How to Bake an Uncertainty Pie

Take Even Parts of Abstract Interpretation, K-relations, and Zonotopes and Mix Thoroughly

2025-02-21

Boris Glavic

bglavic@uic.edu

DBGroup

University of Illinois, Chicago

Outline

Motivation

UIC ENGINEERING
Computer Science

Computations over uncertainty data

- **Queries** and **machine learning** training and inference
- The uncertainty will typically stem from **unrecoverability** of the ground truth clean version of dirty dataset
- Inherent complexity often necessitates approximation
  - **Over-approximation** of the set of **possible** results
  - **Under-approximation** of what is **certainly** known to be true

Connections to abstract interpretation and control theory

- Abstract interpretation [Cou96]
- Reachability analysis [ASB08]

Examples

Query evaluation

- Using K-relations and interval domains for query evaluation

Machine learning: training and inference

- Using convex polytopes for training and inference with linear models

Data Quality Issues

UIC ENGINEERING
Computer Science

Missing Values (NULLs)

Salary	Tax
35,000	3,300
	1,200
149,000	5,000

Constraint Violations

- A is a key for R

A	B
1	1
1	2
2	2
3	1
3	2

Definition (Incomplete databases)

An **incomplete database** $\boldsymbol{D}^{\odot}$ is a set of databases:

$$\boldsymbol{D}^{\odot} = \{\boldsymbol{D}_1, \boldsymbol{D}_2, \dots, \boldsymbol{D}_n\}$$

Uncertainty stemming from dirty data

Given a **"dirty" database** $\boldsymbol{D}$ we consider all **possible clean versions** as an incomplete database $\boldsymbol{D}^{\odot}$

Possible world semantics

Given some function F define its evaluation on an incomplete database, under **possible world semantics** as

$$F(\boldsymbol{D}^{\odot}) = \{F(\boldsymbol{D}_1), \dots, F(\boldsymbol{D}_n)\}$$

UIC ENGINEERING Computer Science

Definition (Certain Answers)

The **certain answers** $\textsc{certain}(Q, \boldsymbol{D}^{\odot})$ to a query Q over an incomplete database $\boldsymbol{D}^{\odot}$ are:

$$\textsc{certain}(Q, \boldsymbol{D}^{\odot}) = \bigcap_{D \in \boldsymbol{D}^{\odot}} Q(D)$$

Definition (Possible Answers)

The **possible answers** $\textsc{certain}(Q, \boldsymbol{D}^{\odot})$ to a query Q over an incomplete database $\boldsymbol{D}^{\odot}$ are:

$$\textsc{possible}(Q, \boldsymbol{D}^{\odot}) = \bigcup_{D \in \boldsymbol{D}^{\odot}} Q(D)$$

Representation Systems

Representation systems

- **representation system** [ILJ84] is a pair $(\mathbb{A}, \text{Mod})$
  - $\mathbb{A} = \{\mathcal{A}_i\}$ - the representations
  - each element $\mathcal{A}$ represents an incomplete database $\text{Mod}(\mathcal{A})$
- (A, Mod) is **closed** under classes of computations $\mathbb{F}$:

$$\forall F \in \mathbb{F} : \text{Mod}(F(\text{D}^\sharp)) = F(\text{Mod}(\text{D}^\sharp))$$

Limitations

- Some representation systems are only closed under relatively small classes of queries
  - e.g., V-tables [LJ84] not closed under selection with inequalities
- Some representation systems are not concise
  - e.g., aggregation over C-tables [LJ84] can result in exponential blowup
- Delaying complexity
  - e.g., [ADT11] handles aggregation, but extracting all worlds is hard
- PTIME is often not good enough
  - e.g., joins can degenerate into cross-products

Over-approximations Of Incomplete Databases

Relax two requirements of representation systems

1. representations are allowed to be **over-approximations**
   - assign each incomplete database $\boldsymbol{D}^{\odot}$ with a representation $\alpha(\boldsymbol{D}^{\odot}) = \mathcal{A}$
   - that can be an over-approximation: $\text{Mod}(\alpha(\boldsymbol{D}^{\odot})) \supseteq \boldsymbol{D}^{\odot}$
2. computations should **preserve** this **over-approximation**

$$\text{Mod}(F(\alpha(\boldsymbol{D}^{\odot}))) \supseteq F(\boldsymbol{D}^{\odot})$$

Under-approximating certain answers

Certain facts for an incomplete databases $\textsc{certain}(\boldsymbol{D}^{\odot}) = \bigcap_{\boldsymbol{D} \in \boldsymbol{D}^{\odot}} \boldsymbol{D}$

- now we require $\alpha(\cdot)$ to under-approximate
- extend representation system with an operation $\textsc{certain}^{\downarrow}$
- require **under-approximation**: $\textsc{certain}^{\downarrow}(\alpha(\boldsymbol{D}^{\odot})) \subseteq \textsc{certain}(\boldsymbol{D}^{\odot})$
- computations should **preserve** this **under-approximation**

$$\textsc{certain}^{\downarrow}(F(\alpha(\boldsymbol{D}^{\odot}))) \subseteq \textsc{certain}(F(\boldsymbol{D}^{\odot}))$$

Outline

Abstract Domains and Transformers

Definition (Abstract Domain)

Given a **concrete domain** $\mathbb{D}$, an **abstract domain** is a set $\mathbb{D}^\sharp$ with two operations:

- **abstraction** $\alpha : \mathcal{P}(\mathbb{D}) \rightarrow \mathbb{D}^\sharp$
- **concretization**: $\gamma : \mathbb{D}^\sharp \rightarrow \mathcal{P}(\mathbb{D})$

such that for any set $S \subseteq \mathcal{P}(\mathbb{D})$

$$\gamma(\alpha(S)) \supseteq S$$

Definition (Abstract Transformers)

Given a function $F : \mathbb{D}_1 \rightarrow \mathbb{D}_2$ and abstract domains $\mathbb{D}_1^\sharp$ and $\mathbb{D}_2^\sharp$ an **abstract transformer** $F^\sharp : \mathbb{D}_1^\sharp \rightarrow \mathbb{D}_2^\sharp$ for F has to fulfill for any $S \subseteq \mathcal{P}(\mathbb{D}_1)$:

$$\gamma\left(F^\sharp(\alpha(S))\right) \supseteq F(S)$$

Example: Interval Arithmetic

Interval Domain

- concrete domain: $\mathbb{R}$
- abstract domain: $\mathbb{R}_I = \{[l, u] \mid l \leq u \wedge l, u \in \mathbb{R}\}$
- abstraction: for $S \subseteq \mathbb{R} : \alpha(S) = [inf\ S, sup\ S]$
- concretization: $\gamma([l.u]) = \{c \mid c \in [u, l]\}$

Interval Arithmetic

- $[a, b] + [c, d] = [a + c, b + d]$
- $[a, b] - [c, d] = [a - d, b - c]$
- $[a, b] \cdot [c, d] = [min(ac, ad, bc, bd), max(ac, ad, bc, bd)]$

[dFS04]

Example: Zonotopes

Zonotope Domain

- concrete domain: $\mathbb{R}^n$
- abstract domain
  - set of variables $\mathcal{E}$
  - affine forms: $\mathbb{A} = \{a_0 + \sum_{\epsilon_i \in \mathcal{E}} a_i \cdot \epsilon_i \mid a_i \in \mathbb{R}\}$ with finite support
  - zonotopes: $\mathcal{Z}_n{}^\sharp = \{\mathbf{z} \mid \mathbf{z} \in \mathbb{A}^n\}$
- abstraction: $\alpha(S) = \mathcal{IH}(S)$ (interval hull)
- concretization: $\gamma(\mathbf{z}) = \{\varphi(\mathbf{z}) \mid \varphi$ is a valuation$\}$
  - valuation: $\varphi : \mathcal{E} \to [-1, 1]$

Zonotopes: Affine Arithmetic

UIC ENGINEERING
Computer Science

Affine Arithmetic

- zonotopes
  - $z_1 = a_0 + \sum_{\epsilon_i \in \mathcal{E}} a_i \cdot \epsilon_i$
  - $z_2 = b_0 + \sum_{\epsilon_i \in \mathcal{E}} b_i \cdot \epsilon_i$
- addition: $z_1 + z_2 = a_0 + b_0 + \sum_{\epsilon_i \in \mathcal{E}} a_i \cdot b_i \cdot \epsilon_i$
- multiplication: $z_1 \cdot z_2 = a_0 b_0 + (\sum_{\epsilon_i \in \mathcal{E}} (a_0 \cdot b_i + a_i \cdot b_0) \cdot \epsilon_i) + c \cdot \epsilon_{new}$
  - $c = (\sum_i |a_i|) \cdot (\sum_i |b_i|)$

Outline

Naturally Ordered Semirings

Natural order

- Semiring $(K, +_K, \cdot_K, 0_K, 1_K)$
- Define $k_1 \leq_K k_2 \Leftrightarrow \exists k_3 : k_1 + k_3 = k_2$
- A semiring is **naturally ordered** if $\leq_K$ is a partial order

Properties of the natural order

- natural order is preserved under semiring operations:
- **addition**: $a \leq_K c \wedge b \leq_K d \Rightarrow a +_K b \leq_K c +_K d$
- **multiplication**: $a \leq_K c \wedge b \leq_K d \Rightarrow a \cdot_K b \leq_K c \cdot_K d$

Incomplete K-relations

UIC ENGINEERING
Computer Science

Definition (Incomplete K-databases)

An incomplete K database $\boldsymbol{D}^{\odot}$ is a set of K databases $\{\boldsymbol{D}_1, \ldots, \boldsymbol{D}_n\}$

$\boldsymbol{D}_1$

name	salary	
Boris	120k	1
Peter	150k	1
Peter	380k	2

$\boldsymbol{D}_2$

name	salary	
Peter	180k	2

K-relations as Abstract Domains

UIC ENGINEERING
Computer Science

Abstract Domain

- Assume a naturally ordered semiring K, then we can use K databases as abstract domains [FHGK19],
- **abstraction**

$$\forall t : \alpha(\boldsymbol{D}^{\odot})(t) = \sup_{\boldsymbol{D} \in \boldsymbol{D}^{\odot}} \boldsymbol{D}(t)$$

- **concretization**

$$\gamma(D) = \{D' \mid \forall t : D'(t) \leq_K D(t)\}$$

Abstract Transformers for Query

Abstract Transformers for Relational Algebra Operators

- Under the standard K-relational semantics for positive relational algebra [GT17], operators are abstract transformers.
- This follows from the preservation of natural order under semiring operations

K-relations with Interval Domain Values

UIC ENGINEERING
Computer Science

- Use interval domain values to encode value uncertainty
- Also for aggregation results [ABC+03, AK08, DK22]

name	salary	
Boris	[120k,120k]	[0,2]
Peter	[140k,400k]	[2,3]

Over-approximating Possible Worlds

Selection

- requires interval arithmetic and embedding of Boolean intervals into the semiring K

Difference

- to be useful requires both upper and lower bounds [GL17]

Aggregation

- for $\mathbb{N}$ there exist an abstract transformer for semi-module expressions [ADT11]

TLDR

UIC ENGINEERING
Computer Science

What has been achieved?

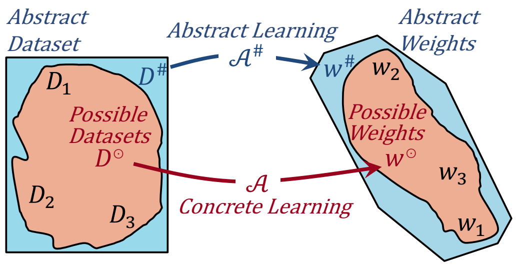
- a semantics for relational algebra over uncertain data with PTIME data complexity
  - closed under full relational algebra with aggregation and order-based operations (e.g., windowed aggregation)
- mechanisms to approximate repairs to common data quality issues and approximate common incomplete and probabilistic data models
- uniform treatment of aggregation results and value uncertainty
- Approximating certain and possible answers
  - under-approximation of certain answers with value uncertainty
  - over-approximation of possible answers with value uncertainty
- Over-approximation of the set of possible worlds

Outline

UIC ENGINEERING Computer Science

Setting

- uncertain training $\boldsymbol{D}^{\odot}$ with features $\boldsymbol{X}^{\odot}$ and labels and test datasets $X_{test}^{\odot}$
- consider linear models trained with ridge regression (l_2 regularization)
- train an over-approximation of all possible models
- compute an over-approximation of all possible inference outcomes



Uncertain Training and Test Data

UIC ENGINEERING
Computer Science

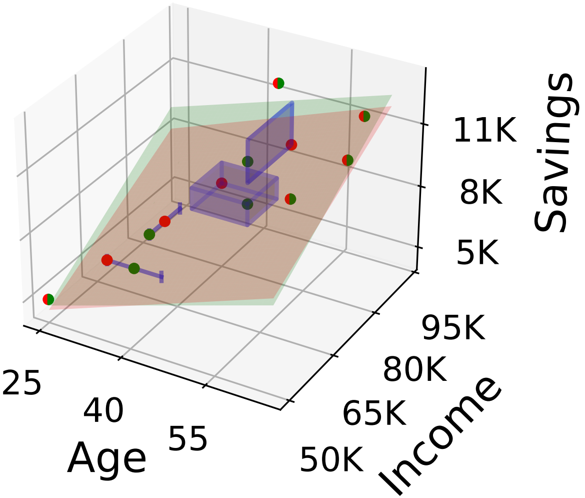
Training Data

Age	Income	Savings
25	50K	5K
NULL	60K	6K
35	NULL	7K
NULL	NULL	[8K,9K]
45	90K	12K
50	NULL	[10K,12K]
55	75K	9K
60	85K	10K
65	80K	13K

Test Data

Age	Income	Savings
25	50K	??
40	60K	??
20	90K	??
70	50K	??

UIC ENGINEERING
Computer Science



● Possible World 1 ● Possible World 2

Sav. = -2855 + 98×Age + 0.1×Inc.

Sav. = -3361 + 84×Age + 0.1×Inc.

- **Each training data world D_i induces a model!**
  - $\boldsymbol{w}_i^* = \mathcal{A}(D_i)$

Possible Predictions

The difference in models leads to a difference in predictions

Predictions in world D_1 with $\boldsymbol{w}_1^*$

$$sav. = -2855 + 98 \cdot Age + 0.1 \cdot Inc.$$

Age	Income	Savings
25	50K	4595
40	60K	7065
20	90K	8105
70	50K	9005

Predictions in world D_2 with $\boldsymbol{w}_2^*$

$$sav. = -2855 + 84 \cdot Age + 0.1 \cdot Inc.$$

Age	Income	Savings
25	50K	4245
40	60K	6505
20	90K	7825
70	50K	8025

Abstract Gradient Decent

Fixed Point Gradient Descent

- Model weights $\boldsymbol{w}$
- Loss L
- Learning Rate η

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} + \left(-\eta \cdot \nabla L(\boldsymbol{w}^{(t)})\right)$$

Abstract Gradient Descent

- Exact abstract transformers exist for all operations of gradient descent for linear regression
  - requires polynomial zonotopes

$$\boldsymbol{w}^{\sharp(t+1)} = \boldsymbol{w}^{\sharp(t)} + \left(-\eta \odot \nabla^{\sharp} L^{\sharp}(\boldsymbol{w}^{\sharp(t)})\right)$$

Equivalence-Based Abstract Fixed Points

Abstract Fixed Points (Equivalence)

$$\gamma\left(\boldsymbol{w}^{\sharp *}\right) = \gamma\left(\boldsymbol{w}^{\sharp *} + \left(-\eta \cdot \nabla^{\sharp} L^{\sharp}(\boldsymbol{w}^{\sharp (t)})\right)\right)$$

Existence of Fixed Points

- Abstract fixed points exists
- Reached once all concreted gradient descent processes have converged

Challenges

- The size of the representation grows exponentially in the number of gradient descent steps

Ridge Regression

Ridge regression

$$L(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{w}) = \frac{1}{n}\sum_{i=1}^{n}(y'_i - y_i)^2 = \frac{1}{n}(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})^T(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})$$

Gradient Descent

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t - \eta\frac{2}{n}(\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w}^t - \boldsymbol{X}^T\boldsymbol{y}) + 2\lambda\boldsymbol{w}^t$$

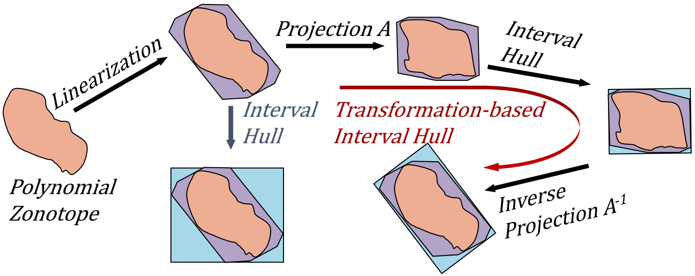
UIC ENGINEERING
Computer Science

Order-Reduction

- **Order-reduction** over-approximates a zonotope with another zonotope of smaller representation size

Linearization

- **Linearization** over-approximates polynomial zonotopes with linear zonotopes



Challenges

- Fixed point may not exist

Abstract Closed Form

Abstract gradient descent with order-reduction and linearization

$$\boldsymbol{w}^{\sharp(t+1)} = \boldsymbol{R}\left(\boldsymbol{w}^{\sharp(t)} + \boldsymbol{L}\left(-\eta \cdot \nabla^{\sharp} L^{\sharp}(\boldsymbol{w}^{\sharp(t)})\right)\right)$$

Decomposition

- Decomposition: $\boldsymbol{w}^{\sharp *} = \boldsymbol{w}_R^{*} + \boldsymbol{w}_D^{\sharp *} + \boldsymbol{w}_N^{\sharp *}$ is a fixed point if:

$$\boldsymbol{w}_R^{*} = \Phi_R(\boldsymbol{w}_R^{*}), \qquad \boldsymbol{w}_D^{\sharp *} = \Phi^{\sharp}{}_D(\boldsymbol{w}_R^{*}, \boldsymbol{w}_D^{\sharp *}) \qquad \boldsymbol{w}_N^{\sharp *} \simeq_{\sharp} \Phi^{\sharp}{}_N(\boldsymbol{w}_R^{*}, \boldsymbol{w}_D^{\sharp *}, \boldsymbol{w}_N^{\sharp *})$$

- Choose order-reduction to construct fixed point

Outline

Conclusions

Abstract Interpretation

- Over-approximate sets of *"possible worlds"*
- Abstract transformers: computations that preserve the over-approximation

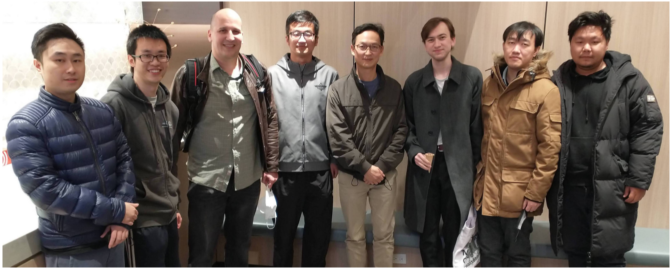
Applications to machine learning

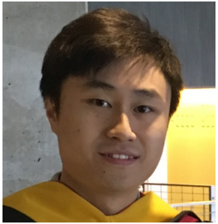
- Can we go beyond linear models

Applications to databases

- Databases with zonotope values: **z-tables**
- What about annotations?
  - can we model correlation

Group & Collaborators





Outline

[ABC$^+$03] Marcelo Arenas, Leopoldo Bertossi, Jan Chomicki, Xin He, Vijay Raghavan, and Jeremy Spinrad.
Scalar aggregation in inconsistent databases.
Theoretical Computer Science, 296(3):405–434, 2003.

[ADT11] Yael Amsterdamer, Daniel Deutch, and Val Tannen.
Provenance for aggregate queries.
In *Proceedings of the thirtieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 153–164. ACM, 2011.

[AK08] Foto N. Afrati and Phokion G. Kolaitis.
Answering aggregate queries in data exchange.
In *Proceedings of the Twenty-Seventh ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2008, June 9-11, 2008, Vancouver, BC, Canada*, pages 129–138, 2008.

[ASB08] Matthias Althoff, Olaf Stursberg, and Martin Buss.
Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization.
In *Proceedings of the 47th IEEE Conference on Decision and Control, CDC 2008, December 9-11, 2008, Cancún, Mexico*, pages 4042–4048. IEEE, 2008.

[Cou96] P. Cousot.
Abstract interpretation.
ACM Computing Surveys (CSUR), 28(2):324–328, 1996.

[dFS04] Luiz Henrique de Figueiredo and Jorge Stolfi.
Affine arithmetic: Concepts and applications.
Numer. Algorithms, 37(1-4):147–158, 2004.

References II

[DK22] Akhil A. Dixit and Phokion G. Kolaitis.
Consistent answers of aggregation queries via SAT.
In *38th IEEE International Conference on Data Engineering, ICDE 2022, Kuala Lumpur, Malaysia, May 9-12, 2022*, pages 924–937. IEEE, 2022.

[FHGK19] Su Feng, Aaron Huber, Boris Glavic, and Oliver Kennedy.
Uncertainty annotated databases - a lightweight approach for approximating certain answers.
In *Proceedings of the 44th International Conference on Management of Data*, pages 1313–1330, 2019.

[GL17] Paolo Guagliardo and Leonid Libkin.
Correctness of sql queries on databases with nulls.
ACM SIGMOD Record, 46(3):5–16, 2017.

[GT17] Todd J Green and Val Tannen.
The semiring framework for database provenance.
In *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 93–99. ACM, 2017.

[ILJ84] Tomasz Imieliński and Witold Lipski Jr.
Incomplete Information in Relational Databases.
Journal of the ACM (JACM), 31(4):761–791, 1984.

[LJ84] Witold Lipski Jr.
On relational algebra with marked nulls.
In *Proceedings of the 3rd ACM SIGACT-SIGMOD symposium on Principles of database systems*, pages 201–203. ACM, 1984.