistic query evaluation are similar

Motivation

Question

How to assess the responsibility of data items when they are both uncertain and involved in a complex task?

Practical motivation

- Shapley-Like scores (Shapley, Banzhaf, etc.) [Laruelle, 1999]: reasonable ways to quantify the responsibility of a data item for a complex task such as query evaluation
- Real data: marred with uncertainty, which may be represented by probability distributions

Theoretical motivation

The tractability landscapes of Shapley value computation and probabilistic query evaluation are similar – How does the Shapley value computation landscape change when the database is probabilistic?

Shapley-like scores

- V: finite set of Boolean variables
- $\varphi : 2^V \to \{0, 1\}$ Boolean function over V
- $c : \mathbb{N} \times \mathbb{N} \to \mathbb{Q}$: coefficient function (assumed to have PTIME evaluation when input in unary)

Shapley-like scores

- V: finite set of Boolean variables
- $\varphi : 2^V \to \{0,1\}$ Boolean function over V
- $c : \mathbb{N} \times \mathbb{N} \to \mathbb{Q}$: coefficient function (assumed to have PTIME evaluation when input in unary)

$$\mathrm{Score}_c(\varphi, V, x) \stackrel{\text{def}}{=} \sum_{E \subseteq V \setminus \{x\}} c(|V|, |E|) \times \left[\varphi(E \cup \{x\}) - \varphi(E)\right].$$

Shapley-like scores

- V: finite set of Boolean variables
- $\varphi : 2^V \to \{0,1\}$ Boolean function over V
- $c : \mathbb{N} \times \mathbb{N} \to \mathbb{Q}$: coefficient function (assumed to have PTIME evaluation when input in unary)

$$\mathrm{Score}_c(\varphi, V, x) \stackrel{\text{def}}{=} \sum_{E \subseteq V \setminus \{x\}} c(|V|, |E|) \times \left[\varphi(E \cup \{x\}) - \varphi(E)\right].$$

Example

- $c_{\text{Shapley}}(k, \ell) \stackrel{\text{def}}{=} \frac{\ell!(k-l-1)!}{k!} = \binom{k-1}{l}^{-1} k^{-1}$: Shapley value [Shapley et al., 1953]
- $c_{\text{Banzhaf}}(k, \ell) \stackrel{\text{def}}{=} 1$: Banzhaf value [Banzhaf III, 1964]
- $c_{\text{PB}}(k, \ell) \stackrel{\text{def}}{=} 2^{-k+1}$: Penrose–Banzhaf power [Kirsch and Langner, 2010]

Boolean functions with uncertain variables

- Product distribution on Boolean variables, $\Pr(x) \in [0, 1]$ for $x \in V$ (i.e., every Boolean variable is assumed to be independent)

Boolean functions with uncertain variables

- Product distribution on Boolean variables, $\Pr(x) \in [0, 1]$ for $x \in V$ (i.e., every Boolean variable is assumed to be independent)
- For $Z \subseteq V$,
  $\Pr(Z) \stackrel{\text{def}}{=} \left(\prod_{x \in Z} \Pr(x) \right) \times \left(\prod_{x \in V \setminus Z} (1 - \Pr(x)) \right)$

Boolean functions with uncertain variables

- Product distribution on Boolean variables, $\Pr(x) \in [0, 1]$ for $x \in V$ (i.e., every Boolean variable is assumed to be independent)
- For $Z \subseteq V$, $\Pr(Z) \stackrel{\text{def}}{=} \left(\prod_{x \in Z} \Pr(x) \right) \times \left(\prod_{x \in V \setminus Z} (1 - \Pr(x)) \right)$
- $\Pr(\varphi) \stackrel{\text{def}}{=} \sum_{Z \subseteq V} \Pr(Z) \varphi(Z)$: the probability of the Boolean function φ to be true, aka, the expected value of the Boolean function

Boolean functions with uncertain variables

- Product distribution on Boolean variables, $\Pr(x) \in [0,1]$ for $x \in V$ (i.e., every Boolean variable is assumed to be independent)
- For $Z \subseteq V$,
  $\Pr(Z) \stackrel{\text{def}}{=} \left(\prod_{x \in Z} \Pr(x) \right) \times \left(\prod_{x \in V \setminus Z} (1 - \Pr(x)) \right)$
- $\Pr(\varphi) \stackrel{\text{def}}{=} \sum_{Z \subseteq V} \Pr(Z) \varphi(Z)$: the probability of the Boolean function φ to be true, aka, the expected value of the Boolean function
- $\text{EScore}_c(\varphi, x) \stackrel{\text{def}}{=} \sum_{\substack{Z \subseteq V \\ x \in Z}} (\Pr(Z) \times \text{Score}_c(\varphi, Z, x))$ the expected score of x for φ

Problems studied

We consider classes of representations of Boolean functions, e.g., Boolean circuits, d-D circuits. We assume $\varphi(\emptyset)$ to be computable in PTIME.

- $\mathsf{EV}(\mathcal{F}) : \varphi \in \mathcal{F} \mapsto \Pr(\varphi)$
- $\mathsf{Score}_c(\mathcal{F}) : (\varphi \in \mathcal{F}, x \in V) \mapsto \mathsf{Score}_c(\varphi, V, x)$ for some coefficient function c
- $\mathsf{EScore}_c(\mathcal{F}) : (\varphi \in \mathcal{F}, x \in V) \mapsto \mathsf{EScore}_c(\varphi, x)$

We look for the complexity of these problems and for (Turing) polynomial-time reductions between problems, denoted $A \leqslant_{\mathrm{P}} B$, for class of Boolean functions (and $A \equiv_{\mathrm{P}} B$ for two-way reductions).

d-D circuits

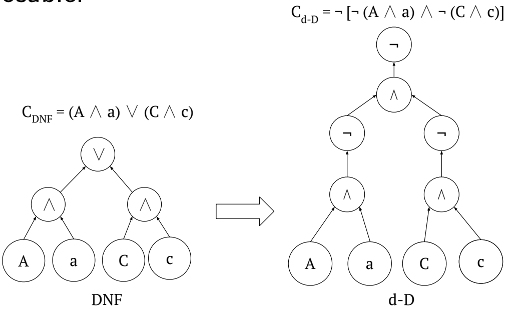
- Determinism: An $\vee$-gate g is deterministic if the Boolean functions captured by each pair of distinct input gates of g have pairwise disjoint models. A Boolean circuit C is deterministic if all the $\vee$-gates in C are deterministic.

d-D circuits

- Determinism: An $\vee$-gate g is deterministic if the Boolean functions captured by each pair of distinct input gates of g have pairwise disjoint models. A Boolean circuit C is deterministic if all the $\vee$-gates in C are deterministic.
- Decomposability: An $\wedge$-gate g is decomposable if for pair of input gates g_1 and g_2, we have $\mathrm{Vars}(g_1) \cap \mathrm{Vars}(g_2) = \emptyset$. A Boolean circuit C is decomposable if all the $\wedge$-gates in C are decomposable.

d-D circuits

- Determinism: An $\vee$-gate g is deterministic if the Boolean functions captured by each pair of distinct input gates of g have pairwise disjoint models. A Boolean circuit C is deterministic if all the $\vee$-gates in C are deterministic.
- Decomposability: An $\wedge$-gate g is decomposable if for pair of input gates g_1 and g_2, we have $\mathrm{Vars}(g_1) \cap \mathrm{Vars}(g_2) = \emptyset$. A Boolean circuit C is decomposable if all the $\wedge$-gates in C are decomposable.



Expected Shapley-like scores

Area

ID	Region	Area	Prob.	Prov.
01	Valparaiso	16,000	0.4	*A*
02	Atacama	75,000	0.3	*B*
03	Metropolitan	15,000	0.6	*C*
04	Maule	30,000	0.8	*D*

Density

ID	Pop_den	Prob.	Prov.
01	110	0.5	*a*
02	4	0.2	*b*
03	461	0.8	*c*
04	34	0.9	*d*

Expected Shapley-like scores

Area

ID	Region	Area	Prob.	Prov.
01	Valparaiso	16,000	0.4	*A*
02	Atacama	75,000	0.3	*B*
03	Metropolitan	15,000	0.6	*C*
04	Maule	30,000	0.8	*D*

Density

ID	Pop_den	Prob.	Prov.
01	110	0.5	*a*
02	4	0.2	*b*
03	461	0.8	*c*
04	34	0.9	*d*

```
SELECT DISTINCT 1 FROM Area a JOIN Density d ON a.ID = d.ID
WHERE Area < 20000 AND Pop_den >= 100
```

Expected Shapley-like scores

Area

ID	Region	Area	Prob.	Prov.
01	Valparaiso	16,000	0.4	A
02	Atacama	75,000	0.3	B
03	Metropolitan	15,000	0.6	C
04	Maule	30,000	0.8	D

Density

ID	Pop_den	Prob.	Prov.
01	110	0.5	a
02	4	0.2	b
03	461	0.8	c
04	34	0.9	d

```
SELECT DISTINCT 1 FROM Area a JOIN Density d ON a.ID = d.ID
WHERE Area < 20000 AND Pop_den >= 100
```

Provenance: $\varphi_{\text{ex}} = (A \wedge a) \vee (C \wedge c)$

Expected Shapley-like scores

Area

ID	Region	Area	Prob.	Prov.
01	Valparaiso	16,000	0.4	A
02	Atacama	75,000	0.3	B
03	Metropolitan	15,000	0.6	C
04	Maule	30,000	0.8	D

Density

ID	Pop_den	Prob.	Prov.
01	110	0.5	a
02	4	0.2	b
03	461	0.8	c
04	34	0.9	d

```
SELECT DISTINCT 1 FROM Area a JOIN Density d ON a.ID = d.ID
WHERE Area < 20000 AND Pop_den >= 100
```

Provenance: $\varphi_{\text{ex}} = (A \wedge a) \vee (C \wedge c)$

$$\Pr(\varphi_{\text{ex}}) = 1-(1-p_A\times p_a)\times(1-p_C\times p_c) = 1-(1-0.4\times0.5)\times(1-0.6\times0.8) = 0.584$$

Expected Shapley-like scores

Area

ID	Region	Area	Prob.	Prov.
01	Valparaiso	16,000	0.4	A
02	Atacama	75,000	0.3	B
03	Metropolitan	15,000	0.6	C
04	Maule	30,000	0.8	D

Density

ID	Pop_den	Prob.	Prov.
01	110	0.5	a
02	4	0.2	b
03	461	0.8	c
04	34	0.9	d

```sql
SELECT DISTINCT 1 FROM Area a JOIN Density d ON a.ID = d.ID
WHERE Area < 20000 AND Pop_den >= 100
```

Provenance: $\varphi_{\text{ex}} = (A \wedge a) \vee (C \wedge c)$

$$\Pr(\varphi_{\text{ex}}) = 1-(1-p_A \times p_a)\times(1-p_C \times p_c) = 1-(1-0.4\times0.5)\times(1-0.6\times0.8) = 0.584$$

$x \in V$	p_x	$\text{Score}_{c_{\text{Shapley}}}(\varphi_{\text{ex}}, V, x)$	$\text{EScore}_{c_{\text{Shapley}}}(\varphi_{\text{ex}}, x)$
A	0.4	0.25	0.076
a	0.5	0.25	0.076
C	0.6	0.25	0.216
c	0.8	0.25	0.216
		1.0	0.584

What is known?

- $\text{Score}_{c_{\text{Shapley}}}(\text{d-D})$ is PTIME [Deutch et al., 2022]

What is known?

- $\text{Score}_{c_{\text{Shapley}}}(\text{d-D})$ is PTIME [Deutch et al., 2022]
- $\text{Score}_{c_{\text{Banzhaf}}}(\text{d-D})$ is PTIME [Abramovich et al., 2023]

What is known?

- $\text{Score}_{c_{\text{Shapley}}}(\text{d-D})$ is PTIME [Deutch et al., 2022]
- $\text{Score}_{c_{\text{Banzhaf}}}(\text{d-D})$ is PTIME [Abramovich et al., 2023]
- $\text{Score}_c(\mathcal{F}) \leqslant_{\text{P}} \text{EScore}_c(\mathcal{F})$ for any $\mathcal{F}$, c: just compute EScore_c with all probabilities set to 1

What is known?

- $\mathrm{Score}_{c_{\mathrm{Shapley}}}(\text{d-D})$ is PTIME [Deutch et al., 2022]
- $\mathrm{Score}_{c_{\mathrm{Banzhaf}}}(\text{d-D})$ is PTIME [Abramovich et al., 2023]
- $\mathrm{Score}_c(\mathcal{F}) \leqslant_{\mathrm{P}} \mathrm{EScore}_c(\mathcal{F})$ for any $\mathcal{F}$, c: just compute EScore_c with all probabilities set to 1
- $\mathrm{Score}_{c_{\mathrm{Shapley}}}(\mathcal{F}) \equiv_{\mathrm{P}} \mathrm{EV}(\mathcal{F})$ for any class $\mathcal{F}$ closed under $\vee$-substitutions [Kara et al., 2024] and when probabilities are uniform (unweighted model counting)

What have we shown?

Theorem

- $\mathsf{EScore}_c(\mathcal{F}) \leqslant_{\mathsf{P}} \mathsf{EV}(\mathcal{F})$ *for any* $\mathcal{F}$, c

What have we shown?

Theorem

- $\mathsf{EScore}_c(\mathcal{F}) \leqslant_{\mathsf{P}} \mathsf{EV}(\mathcal{F})$ *for any* $\mathcal{F}$, c
- $\mathsf{EScore}_{c_{\mathsf{Shapley}}}(\mathcal{F}) \equiv_{\mathsf{P}} \mathsf{EV}(\mathcal{F})$ *for any* $\mathcal{F}$

What have we shown?

Theorem

- $\mathsf{EScore}_c(\mathcal{F}) \leqslant_{\mathsf{P}} \mathsf{EV}(\mathcal{F})$ *for any* $\mathcal{F}$, c
- $\mathsf{EScore}_{c_{\mathsf{Shapley}}}(\mathcal{F}) \equiv_{\mathsf{P}} \mathsf{EV}(\mathcal{F})$ *for any* $\mathcal{F}$
- $\mathsf{EScore}_{c_{\mathsf{Banzhaf}}}(\mathcal{F}) \equiv_{\mathsf{P}} \mathsf{EV}(\mathcal{F})$ *for any* $\mathcal{F}$ *closed under conditioning and also closed under either conjunctions or disjunctions with fresh variables (e.g., d-Ds)*

What have we shown?

Theorem

- $\mathrm{EScore}_c(\mathcal{F}) \leqslant_{\mathrm{P}} \mathrm{EV}(\mathcal{F})$ *for any* $\mathcal{F}$, c
- $\mathrm{EScore}_{c_{\mathrm{Shapley}}}(\mathcal{F}) \equiv_{\mathrm{P}} \mathrm{EV}(\mathcal{F})$ *for any* $\mathcal{F}$
- $\mathrm{EScore}_{c_{\mathrm{Banzhaf}}}(\mathcal{F}) \equiv_{\mathrm{P}} \mathrm{EV}(\mathcal{F})$ *for any* $\mathcal{F}$ *closed under conditioning and also closed under either conjunctions or disjunctions with fresh variables (e.g., d-Ds)*

Proof techniques: inverting expected values and sums, decomposing sums by size of sets, polynomial interpolation

What have we shown?

Theorem

- $\mathrm{EScore}_c(\mathcal{F}) \leqslant_{\mathrm{P}} \mathrm{EV}(\mathcal{F})$ *for any* $\mathcal{F}$, c
- $\mathrm{EScore}_{c_{\mathrm{Shapley}}}(\mathcal{F}) \equiv_{\mathrm{P}} \mathrm{EV}(\mathcal{F})$ *for any* $\mathcal{F}$
- $\mathrm{EScore}_{c_{\mathrm{Banzhaf}}}(\mathcal{F}) \equiv_{\mathrm{P}} \mathrm{EV}(\mathcal{F})$ *for any* $\mathcal{F}$ *closed under conditioning and also closed under either conjunctions or disjunctions with fresh variables (e.g., d-Ds)*

Proof techniques: inverting expected values and sums, decomposing sums by size of sets, polynomial interpolation

The tractability landscape of $\mathrm{EScore}_{c_{\mathrm{Shapley}}}$ (and $\mathrm{EScore}_{c_{\mathrm{Banzhaf}}}$ under a mild condition) is exactly the same as that of EV

Exact algorithms

In the case where we have a d-D C, possible to design specific algorithms (extending those of [Deutch et al., 2022, Abramovich et al., 2023]) for EScore_c with complexity (ignoring arithmetic costs):

- $O\left(|C| \times |V|^5 + \mathrm{T}_c(|V|) \times |V|^2\right)$ where $\mathrm{T}_c(\alpha)$ is the cost of computing the coefficient function on inputs $\leqslant \alpha$

Exact algorithms

In the case where we have a d-D C, possible to design specific algorithms (extending those of [Deutch et al., 2022, Abramovich et al., 2023]) for EScore_c with complexity (ignoring arithmetic costs):

- $O\left(|C| \times |V|^5 + \text{T}_c(|V|) \times |V|^2\right)$ where $\text{T}_c(\alpha)$ is the cost of computing the coefficient function on inputs $\leqslant \alpha$
- $O\left(|V|^2 \times (|C||V| + |V|^2 + \text{T}_c(|V|))\right)$ when all probabilities are identical

Exact algorithms

In the case where we have a d-D C, possible to design specific algorithms (extending those of [Deutch et al., 2022, Abramovich et al., 2023]) for EScore_c with complexity (ignoring arithmetic costs):

- $O\left(|C| \times |V|^5 + \mathrm{T}_c(|V|) \times |V|^2\right)$ where $\mathrm{T}_c(\alpha)$ is the cost of computing the coefficient function on inputs $\leqslant \alpha$
- $O\left(|V|^2 \times (|C||V| + |V|^2 + \mathrm{T}_c(|V|))\right)$ when all probabilities are identical
- $O(|C| \times |V|)$ for c_{Banzhaf}

Application to probabilistic databases

- TID database, Boolean query q in some query language
- Define Score_c, EScore_c of a tuple for a query as Score_c, EScore_c of the Boolean provenance of the query over the database
- We compare to PQE (Probabilistic Query Evaluation, i.e., computing the probability of a Boolean query)

Theorem

- $\mathrm{EScore}_c(q) \leqslant_{\mathrm{P}} \mathrm{PQE}(q)$ *for any c, query q (whatever the query language!)*
- $\mathrm{EScore}_{c_{\mathrm{Shapley}}} \equiv_{\mathrm{P}} \mathrm{PQE}(q)$ *for any query q (whatever the query language!)*

Application to probabilistic databases

- TID database, Boolean query q in some query language
- Define Score_c, EScore_c of a tuple for a query as Score_c, EScore_c of the Boolean provenance of the query over the database
- We compare to PQE (Probabilistic Query Evaluation, i.e., computing the probability of a Boolean query)

Theorem

- $\mathrm{EScore}_c(q) \leqslant_{\mathrm{P}} \mathrm{PQE}(q)$ *for any c, query q (whatever the query language!)*
- $\mathrm{EScore}_{c_{\mathrm{Shapley}}} \equiv_{\mathrm{P}} \mathrm{PQE}(q)$ *for any query q (whatever the query language!)*

We inherit all tractability and intractability results for PQE, e.g., dichotomy for UCQs [Dalvi and Suciu, 2013] or queries closed under homomorphisms [Amarilli, 2023]

Set-up

- Implementation of all algorithms within ProvSQL
- Same experimental set-up as in [Deutch et al., 2022]: 1 GB TPC-H database, 8 TPC-H queries with some adaptations (e.g., removing aggregates), computation of Shapley/Banzhaf scores for all input tuples
- Non-Boolean queries: computation for every output tuple
- Proof-of-feasibility rather than in-depth experiments
- Compilation to d-D:
  - Check whether Boolean circuit is already an independent circuit
  - Otherwise, try to find a low-treewidth decomposition of the circuit, and use it to build a d-D
  - Otherwise, use an external knowledge compiler (but never required)

Results

# Output tuples	Provenance time (s)	Compilation time (s)	Shapley time (s) Determ.	Shapley time (s) Expect.	Banzhaf time (s)
11620	2.125	1.226	0.762	1.758	0.467
5	1.117	0.044	0.766	40.910	0.191
4	1.215	0.017	0.269	9.381	0.085
1783	1.229	0.018	0.023	0.037	0.015
61	0.174	0.001	0.001	0.002	0.001
466	0.247	0.084	0.159	0.455	0.094
91159	2.711	0.749	0.655	1.008	0.489
56	1.223	0.000	0.000	0.000	0.000

Results

# Output tuples	Provenance time (s)	Compilation time (s)	Shapley time (s) Determ.	Shapley time (s) Expect.	Banzhaf time (s)
11620	2.125	1.226	0.762	1.758	0.467
5	1.117	0.044	0.766	40.910	0.191
4	1.215	0.017	0.269	9.381	0.085
1783	1.229	0.018	0.023	0.037	0.015
61	0.174	0.001	0.001	0.002	0.001
466	0.247	0.084	0.159	0.455	0.094
91159	2.711	0.749	0.655	1.008	0.489
56	1.223	0.000	0.000	0.000	0.000

Very encouraging! Shapley value computation does not have such a huge overhead!

Main message

- Expected Shapley value computation is not (much) more costly than probabilistic query evaluation
- Landscape seems clearer than for deterministic Shapley value computation
- PQE (and Expected Shapley value computation) is quite feasible in practice, even on large datasets
- Connection to SHAP-score [Van den Broeck et al., 2022] is not quite clear (there is also a probability distribution, but not used in the same way)
- What are feasible approximations?

Bibliography I

Omer Abramovich, Daniel Deutch, Nave Frost, Ahmet Kara, and Dan Olteanu. Banzhaf values for facts in query answering, 2023. arXiv preprint arXiv:2308.05588.

Antoine Amarilli. Uniform reliability for unbounded homomorphism-closed graph queries. In *ICDT*, volume 255 of *LIPIcs*, pages 14:1–14:17, 2023.

John F Banzhaf III. Weighted voting doesn't work: A mathematical analysis. *Rutgers L. Rev.*, 19:317, 1964.

Nilesh Dalvi and Dan Suciu. The dichotomy of probabilistic inference for unions of conjunctive queries. *Journal of the ACM (JACM)*, 59(6):1–87, 2013.

Daniel Deutch, Nave Frost, Benny Kimelfeld, and Mikaël Monet. Computing the shapley value of facts in query answering. In *SIGMOD Conference*, pages 1570–1583, 2022.

Ahmet Kara, Dan Olteanu, and Dan Suciu. From shapley value to model counting and back. In *PODS*, 2024.

Bibliography II

Werner Kirsch and Jessica Langner. Power indices and minimal winning coalitions. *Social Choice and Welfare*, 34(1):33–46, 2010. ISSN 01761714, 1432217X.

Annick Laruelle. On the choice of a power index. Technical report, Instituto Valenciano de Investigaciones Económicas, 1999.

Lloyd S. Shapley et al. A value for n-person games. In Harold William Kuhn and Albert William Tucker, editors, *Contributions to the Theory of Games (AM-28), Volume II*. Princeton University Press, 1953.

Guy Van den Broeck, Anton Lykov, Maximilian Schleich, and Dan Suciu. On the tractability of SHAP explanations. *Journal of Artificial Intelligence Research*, 74:851–886, 2022.

Supplementary

Link to ProvSQL: ProvSQL

Queries used: Link to queries used