ULDBs: Databases with Uncertainty and Lineage

Omar Benjelloun
Anish Das Sarma
Alon Halevy
Jennifer Widom

Presented by: Sindhura Tokala

Background

- Results of IR queries are *ranked* and *uncertain*

Example:

```
SELECT   *
FROM     Actor A
WHERE    A.name ≈ 'Kevin'
   and   1995 =
         SELECT   MIN(F.year)
         FROM     Film F, Casts C
         WHERE    C.filmid = F.filmid
                  and C.actorid = A.actorid
                  and F.rating ≈ "high"
```

Example

```
SELECT   *
FROM     Actor A
WHERE    A.name ≈ 'Kevin'
  and    1995 =
         SELECT   MIN(F.year)
         FROM     Film F, Casts C
         WHERE    C.filmid = F.filmid
                  and C.actorid = A.actorid
                  and F.rating ≈ "high"
```

Probabilistic Databases

- Each tuple has a probability of belonging to the database

Example

$S^p =$

	A	B	
s_1	'm'	1	0.8
s_2	'n'	1	0.5

$T^p =$

	C	D	
t_1	1	'p'	0.6

Possible Worlds

$S^p =$

	A	B	
s_1	'm'	1	0.8
s_2	'n'	1	0.5

$T^p =$

	C	D	
t_1	1	'p'	0.6

$pwd(D^p) =$

database instance	probability
$D_1 = \{s_1, s_2, t_1\}$	0.24
$D_2 = \{s_1, t_1\}$	0.24
$D_3 = \{s_2, t_1\}$	0.06
$D_4 = \{t_1\}$	0.06
$D_5 = \{s_1, s_2\}$	0.16
$D_6 = \{s_1\}$	0.16
$D_7 = \{s_2\}$	0.04
$D_8 = \phi$	0.04

Query evaluation on probabilistic Databases

- Consider the query:

$$q(u) : -S^p(x, y), T^p(z, u), y = z$$ **on**

$S^p =$

	A	B	
s_1	'm'	1	0.8
s_2	'n'	1	0.5

$T^p =$

	C	D	
t_1	1	'p'	0.6

$q^{pwd}(D^p) =$

answer	probability
$\{'p'\}$	0.54
$\emptyset$	0.46

Problems with such data:

- Handling “Data Lineage”
- Handling “Uncertainty”

Data Lineage

- Metadata Management
- Functionality of determining:
  - Where the data came from
  - How it is transformed
  - Where it is going

Lineage => Resolve Uncertainty

- Example: Search Engines

Uncertainty::Ranking

Lineage::Text Snippet

Example:

- Two sets of base data: A and B
- Only one base set is correct
- Derived queries should not produce data that is mixed from A and B
- Lineage helps encode relationships that arise between base and derived data.

Agenda

- Define ULDBs: uncertain databases with lineage
- Combine Lineage and Uncertainty
- Querying ULDBs
- Extend ULDBs with confidence values.

Problem Setup:

- Database *D*
- Set of relations $\bar{R}$ = R_1, R_2,..... R_n
- Each R_i is a multi-set of tuples
- Each tuple has a unique identifier
- $I(\bar{R})$ denotes identifiers in relations R_1, R_2,..... R_n

Databases with Lineage(LDB)

- Lineage of a tuple identifies the data from which it was derived
- External Lineage: derived from outside the LDB
- Internal Lineage: references to other tuples in the LDB

LDB Representation

- Triple ($\bar{R}$, S, λ), where
- $\bar{R}$: set of relations
- S: set of symbols containing I($\bar{R}$)
- λ: lineage function from S to 2^S

Example

Saw

ID	witness	car
21	Amy	Mazda
22	Amy	Toyota
23	Betty	Honda

Drives

ID	person	car
31	Jimmy	Mazda
32	Jimmy	Toyota
33	Billy	Mazda
34	Billy	Honda

Accuses

ID	witness	person
41	Amy	Jimmy
42	Amy	Jimmy
43	Amy	Billy
44	Betty	Billy

$\lambda(41) = \{21, 31\}$
$\lambda(42) = \{22, 32\}$
$\lambda(43) = \{21, 33\}$
$\lambda(44) = \{23, 34\}$

- Drives(person,car)
- Saw(witness,car)
- Accuses(witness,person) : from the query $\pi_{witness,person}(Saw \bowtie Drives)$

Uncertain Database Representation

- X-tuples
- X-relations

Definitions

- x-tuple: multiset of one or more tuples, called *alternatives* (mutually exclusive values for the tuple)
- maybe x-tuple: tuples that may be present or absent.

ID	Saw(witness, car)
21	(Amy,Mazda) || (Amy,Toyota)
23	(Betty,Honda)

?

Definitions

- x-relation: multiset of x-tuples.
- Construction:
  - Pick one alternative from each x-tuple that is not a maybe-x-tuple
  - Pick zero or one alternative from each x-tuple that is a maybe-x-tuple.

ID	Saw(witness, car)
21	(Amy,Mazda) || (Amy,Toyota)
23	(Betty,Honda)

?

Combining Lineage and Uncertainty

ULDB

- Triple ($\bar{R}$, S, λ)
- $\bar{R}$: set of x-relations
- S : set of symbols containing I($\bar{R}$)
- λ : lineage function from S to 2^S
- I($\bar{R}$): tuple alternatives *(i, j)*
  - *i*: x tuple
  - *j*: index of one of its alternatives

ID	Saw(witness, car)	
21	(Amy,Mazda) || (Amy,Toyota)	?
23	(Betty,Honda)	

ID	Drives(person, car)
31	(Jimmy,Mazda)
32	(Jimmy,Toyota)
33	(Billy,Mazda)
34	(Billy,Honda)

ID	Accuses(witness, person)		
41	(Amy,Jimmy)	?	$\lambda(41,1)=\{(21,1),(31,1)\}$
42	(Amy,Jimmy)	?	$\lambda(42,1)=\{(21,2),(32,1)\}$
43	(Amy,Billy)	?	$\lambda(43,1)=\{(21,1),(33,1)\}$
44	(Betty,Billy)	?	$\lambda(44,1)=\{(23,1),(34,1)\}$

Accuses with Uncertainty and Lineage

LDB of a ULBD *D*

- D_k is a possible LDB of *D*, $S_k \subseteq S$
- D_k is the triple (R_k, S_k, λ_k)
- R_k includes exactly the alternatives of x-tuples in $\bar{R}$ such that $s_{(i,j)} \in S_k$ and λ_k is the restriction of λ to S_k

Conditions

- Let $s_{(i,\ j)} \in S_k$, then for every $j' \neq j$, $s(i,j') \notin S_k$
- $\forall\ s_{(i,\ j)} \in S_k, \lambda(s_{(i,\ j)}) \subseteq S_k$
- If for some x-tuple t_i there does not exist a $s_{(i,j)} \in S_k$, then t_i is a maybe x-tuple, and $\forall s_{(i,j)} \in t_i$, $\lambda(s_{(i,j)}) = \emptyset$ or $\lambda(s_{(i,j)}) \notin S_k$

Example:

ID	Saw(witness, car)
21	(Amy,Mazda) || (Amy,Toyota)
23	(Betty,Honda)

?

ID	Drives(person, car)
31	(Jimmy,Mazda)
32	(Jimmy,Toyota)
33	(Billy,Mazda)
34	(Billy,Honda)

ID	Accuses(witness, person)		
41	(Amy,Jimmy)	?	$\lambda(41,1)=\{(21,1),(31,1)\}$
42	(Amy,Jimmy)	?	$\lambda(42,1)=\{(21,2),(32,1)\}$
43	(Amy,Billy)	?	$\lambda(43,1)=\{(21,1),(33,1)\}$
44	(Betty,Billy)	?	$\lambda(44,1)=\{(23,1),(34,1)\}$

- Let $s_{(i,j)} \in S_k$, then for every $j' \neq j$, $s(i,j') \notin S_k$
- $\forall s_{(i,j)} \in S_k, \lambda(s_{(i,j)}) \subseteq S_k$
- If for some x-tuple t_i there does not exist a $s_{(i,j)} \in S_k$, then t_i is a maybe x-tuple, and $\forall s_{(i,j)} \in t_i, \lambda(s_{(i,j)}) = \emptyset$ or $\lambda(s_{(i,j)}) \notin S_k$

Completeness for ULDBs

- Given any set of possible LDBs P = {P1, P2, . . . , Pm} over relations R = {R1,R2, . ,Rn}, there exists a ULDB D = (R, S, λ) whose possible LDBs are P.

Well-Behaved Lineage

The lineage of an x-tuple t_i is well-behaved if it satisfies the following three conditions:

1. Acyclic: $\forall\, s_{(i,j)},\ s_{(i,j)} \notin \lambda^*(s_{(i,j)})$
2. Deterministic: $\forall\, s_{(i,j)},\ s_{(i,j')}$, if $j \neq j'$ then either $\lambda(s_{(i,j)}) \neq \lambda(s_{(i,j')})$ or $\lambda(s(i,j)) = \varnothing$.

3. Uniform: $\forall\, s_{(i,j)},\ s_{(i,j')},\ B(s_{(i,j)}) = B(s(i, j'))$,

where $B(s(i,j)) = \{t_k \mid \exists s_{(k,l)},\ s_{(k,l)} \in \lambda(s(i, j))\}$

Conditions:

1. There are no cycles;
2. All alternatives of an x-tuple have distinct lineage; and
3. Their lineage points to alternatives of the exact same set of x-tuples.

Querying ULDBs

DL-monotonic query

Function Q from LDBs to LDBs that satisfies the following conditions:

- Constrains the lineage of a result tuple to be a minimal subset of the database that produces exactly that tuple.
- Enforces monotonicity on both data and lineage.

DL Monotonic Example

Saw

ID	witness	car
21	Amy	Mazda
22	Amy	Toyota
23	Betty	Honda

Drives

ID	person	car
31	Jimmy	Mazda
32	Jimmy	Toyota
33	Billy	Mazda
34	Billy	Honda

Accuses

ID	witness	person
41	Amy	Jimmy
42	Amy	Jimmy
43	Amy	Billy
44	Betty	Billy

$\lambda(41) = \{21, 31\}$
$\lambda(42) = \{22, 32\}$
$\lambda(43) = \{21, 33\}$
$\lambda(44) = \{23, 34\}$

ULDB Minimality

Data Minimality

- An alternative (i, j) of an x-tuple t_i in a ULDB D is said to be extraneous if removing it from the x-relation does not change the possible instances of D.
- a '?' on an x-tuple in D is said to be extraneous if removing it does not change the possible instances of D.
- A ULDB D is D-minimal if it does not include any extraneous alternatives or '?'s.

Example

ID	Saw(witness, car)
1	(Carol,Acura) || (Carol,Lexus)

ID	Car1(car)
2	Acura

ID	Car2(car)
3	Lexus

Saw1 = (Car1 ⋈ Saw) Saw2 = (Car2 ⋈ Saw)

ID	Saw1(witness,car)
4	(Carol,Acura)

ID	Saw2(witness,car)
5	(Carol,Lexus)

$\lambda(4,1)=\{(1,1),(2,1)\}$ $\lambda(5,1)=\{(1,2),(3,1)\}$

$(Saw1 \bowtie_{witness} Saw2)$

ID	(witness,car1,car2)
6	(Carol,Acura,Lexus)

? $\lambda(6,1)=\{(4,1),(5,1)\}$ → Extraneous

Lineage Minimality

- A ULDB D = ($\bar{R}$, S, λ) is L-minimal if for any D′ = ($\bar{R}$, S′, λ′) over the same x-relations $\bar{R}$ such that:
  - S′ ⊆ S, λ′* ⊆ λ*
  - D and D′ have the same internal lineage
- D′ has the same possible instances as D only if S′ = S and λ′∗ = λ∗.

Membership Queries

- Tuple Membership: determine weather t ϵ R, in some (resp. all) possible instance(s) of D.
- Instance Membership: Given a ULDB D containing a relation R, and a multiset T of tuples, determine whether R contains exactly the tuples of T in some (or all) possible instance(s) of D.

Extraction: Extract a subset of the database

- Let D be a well-behaved ULDB with x-relations R and possible instances P, and let $\bar{X}$ be a subset of R.
- The problem of extracting $\bar{X}$ from R is to return a well-behaved ULDB D′ with R′ = $\bar{X}$ and possible instances P′, such that the restriction of P to $\bar{X}$ equals P′ with respect to data and internal lineage

Confidences and Probabilistic Data

Confidences and Probabilistic Data

- Each base alternative *a* is associated with a confidence value *c(a)*
- The probability of a possible instance is the product of the confidences of the base alternatives and '?' chosen in it

Example

ID	Saw(witness, car)
11	(Amy,Acura):0.8
12	(Betty,Acura):0.4 || (Betty,Mazda):0.6

?

ID	Drives(person, car)
51	(Hank,Acura):0.6

?

- Instance where Amy saw an Acura, Betty saw a Mazda, and Hank does not drive an Acura has confidence 0.8 * 0.6 * (1 - 0.6) ≈ 0.20

Query Processing

- **Data Computation:** in which we compute the data and lineage in query results, just as in ULDBs without confidences
- **Confidence Computation:** in which we compute confidence values for query results based on their lineage (and confidence values on base data)

Related Work

- Query answering in probabilistic databases.
- Approximate query answering.
- Integrating lineage (provenance)

Conclusions

- ULDBs can represent any finite set of possible instances containing data and lineage.
- ULDBs can be extended naturally to represent and query probabilistic data.
- ULDB allows query evaluation to be decoupled from computation of confidences

Future Work

- Algorithms and Optimizations when computing confidences
- On-demand confidence computation
- Incremental propagation of confidence updates
- Ordering queries based on confidences
- Efficient update algorithms
- Implementation, Theory.
- Incomplete relations, versioning of data, uncertainty and lineage

References

1. Dan Suciu and Nilesh Dalvi, “Foundations of Probabilistic Answers to Queries”, SIGMOD 2005 Tutorial.