Provenance, Probabilities, and Power Indices in Databases

Pierre Senellart

ENS Paris-Saclay, 1st March 2024

Mon parcours (1/2)

2000 Entrée à l'ENS Ulm après une prépa MP

2001 Stage de L3 en Belgique sur la détection de synonymes dans le graphe d'un dictionnaire ☺ [Blondel et al., 2004]

2002 Stage de M1 en Suède sur la vérification de circuits logiques codant des multiplicateurs ☹

2002-2003 M2 de Données, IA, IHM à Orsay

2003 Stage de M2 sur la découverte par crawl des limites d'un site Web (avec S. Abiteboul, Inria) [Senellart, 2005]

2003-2004 Césure d'un an pour travailler dans l'industrie (traduction automatique) [Attnäs et al., 2005, Senellart and Senellart, 2005]

Mon parcours (2/2)

2004-2007 Thèse sur le Web caché, XML probabiliste...
(avec S. Abiteboul, Inria) [Senellart, 2007]

2007 Post-doc en Allemagne sur l'archivage du Web
[Spaniol et al., 2009]

2007 Recruté Maître de conférences à Télécom Paris

2012 Habilitation à diriger les recherches

2016 Recruté Professeur des Universités à l'ENS Ulm

Mon boulot maintenant (1/2)

Enseignement

- Cours à l'ENS (Informatique pratique L3, Bases de données L3, Projet de Recherche M1), à PSL (M2 IASD, CPES)
- Administration de l'enseignement: gestion de Master, gestion du concours d'entrée, coordination des enseignements à PSL

Recherche

- Réfléchir à des sujets intéressants (provenance, incertitude des données, crawl du Web, extraction d'information depuis des PDF. . .), trouver des solutions, les tester
- Encadrer des doctorants, stagiaires, ingénieurs, post-doctorants sur ces sujets
- Rédiger des articles avec nos résultats, les présenter
- Implémenter ça dans des logiciels, les publier, en assurer le support

Mon boulot maintenant (2/3)

Gestion de la recherche

- Responsabilité (budget, projet, rapports, personnels) d'une équipe de recherche `https://team.inria.fr/valda/`
- Adjoint du directeur du Département Informatique d'Ulm (idem, mais à l'échelle du département, et seulement adjoint)
- Président (élu) de la section 6 (Aspects symboliques de l'informatique) du Comité national de la recherche scientifique: recrutement des chercheurs CNRS, évaluation des chercheurs et des laboratoires CNRS `https://cn6.fr/`

Mon boulot maintenant (3/3)

Service à la communauté

- Comités de programme, relectures, etc. pour des conférences et journaux
- Militer pour l'accès ouvert non commercial aux publications scientifiques, à différents niveaux
- Participation aux travaux du *Comité éthique et scientifique de Parcoursup*: chaque année, étudier en profondeur certains sujets liés à Parcoursup pour un rapport (public!) au parlement

Outline

Provenance management

- Data management all about query evaluation

Provenance management

- Data management all about query evaluation
- What if we want something more than the query result?
  - Where does the result come from?
  - Why was this result obtained?
  - How was the result produced?
  - What is the probability of the result?
  - How many times was the result obtained?
  - How would the result change if part of the input data was missing?
  - What is the minimal security clearance I need to see the result?
  - What is the most economical way of obtaining the result?
  - How can a result be explained in layman terms?
  - Which part of the input contributes the most to the result?

Provenance management

- Data management all about query evaluation
- What if we want something more than the query result?
  - Where does the result come from?
  - Why was this result obtained?
  - How was the result produced?
  - What is the probability of the result?
  - How many times was the result obtained?
  - How would the result change if part of the input data was missing?
  - What is the minimal security clearance I need to see the result?
  - What is the most economical way of obtaining the result?
  - How can a result be explained in layman terms?
  - Which part of the input contributes the most to the result?
- Provenance management: along with query evaluation, record additional bookkeeping information allowing to answer (all!) the questions above

Provenance management

- Data management all about query evaluation
- What if we want something more than the query result?
  - Where does the result come from?
  - Why was this result obtained?
  - How was the result produced?
  - What is the probability of the result?
  - How many times was the result obtained?
  - How would the result change if part of the input data was missing?
  - What is the minimal security clearance I need to see the result?
  - What is the most economical way of obtaining the result?
  - How can a result be explained in layman terms?
  - Which part of the input contributes the most to the result?
- Provenance management: along with query evaluation, record additional bookkeeping information allowing to answer (all!) the questions above

Data model: annotated relations

- Relational data model: data decomposed into relations (sets or multisets of tuples), with labeled attributes...

Data model: annotated relations

- Relational data model: data decomposed into relations (sets or multisets of tuples), with labeled attributes...

name	position	city
John	Director	New York
Paul	Janitor	New York
Dave	Analyst	Paris
Ellen	Field agent	Berlin
Magdalen	Double agent	Paris
Nancy	HR director	Paris
Susan	Analyst	Berlin

Data model: annotated relations

- Relational data model: data decomposed into relations (sets or multisets of tuples), with labeled attributes…
- … with an extra provenance annotation for each tuple (think of it first as a tuple id)

name	position	city	prov
John	Director	New York	x_1
Paul	Janitor	New York	x_2
Dave	Analyst	Paris	x_3
Ellen	Field agent	Berlin	x_4
Magdalen	Double agent	Paris	x_5
Nancy	HR director	Paris	x_6
Susan	Analyst	Berlin	x_7

Queries

- A query is an arbitrary function that maps databases over a fixed database schema $\mathcal{D}$ to relations over some relational schema $\mathcal{R}$
- The query does not consider or produce any provenance annotations; we will give semantics for the provenance annotations of the output, based on that of the input
- In practice, one often restricts to specific query languages:
  - Monadic-Second Order logic (MSO)
  - First-Order logic (FO) or the relational algebra, or fragments thereof (such as UCQs, i.e., the positive relational algebra)
  - SQL with aggregate functions
  - etc.

Boolean provenance [Imieliński and Lipski, 1984]

- $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ finite set of Boolean events
- Provenance annotation: Boolean function over $\mathcal{X}$, i.e., a function of the form: $(\mathcal{X} \to \{\bot, \top\}) \to \{\bot, \top\}$
- Interpretation: possible-world semantics
  - every valuation $\nu : \mathcal{X} \to \{\bot, \top\}$ denotes a possible world of the database
  - the provenance of a tuple on ν evaluates to $\bot$ or $\top$ depending whether this tuple exists in that possible world
  - for example, if every tuple of a database is annotated with the indicator function of a distinct Boolean event, the set of possible worlds is the set of all subdatabases

Example of possible worlds

name	position	city	prov
John	Director	New York	x_1
Paul	Janitor	New York	x_2
Dave	Analyst	Paris	x_3
Ellen	Field agent	Berlin	x_4
Magdalen	Double agent	Paris	x_5
Nancy	HR director	Paris	x_6
Susan	Analyst	Berlin	x_7

$$\nu: \quad \begin{array}{ccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ \top & \top & \top & \top & \top & \top & \top \end{array}$$

Example of possible worlds

name	position	city	prov
John	Director	New York	x_1
Dave	Analyst	Paris	x_3
Magdalen	Double agent	Paris	x_5
Susan	Analyst	Berlin	x_7

$$\nu: \quad \begin{array}{ccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ \top & \bot & \top & \bot & \top & \bot & \top \end{array}$$

Boolean provenance of query results

- $\nu(D)$: the subdatabase of D where all tuples whose provenance annotation evaluates to $\bot$ by ν are removed
- The Boolean provenance $\mathrm{prov}_{q,D}(t)$ of tuple $t \in q(D)$ is the function:

$$\nu \mapsto \begin{cases} \top \text{ if } t \in q(\nu(D)) \\ \bot \text{ otherwise} \end{cases}$$

Example (What cities are in the table?)

name	position	city	prov
John	Director	New York	x_1
Paul	Janitor	New York	x_2
Dave	Analyst	Paris	x_3
Ellen	Field agent	Berlin	x_4
Magdalen	Double agent	Paris	x_5
Nancy	HR director	Paris	x_6
Susan	Analyst	Berlin	x_7

city	prov
New York	$x_1 \vee x_2$
Paris	$x_3 \vee x_5 \vee x_6$
Berlin	$x_4 \vee x_7$

Computing Boolean provenance

Theorem ([Imieliński and Lipski, 1984], Folklore)

Computing the provenance of Boolean query results for first-order queries (= the relational algebra, core of SQL) as Boolean functions is always possible and can be done in PTIME.

Proof.

Each operator of the relational algebra transforms into an operation on provenance: e.g., $\wedge$ for cross products or joins, $\vee$ for duplicate elimination, $\wedge$ for difference. . . $\square$

Beyond Boolean provenance

- Can generalize Boolean provenance to arbitrary semiring provenance [Green and Tannen, 2006], that captures more detail about query evaluation than what Boolean provenance does
- For non-monotone queries, need for semirings with monus [Geerts and Poggi, 2010], though the theory is not as clean as with regular semirings [Amsterdamer et al., 2011a]
- Possible to also give a semantics to the provenance of aggregate queries, but need to introduce an algebraic structure at the value level (provenance semimodules [Amsterdamer et al., 2011b])
- Also works for queries with recursion such as Datalog [Green and Tannen, 2006, Deutch et al., 2014] though only for some semirings [Ramusat, 2022]

Outline

Who am I?

Boolean Provenance

Representations

Probabilistic Databases

Power Indices

Conclusion

Provenance representations matter

- So far, we have shown Boolean provenance as Boolean formulas, with no constraints on the shape of the formula
- May not be the most compact representation
- May not be the most convenient representation for implementing provenance support in a database system
- May not be the most efficient representation to perform further computations from the provenance

Provenance formulas

- Quite straightforward
- Formalism used in most of the provenance literature
- PTIME data complexity
- Expanding formulas (e.g., computing a DNF or CNF representation of the provenance) can result in an exponential blowup

Example

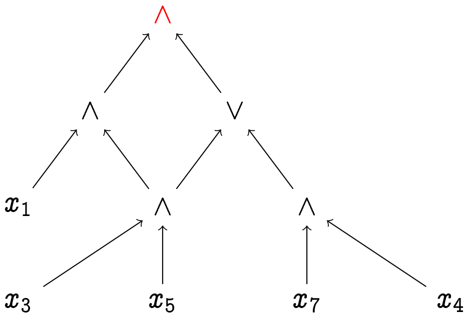
Is there a city with both an analyst and an agent, and if Paris is such a city, is there a director in the agency?

$$((x_3 \wedge x_5) \vee (x_4 \wedge x_7)) \wedge ((x_3 \wedge x_5) \wedge x_1)$$

Provenance circuits [Deutch et al., 2014, Amarilli et al., 2015]

- Use Boolean circuits to represent provenance
- Every time an operation reuses a previously computed result, link to the previously created circuit gate
- Allow linear-time data complexity of provenance computation when restricted to bounded-treewidth databases [Amarilli et al., 2015], for arbitrary MSO queries
- Formulas can be quadratically larger than provenance circuits for MSO formulas, $(\log\log)$-larger for positive relational algebra queries [Wegener, 1987, Amarilli et al., 2016]

Example provenance circuit



OBDDs and d-Ds

- Various subclasses of Boolean circuits commonly used [Darwiche and Marquis, 2002]:
  - OBDD: Ordered Binary Decision Diagrams [Bryant, 1986]
  - d-D: deterministic (= $\vee$ children are mutually exclusive) Decomposable (= $\wedge$ children are on disjoint variables) circuits
- OBDDs can be obtained in PTIME data complexity on bounded-treewidth databases [Amarilli et al., 2016]
- d-Ds can be obtained in linear-time data complexity on bounded-treewidth databases [Amarilli et al., 2016]
- d-Ds can be obtained in polynomial-time for *some* monotone first-order queries on arbitrary databases [Monet, 2020]

Outline

Probabilistic databases

[Green and Tannen, 2006, Suciu et al., 2011]

- Tuple-independent database: each tuple t in a database is annotated with independent probability $\Pr(t)$ of existing
- Probability of a possible world $D' \subseteq D$:

$$\Pr(D') = \prod_{t \in D'} \Pr(t) \times \prod_{t \in D' \setminus D} (1 - \Pr(t'))$$

- Probability of a tuple for a query q over D:

$$\Pr(t \in q(D)) = \sum_{\substack{D' \subseteq D \\ t \in q(D')}} \Pr(D')$$

- If $\Pr(x_i) := \Pr(x_i)$ where x_i is the provenance annotation of tuple x_i then $\Pr(t \in q(D)) = \Pr(\mathrm{prov}_{q,D}(t))$
- Computing the probability of a query in probabilistic databases thus amounts to computing Boolean provenance, and then computing the probability of a Boolean function
- Also works for more complex probabilistic models

Example of probability computation

name	position	city	prov	prob
John	Director	New York	x_1	0.5
Paul	Janitor	New York	x_2	0.7
Dave	Analyst	Paris	x_3	0.3
Ellen	Field agent	Berlin	x_4	0.2
Magdalen	Double agent	Paris	x_5	1.0
Nancy	HR director	Paris	x_6	0.8
Susan	Analyst	Berlin	x_7	0.2

city	prov
New York	$x_1 \vee x_2$
Paris	$x_3 \vee x_5 \vee x_6$
Berlin	$x_4 \vee x_7$

Example of probability computation

name	position	city	prov	prob
John	Director	New York	x_1	0.5
Paul	Janitor	New York	x_2	0.7
Dave	Analyst	Paris	x_3	0.3
Ellen	Field agent	Berlin	x_4	0.2
Magdalen	Double agent	Paris	x_5	1.0
Nancy	HR director	Paris	x_6	0.8
Susan	Analyst	Berlin	x_7	0.2

city	prov	prob
New York	$x_1 \vee x_2$	$1-(1-0.5)\times(1-0.7)=0.85$
Paris	$x_3 \vee x_5 \vee x_6$	1.00
Berlin	$x_4 \vee x_7$	$1-(1-0.2)\times(1-0.2)=0.36$

Complexity of probabilistic query evaluation

- Computing the probability of a query result (or of a Boolean function) is a #P-hard problem (as hard as counting the number of accepting paths of a non-deterministic polynomial-time Turing machine)
- Dichotomy result for UCQs [Dalvi and Suciu, 2012]: there is a (PTIME) algorithm that, given a UCQ, decides whether probabilistic query evaluation of this UCQ is PTIME; if not, it is #P-hard
- Computing the probability of an OBDD or a d-D representation is linear-time (ignoring the cost of arithmetic operations, PTIME otherwise)!

Outline

Motivation

- Power indices (Shapley, Banzhaf, etc.) [Laruelle, 1999]: reasonable ways to quantify the responsibility of an individual in a complex task (e.g., a variable in a Boolean function, a tuple in query evaluation)

Motivation

- Power indices (Shapley, Banzhaf, etc.) [Laruelle, 1999]: reasonable ways to quantify the responsibility of an individual in a complex task (e.g., a variable in a Boolean function, a tuple in query evaluation)
- Real data: marred with uncertainty, which may be represented by probability distributions

Motivation

- Power indices (Shapley, Banzhaf, etc.) [Laruelle, 1999]: reasonable ways to quantify the responsibility of an individual in a complex task (e.g., a variable in a Boolean function, a tuple in query evaluation)
- Real data: marred with uncertainty, which may be represented by probability distributions
- But how to assess responsibility of data items when they are both uncertain and involved in a complex task?

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)

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

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)

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

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]

Probabilistic Setting

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

Probabilistic Setting

- 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 $Z \subseteq V$,
  $\Pr(Z) \stackrel{\text{def}}{=} \left(\prod_{x \in Z} \Pr(x) \right) \times \left(\prod_{x \in V \setminus Z} (1 - \Pr(x)) \right)$

Probabilistic Setting

- 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

Probabilistic Setting

- 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
- $\mathsf{EScore}_c(\varphi, x) \stackrel{\text{def}}{=} \sum_{\substack{Z \subseteq V \\ x \in Z}} (\Pr(Z) \times \mathsf{Score}_c(\varphi, Z, x))$ 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 problem and for (Turing) polynomial-time reductions between problems, denoted $A \leqslant_{\mathsf{P}} B$, for class of Boolean functions (and $A \equiv_{\mathsf{P}} B$ for two-way reductions).

What is known?

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

What is known?

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

What is known?

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

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
- $\text{Score}_{c_{\text{Shapley}}}(\mathcal{F}) \equiv_{\text{P}} \text{EV}(\mathcal{F})$ for any class $\mathcal{F}$ closed under $\vee$-substitutions [Kara et al., 2023] and when probabilities are uniform (unweighted model counting)

Recent results [Karmakar et al., 2024]

Theorem

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

Recent results [Karmakar et al., 2024]

Theorem

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

Recent results [Karmakar et al., 2024]

Theorem

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

Recent results [Karmakar et al., 2024]

Theorem

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

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

Recent results [Karmakar et al., 2024]

Theorem

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

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

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

Expected Power Indices in 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

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

Expected Power Indices in 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

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

$\Rightarrow$ We inherit all tractability and intractability results for PQE, e.g., dichotomy for UCQs [Dalvi and Suciu, 2012] or queries closed under homomorphisms [Amarilli, 2023]

Example computation of (expected) Shapley value

Query: there exists a city with at least two persons.

name	city	Shapley value	prob	Exp. Shapley value
John	New York	0.114	0.5	0.090
Paul	New York	0.114	0.7	0.090
Dave	Paris	0.181	0.3	0.095
Ellen	Berlin	0.114	0.2	0.009
Magdalen	Paris	0.181	1.0	0.322
Nancy	Paris	0.181	0.8	0.298
Susan	Berlin	0.114	0.2	0.009

Outline

Main message

- Rich foundations of provenance management
- Connection to problems in logics, complexity theory, graph theory, algebra, knowledge compilation, game theory, etc.
- Representations matter, efficient representations are important
- Wide variety of applications are made possible by possible: probabilistic query evaluation, computation of power indices, but also enumeration of query results, sampling of results. . .
- Implementable! See `https://github.com/PierreSenellart/provsql`

Bibliography I

Omer Abramovich, Daniel Deutch, Nave Frost, Ahmet Kara, and Dan Olteanu. Banzhaf Values for Facts in Query Answering. *arXiv preprint arXiv:2308.05588*, 2023.

Antoine Amarilli. Uniform reliability for unbounded homomorphism-closed graph queries. In *ICDT*, volume 255 of *LIPIcs*, pages 14:1–14:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. URL `https://arxiv.org/abs/2209.11177`.

Antoine Amarilli, Pierre Bourhis, and Pierre Senellart. Provenance circuits for trees and treelike instances. In *Proc. ICALP*, pages 56–68, Kyoto, Japan, July 2015.

Antoine Amarilli, Pierre Bourhis, and Pierre Senellart. Tractable lineages on treelike instances: Limits and extensions. In *Proc. PODS*, pages 355–370, San Francisco, USA, June 2016.

Bibliography II

Yael Amsterdamer, Daniel Deutch, and Val Tannen. On the limitations of provenance for queries with difference. In *TaPP*, 2011a.

Yael Amsterdamer, Daniel Deutch, and Val Tannen. Provenance for aggregate queries. In *PODS*, 2011b.

Mats Attnäs, Pierre Senellart, and Jean Senellart. Integration of SYSTRAN MT systems in an open workflow. In *Proc. MT Summit*, Phuket, Thailand, September 2005.

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

Vincent D. Blondel, Anahí Gajardo, Maureen Heymans, Pierre Senellart, and Paul Van Dooren. A measure of similarity between graph vertices: applications to synonym extraction and Web searching. *SIAM Review*, 46(4):647–666, 2004.

Bibliography III

Randal E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Trans. Computers*, 35(8):677–691, 1986. doi: 10.1109/TC.1986.1676819. URL https://doi.org/10.1109/TC.1986.1676819.

Nilesh Dalvi and Dan Suciu. The dichotomy of probabilistic inference for unions of conjunctive queries. *J. ACM*, 59(6), 2012.

Adnan Darwiche and Pierre Marquis. A knowledge compilation map. *J. Artif. Intell. Res.*, 17:229–264, 2002. doi: 10.1613/JAIR.989. URL https://doi.org/10.1613/jair.989.

Daniel Deutch, Tova Milo, Sudeepa Roy, and Val Tannen. Circuits for Datalog provenance. In *ICDT*, 2014.

Bibliography IV

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

Floris Geerts and Antonella Poggi. On database query languages for k-relations. *J. Applied Logic*, 8(2), 2010.

Todd J. Green and Val Tannen. Models for incomplete and probabilistic information. *IEEE Data Eng. Bull.*, 29(1), 2006.

Tomasz Imieliński and Jr. Lipski, Witold. Incomplete information in relational databases. *J. ACM*, 31(4), 1984.

Ahmet Kara, Dan Olteanu, and Dan Suciu. From Shapley Value to Model Counting and Back. *arXiv preprint arXiv:2306.14211*, 2023.

Bibliography V

Pratik Karmakar, Mikaël Monet, Pierre Senellart, and Stéphane Bressan. Expected shapley-like scores of boolean functions: Complexity and applications to probabilistic databases. *CoRR*, abs/2401.06493, 2024. doi: 10.48550/ARXIV.2401.06493. URL https://doi.org/10.48550/arXiv.2401.06493.

Werner Kirsch and Jessica Langner. Power indices and minimal winning coalitions. *Social Choice and Welfare*, 34(1):33–46, 2010. ISSN 01761714, 1432217X. URL http://www.jstor.org/stable/41108037.

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

Bibliography VI

Mikaël Monet. Solving a special case of the intensional vs extensional conjecture in probabilistic databases. In Dan Suciu, Yufei Tao, and Zhewei Wei, editors, *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2020, Portland, OR, USA, June 14-19, 2020*, pages 149–163. ACM, 2020. doi: 10.1145/3375395.3387642. URL https://doi.org/10.1145/3375395.3387642.

Yann Ramusat. *The Semiring-Based Provenance Framework for Graph Databases*. Theses, Ecole normale supérieure - ENS PARIS ; PSL University, April 2022. URL https://inria.hal.science/tel-03896482.

Pierre Senellart. Identifying Websites with Flow Simulation. In *Proc. ICWE*, pages 124–129, Sydney, Australia, July 2005.

Bibliography VII

Pierre Senellart. *Comprendre le Web caché. Understanding the Hidden Web.* PhD thesis, Université Paris–Sud, Orsay, France, December 2007.

Pierre Senellart and Jean Senellart. SYSTRAN Translation Stylesheets: Machine Translation driven by XSLT. In *Proc. XML Conference & Exposition*, Atlanta, USA, November 2005.

Lloyd S Shapley et al. A value for n-person games. 1953.

Marc Spaniol, Dimitar Denev, Arturas Mazeika, Pierre Senellart, and Gerhard Weikum. Data Quality in Web Archiving. In *Proc. WICOW*, pages 19–26, Madrid, Spain, April 2009.

Dan Suciu, Dan Olteanu, Christopher Ré, and Christoph Koch. *Probabilistic Databases.* Morgan & Claypool, 2011.

Ingo Wegener. *The Complexity of Boolean Functions.* Wiley, 1987.