CMSC 724: Uncertainty in Databases

Amol Deshpande

University of Maryland, College Park

September 1, 2010

Outline

1. Overview
2. Probabilistic Databases

Uncertainty

- Relational databases assume “exact” data
- Increasingly breaking down
  - “Sensor” data
    - Physical/hardware sensors
    - Also data observed by people (called “persono-scopes”)
  - Data integration type of applications
    - Confidence/trust/“quality” issues
    - Machine generated schema mappings
    - Automatically extracted information/knowledge
  - Reasoning using Machine Learning techniques
    - Many classifiers provide probabilities of belonging to different classes
  - Data may be exact, but queries may not be
    - inst-name like “UMD” (does this match “Univ of MD” ?)
    - Common in Information Retrieval

Types of uncertainty

- "Nulls" (very interesting early work on this)
- More generally, missing values/incomplete databases
- Existence uncertainty: Tuples may or may not exist
  - Can be modeled logically, probabilities, fuzzy logic etc...
- Attribute values are uncertain
  - I saw a car, but not sure of the color
  - Sensor values have "noise" in them
- Approximations/Summarization
- Confidence/Trust/Quality
- "Correlations"
  - Either this tuple exists or that
  - If this attribute is 1, then that attribute is 2

Lineage/Data Provenance

- Another thing databases don't do well
- Goal: Trace back a query result to original data
- Provenance/Lineage
  - Carry around enough information during query processing to trace back

Lineage/Data Provenance

- Another thing databases don't do well
- Goal: Trace back a query result to original data
- Provenance/Lineage
  - Carry around enough information during query processing to trace back
- Very important in sensor applications
  - Vision: Highly automated systems that process vast amounts of data to decide what to do next
  - Large amounts of aggregation, distributed processing
  - A small mistake/noise can cause large errors later on
  - Custom-build sensor processing code makes it harder

Uncertainty Management

- Approach
  - Lets assume relational database for now
  - Attach numbers with the tuples or attribute values
  - Propagate numbers through the operations
- Key questions:
  - How to operate on the numbers ?
    - Probability Theory is a natural choice, but not the only one
    - Rich literature on other types of operations
  - How to do query processing ?
    - Actually very challenging...
    - Concrete results known only under simplifying assumptions

Outline

Probabilistic Databases

- Use probabilities to model the uncertainty
- Use laws of probability to process the data
- Tuple-level uncertainty
  - All attributes in a tuple are known precisely
  - Existence of the tuple is uncertain
- Attribute-level uncertainty
  - Tuples (identified by "keys") exist for certain
  - Exact value of an attribute is uncertain

A Tuple-level Uncertainty Model

- Proposed by Fuhr et al. in Information Retrieval Context
  - Later work by Dalvi and Suciu, by us here at UMD (Prithviraj Sen)
  - Examples from Dalvi and Suciu, VLDB 2004
- A simple probabilistic database

$S^p =$

	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

- The best way to think of probabilistic databases
- A world is a fixed set of tuples – no uncertainty
- Assuming "independence":

$S^p =$

	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

- The best way to think of probabilistic databases
- A world is a fixed set of tuples – no uncertainty
- Assuming "independence":

$S^p =$

	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

- Evaluate on each possible world
- Combine the results together somehow
  - Semantics of this can be tricky
- Example: $\pi_D(S \bowtie_{B=C} T)$

$S^p =$

	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	Result
$D_1 = \{s_1, s_2, t_1\}$	0.24	{'p'}
$D_2 = \{s_1, t_1\}$	0.24	{'p'}
$D_3 = \{s_2, t_1\}$	0.06	{'p'}
$D_4 = \{t_1\}$	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	{}

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

Query Evaluation

- Another Example: $\pi_A(S \bowtie_{B=C} T)$

$S^p =$

	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	Result
$D_1 = \{s_1, s_2, t_1\}$	0.24	{'m','n'}
$D_2 = \{s_1, t_1\}$	0.24	{'m'}
$D_3 = \{s_2, t_1\}$	0.06	{'n'}
$D_4 = \{t_1\}$	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	{}

=

answer	probability
$\{'m','n'\}$	0.24
$\{'m'\}$	0.24
$\{'n'\}$	0.06
$\emptyset$	0.46

Query Evaluation

- Can't convert to possible worlds and evaluate the query
  - Too many possible worlds
  - With just 20 uncertain tuples, number of worlds is at least $2^{20} = 1$ million.

Query Evaluation

- Can't convert to possible worlds and evaluate the query
  - Too many possible worlds
  - With just 20 uncertain tuples, number of worlds is at least $2^{20} = 1$ million.
- Extensional semantics
  - Combine probabilities when tuples are put together
  - Issues with correlations
- Intensional semantics
  - Represent each result as a boolean formula
  - Evaluate it at end
  - #P-Complete

Extensional: Example 1

- $\pi_D(S \bowtie_{B=C} T) \rightarrow$ Do Join first, then Projection

$S^p =$

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

$T^p =$

	C	D	
t_1	1	'p'	0.6

Extensional: Example 1

- $\pi_D(S \bowtie_{B=C} T) \rightarrow$ Do Join first, then Projection

$S^p =$

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

$T^p =$

	C	D	
t_1	1	'p'	0.6

Joins: assume independence

A	B	C	D	*prob*
'm'	1	1	'p'	0.8*0.6 = 0.48
'n'	1	1	'p'	0.5*0.6 = 0.30

Extensional: Example 1

- $\pi_D(S \bowtie_{B=C} T) \rightarrow$ Do Join first, then Projection

$S^p =$

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

$T^p =$

	C	D	
t_1	1	'p'	0.6

Joins: assume independence

A	B	C	D	*prob*
'm'	1	1	'p'	0.8*0.6 = 0.48
'n'	1	1	'p'	0.5*0.6 = 0.30

Projection: union probability; assume independence

D	*prob*
'p'	(1 - (1 - 0.48)(1 - 0.3)) = 0.636

Extensional: Example 1

- $\pi_D(S \bowtie_{B=C} T) \rightarrow$ Do Join first, then Projection

$S^p =$

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

$T^p =$

	C	D	
t_1	1	'p'	0.6

Joins: assume independence

A	B	C	D	*prob*
'm'	1	1	'p'	0.8*0.6 = 0.48
'n'	1	1	'p'	0.5*0.6 = 0.30

Projection: union probability; assume independence

D	*prob*
'p'	(1 - (1 - 0.48)(1 - 0.3)) = 0.636

Umm.. This is wrong !!
Why ? The two tuples above are not independent.

Extensional: Example 2

- $\pi_D(S \bowtie_{B=C} T) \rightarrow$ Do Projection first, then Join

$S^p =$

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

$T^p =$

	C	D	
t_1	1	'p'	0.6

Projection: union probability; assume independence

B	*prob*
1	$(1 - (1 - 0.8)(1 - 0.5)) = 0.9$

Join: assume independence

B	C	D	*prob*
1	1	'p'	$0.9 * 0.6 = 0.54$

This is correct.
The correctness unfortunately depends on the plan used.
Called "safe plans" [Dalvi, Suciu 2004]

Extensional Evaluation: Safe Plans

- "Efficient Query Evaluation.."; Dalvi et al.
- Safe Plans:
  - Plans for which extensional evaluation returns correct probabilities
  - Carefully manage the operations (especially "project"s) to ensure safe-ness
    - Basically avoid emergence of correlated tuples

Extensional Evaluation: Safe Plans

- “Efficient Query Evaluation..”; Dalvi et al.
- Safe Plans:
  - Plans for which extensional evaluation returns correct probabilities
  - Carefully manage the operations (especially “project”s) to ensure safe-ness
    - Basically avoid emergence of correlated tuples
  - Only works if the base tuples are independent

Extensional Evaluation: Safe Plans

- "Efficient Query Evaluation.."; Dalvi et al.
- Safe Plans:
  - Plans for which extensional evaluation returns correct probabilities
  - Carefully manage the operations (especially "project"s) to ensure safe-ness
    - Basically avoid emergence of correlated tuples
  - Only works if the base tuples are independent
- Dichotomy (Dalvi et al., PODS 2007):
  - A query either has a safe plan, or is #P-Hard

Extensional Evaluation: Safe Plans

- “Efficient Query Evaluation..”; Dalvi et al.
- Safe Plans:
  - Plans for which extensional evaluation returns correct probabilities
  - Carefully manage the operations (especially “project”s) to ensure safe-ness
    - Basically avoid emergence of correlated tuples
  - Only works if the base tuples are independent
- Dichotomy (Dalvi et al., PODS 2007):
  - A query either has a safe plan, or is #P-Hard
  - Nothing about the actual instance though
    - A specific relation instance may still have a safe plan
  - This is a pessimistic, worst-case bound

Intensional

- $\pi_D(S \bowtie_{B=C} T) \rightarrow$ Do Join first, then Projection

$S^p =$

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

$T^p =$

	C	D	
t_1	1	'p'	0.6

Intensional

- $\pi_D(S \bowtie_{B=C} T) \rightarrow$ Do Join first, then Projection

$S^p =$

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

$T^p =$

	C	D	
t_1	1	'p'	0.6

Join (intersection):

A	B	C	D	E
'm'	1	1	'p'	$s_1 \wedge t_1$
'n'	1	1	'p'	$s_2 \wedge t_1$

Intensional

- $\pi_D(S \bowtie_{B=C} T) \rightarrow$ Do Join first, then Projection

$S^p =$

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

$T^p =$

	C	D	
t_1	1	'p'	0.6

Join (intersection):

A	B	C	D	E
'm'	1	1	'p'	$s_1 \wedge t_1$
'n'	1	1	'p'	$s_2 \wedge t_1$

Projection (union):

D	E
'p'	$(s_1 \wedge t_1) \vee (s_2 \wedge t_1)$

Intensional

- $\pi_D(S \bowtie_{B=C} T) \rightarrow$ Do Join first, then Projection

$S^p =$

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

$T^p =$

	C	D	
t_1	1	'p'	0.6

Join (intersection):

A	B	C	D	E
'm'	1	1	'p'	$s_1 \wedge t_1$
'n'	1	1	'p'	$s_2 \wedge t_1$

Projection (union):

D	E
'p'	$(s_1 \wedge t_1) \vee (s_2 \wedge t_1)$

- Evaluate the expression at the end (this can be #P-Hard)
- Can handle correlations (between tuples)

Query Evaluation

- Intensional vs Extensional
  - Extensional is fast, but not always correct/applicable
  - Intensional is too expensive

Query Evaluation

- Intensional vs Extensional
  - Extensional is fast, but not always correct/applicable
  - Intensional is too expensive
- Sen and Deshpande, ICDE 2007
  - Using Graphical Models to represent the correlations between tuples
  - "Closed"

Query Evaluation

- Intensional vs Extensional
  - Extensional is fast, but not always correct/applicable
  - Intensional is too expensive
- Sen and Deshpande, ICDE 2007
  - Using Graphical Models to represent the correlations between tuples
  - "Closed"
  - Always correct (which means still #P-Hard in general)
  - Can achieve the best of both worlds
    - If data doesn't have correlations, as fast as extensional

Query Evaluation

- Intensional vs Extensional
  - Extensional is fast, but not always correct/applicable
  - Intensional is too expensive
- Sen and Deshpande, ICDE 2007
  - Using Graphical Models to represent the correlations between tuples
  - "Closed"
  - Always correct (which means still #P-Hard in general)
  - Can achieve the best of both worlds
    - If data doesn't have correlations, as fast as extensional
- Still many unanswered questions
  - Can we do better if approximations are allowed ?
    - Probabilities are rarely required exactly
    - "Ranking" is all thats needed

Attribute-level Uncertainty

- In some cases easier, in some cases harder to deal with
- Continuous distributions common (e.g. in sensor applications)
- Hard to do exact querying over continuous attributes
- What about correlations ?
    - Temperatures are nearby locations are usually correlated
- Much work on this topic; some we are doing here

Lineage

- Given a tuple t, we would like to know:
  - When t was derived ?
  - How t was derived ?
  - What data was used to derive t ?
  - Who (which user) cause t to be derived ?
- Can keep the information along with the t
  - Propagate it as the new result tuples are generated
  - Clearly not feasible
  - Try storing with relations instead of each tuple
  - Annotate the tuples somehow
  - ...
- A lot of open questions, few answers

Discussion

- Very interesting “hot” area
- Lot of work by many groups
  - On defining models, efficient query processing, building systems etc. . .
  - We are doing a bunch of work here
- No real consensus yet emerging
- “Killer Application” ??
- Key is keeping it simple, otherwise no one will use it