

Probabilistic Databases

Adapted from: D. Suciu, N. Dalvi: Foundations of Probabilistic Answers to Queries. ACM SIGMOD Conf. 2005.
<http://www.cs.washington.edu/~suciu>

Extended list of references available from
<http://www.cs.washington.edu/~suciu>

1

Databases Today are Deterministic

- An item either is in the database or is not
- A tuple either is in the query answer or is not
- This applies to all state-of-the-art data models:
 - relational, OO, hierarchical, XML, ...

2

What is a Probabilistic Database ?

- “An item belongs to the database” is a probabilistic event
- “A tuple is an answer to a query” is a probabilistic event
- Can be extended to all data models; we discuss only probabilistic *relational* data

3

Two Types of Probabilistic Data

- Database is deterministic
Query answers are probabilistic
- Database is probabilistic
Query answers are probabilistic

4

Overview

- Part I: Applications: Managing Imprecisions
- Part II: Probabilistic Data Semantics
- Part III: Representation Formalisms
- Part IV: Algorithms, Implementation Techniques

5

Part I

Applications: Managing Imprecision

6

Outline

1. Ranking query answers
2. Record linkage
3. Quality in data integration
4. Inconsistent data
5. Information disclosure

7

1. Ranking Query Answers

Database is deterministic

The query returns a *ranked list of tuples*

- User interested in Top-N answers.

8

Ranking:

Compute a similarity score between a tuple and the query

```
Q = SELECT R.*, x
      FROM R
      ORDER BY score(abs(A1-v1), ..., abs(Am-vm)) as x
      STOP AFTER N;
```

Query is a vector:

$$Q = (v_1, \dots, v_m)$$

Tuple is a vector:

$$T = (u_1, \dots, u_m)$$

- Rank tuples according to some similarity function; e.g., their TF/IDF similarity to the query.
- Scoring function encapsulates all the magic!

9

[Motto:1988,Dalvi&S:2004]

Similarity Predicates in SQL

Beyond a single table:

“Find the good deals in a neighborhood !”

```
SELECT *
FROM Houses x
WHERE x.bedrooms ~ 4 AND x.style ~ 'craftsman' AND x.price ~ 600k
AND NOT EXISTS
(SELECT *
 FROM Houses y
 WHERE x.district = y.district AND x.ID != y.ID
 AND y.bedrooms ~ 4 AND y.style ~ 'craftsman' AND y.price ~ 600k)
```

Users specify similarity predicates with ~

System combines atomic similarities using probabilities

Types of Similarity Predicates

- String edit distances:
 - Levenstein distance, Q-gram distances
- TF/IDF scores
- Ontology distance / semantic similarity:
 - Wordnet
- Phonetic similarity:
 - SOUNDEX

[Theobald&Weikum:2002,
Hung,Deng&Subrahmanian:2004]

11

Summary on Ranking Query Answers

Types of imprecision addressed:

Data is precise, query answers are imprecise:

- User has limited understanding of the data
- User has limited understanding of the schema
- User has personal preferences

Probabilistic approach would...

- Principled semantics for complex queries
- Integrate well with other types of imprecision

12

2. Record Linkage

Determine if two data records describe the same object

Scenarios:

- Join/merge two relations;
e.g., bluewin users and SwissMobile users
- Remove duplicates from a single relation
- Validate incoming tuples against a reference

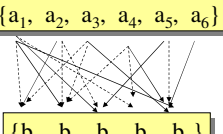
13

Fellegi-Sunter Model

A **probabilistic** model/framework

- Given two sets of records A, B:

Goal: partition $A \times B$ into:

- Match $A = \{a_1, a_2, a_3, a_4, a_5, a_6\}$
 - Uncertain
 - Non-match $B = \{b_1, b_2, b_3, b_4, b_5\}$
- 

[Cohen: Tutorial]

Non-Fellegi Sunter Approaches

Deterministic linkage

- Normalize records, then test equality
 - E.g. for addresses
 - Very fast when it works
- Hand-coded rules for an “acceptable match”
 - E.g. “same SSN”; or “same last name AND same birth date”
 - Difficult to tune

15

[Cohen:1998]

Application: Data Integration

WHIRL

- All attributes in all tables are of type *text*
- Datalog queries with two kinds of predicates:
 - Relational predicates
 - Similarity predicates $X \sim Y$

16

[Cohen:1998]

WHIRL Example

$Q_1(*) :- P(\text{Company}_1, \text{Industry}_1),$
 $Q(\text{Company}_2, \text{Website}),$
 $R(\text{Industry}_2, \text{Analysis}),$
 $\text{Company}_1 \sim \text{Company}_2,$
 $\text{Industry}_1 \sim \text{Industry}_2$

Score of an answer tuple = product of similarities

17

Summary on Record Linkage

Types of imprecision addressed:

Same entity represented in different ways

- Misspellings, lack of canonical representation, etc.

A probability model would...

- Allow system to use the match probabilities: cheaper, on-the-fly
- But need to model complex probabilistic correlations: is one set a reference set? how many duplicates are expected?

18

[Florescu,Koller,Levy97;Chang,GarciaMolina00;Mendelzon,Mihaila01]

3. Quality in Data Integration

Use of probabilistic information to reason about soundness, completeness, and overlap of sources

Applications (optimized access to data sources):

- Order access to information sources
- Compute confidence scores for the answers

19

[Mendelzon&Mihaila:2001]

Global Historical Climatology Network

- Integrates climatic data from:
 - 6000 temperature stations
 - 7500 precipitation stations
 - 2000 pressure stations

Soundness of a data source:
what fraction of items are correct

Completeness data source:
what fractions of items it actually contains

20

[Florescu,Koller,Levy:1997]

Goal 1: completeness → order source accesses

S₅ S₇₄ S₂ S₃₁ ...

[Mendelzon&Mihaila:2001]

Goal 2: soundness → query confidence

Q(y, v) :-
Temperature(s, y, m, v), Station(s, lat, lon, "US"),
y ≥ 1950, y ≤ 1955, lat ≥ 48, lat ≤ 49

Answer:

Year	Value	Confidence
1952	55° F	0.7
1954	-22° F	0.9
...

21

Summary: Quality in Data Integration

Types of imprecision addressed

Overlapping, inconsistent, incomplete data sources

- Data is probabilistic
- Query answers are probabilistic

They use already a probabilistic model

- Needed: complex probabilistic spaces. E.g. a tuple t in V₁ has 60% probability of also being in V₂
- Query processing still in its infancy

22

[Bertosi&Chomicki:2003]

4. Inconsistent Data

Goal:
consistent query answers
from *inconsistent* databases

Applications:

- Integration of autonomous data sources
- Un-enforced integrity constraints
- Temporary inconsistencies

23

[Bertosi&Chomicki:2003]

The Repair Semantics

Consider all "repairs"

Key (?!?)

Name	Affiliation	State	Area
Miklau	UW	WA	Data security
Dalvi	UW	WA	Prob. Data
Balazinska	UW	WA	Data streams
Balazinska	MIT	MA	Data streams
Miklau	Umass	MA	Data security

Find people in State=WA ⇒ Dalvi

Find people in State=MA ⇒ ∅

High precision, but low recall

24

Alternative Probabilistic Semantics

Name	Affiliation	State	Area	P
Miklau	UW	WA	Data security	0.5
Dalvi	UW	WA	Prob. Data	1
Balazinska	UW	WA	Data streams	0.5
Balazinska	MIT	MA	Data streams	0.5
Miklau	Umass	MA	Data security	0.5

State=WA \Rightarrow Dalvi, Balazinska(0.5), Miklau(0.5)

State=MA \Rightarrow Balazinska(0.5), Miklau(0.5)

Lower precision, but better recall

25

[Widom:2005]

Trio

- Witness 1: Car green; driver was a man, 20 years.
- Witness 2: Car yellow; driver was 35 years old.
- Statements are partly contradictory. Statements are partly independent.
- Need to gather facts and weigh options:
 - What is the probability that the driver was a man?
 - What is the probability that the car is green and the driver is 35 years old?
 - (Which witness do you trust; do you trust a witness all or nothing?)

26

Summary: Inconsistent Data

Types of imprecision addressed:

- Data from different sources is contradictory
- Data is uncertain, hence, arguably, probabilistic
- Query answers are probabilistic

A probabilistic would...

- Give better recall and precision!
- Needs to support disjoint tuple events

27

5. Information Disclosure

Goal

- Disclose some information (V) while protecting private or sensitive data S

Applications:

- Privacy preserving data mining \rightarrow V=anonymized transactions
- Data exchange \rightarrow V=standard view(s)
- K-anonymous data \rightarrow V=k-anonymous table

S = some atomic fact that is private

28

[Evfimievski,Gehrke,Srikant:03; Miklau&S:04;Miklau,Dalvi&S:05]

$\Pr(S)$ = a priori probability of S

$\Pr(S | V)$ = a posteriori probability of S

29

[Evfimievski,Gehrke,Srikant:03; Miklau&S:04;Miklau,Dalvi&S:05]

Information Disclosure

- If $\rho_1 < \rho_2$, a ρ_1, ρ_2 privacy breach:

$$\Pr(S) \leq \rho_1 \text{ and } \Pr(S | V) \geq \rho_2$$

- Perfect security:

$$\Pr(S) = \Pr(S | V)$$

- Practical security:

$$\lim_{\text{Database size remains fixed, domain size} \rightarrow \infty} \Pr(S | V) = 0$$

Summary: Information Disclosure

Is this a type of imprecision in data ?

- Yes: it's the adversary's uncertainty about the private data.
- The only type of imprecision that is good

Techniques

- Probabilistic methods: long history [Shannon'49]
- Definitely need conditional probabilities

31

Summary: Information Disclosure

Important fundamental duality:

- Query answering: want Probability $\lesssim 1$
- Information disclosure: want Probability $\gtrsim 0$

They share the same fundamental concepts and techniques

32

Summary: Information Disclosure

What is required from the probabilistic model

- Don't know the possible instances
- Express the adversary's knowledge:
 - Cardinalities: $\text{Size}(\text{Employee}) \simeq 1000$
 - Correlations between values: $\text{area-code} \rightsquigarrow \text{city}$
- Compute conditional probabilities

33

6. Other Applications

- Sensor data [Deshpande, Guestrin, Madden:2004]
- Personal information management
 - Semex [Dong&Halevy:2005, Dong, Halevy, Madhavan:2005]
 - Heystack [Karger et al. 2003], Magnet [Sinha&Karger:2005]
- Using statistics to answer queries [Dalvi&S:2005]

34

Summary on Part I: Applications

Common in these applications:

- Data in database and/or in query answer is uncertain, ranked; sometimes probabilistic

Need for common probabilistic model:

- Main benefit: uniform approach to imprecision
- Other benefits:
 - Handle complex queries (instead of single table TF/IDF)
 - Cheaper solutions (on-the-fly record linkage)
 - Better recall (constraint violations)

35

Part II

Probabilistic Data Semantics

36

Outline

- The possible worlds model
- Query semantics

37

Possible Worlds Semantics

Attribute domains:

$\text{int, char}(30), \text{varchar}(55), \text{datetime}$

values: $2^{32}, 2^{120}, 2^{440}, 2^{64}$

Relational schema:

$\text{Employee}(\text{name:varchar}(55), \text{dob:datetime}, \text{salary:int})$

of tuples: $2^{440} \times 2^{64} \times 2^{23}$

Database schema:

of instances: $2^{440} \times 2^{64} \times 2^{23}$

$\text{Employee}(\dots), \text{Projects}(\dots), \text{Groups}(\dots), \text{WorksFor}(\dots)$

of instances: N (= BIG but finite)

The Definition

The set of all possible database instances:

$\text{INST} = \{I_1, I_2, I_3, \dots, I_N\}$

will use Pr or \mathbb{P} interchangeably

Definition A probabilistic database \mathbb{P} is a probability distribution on INST

$\text{Pr} : \text{INST} \rightarrow [0,1]$ s.t. $\sum_{i=1,N} \text{Pr}(I_i) = 1$

Definition A possible world is I s.t. $\text{Pr}(I) > 0$

39

$\mathbb{P} =$ Example

Customer	Address	Product
John	Seattle	Gizmo
John	Seattle	Camera
Sue	Denver	Gizmo

$\text{Pr}(I_1) = 1/3$

Customer	Address	Product
John	Boston	Gadget
Sue	Denver	Gizmo

$\text{Pr}(I_2) = 1/12$

Customer	Address	Product
John	Seattle	Gizmo
John	Seattle	Camera
Sue	Seattle	Camera

$\text{Pr}(I_3) = 1/2$

Customer	Address	Product
John	Boston	Gadget
Sue	Seattle	Camera

$\text{Pr}(I_4) = 1/12$

Possible worlds = $\{I_1, I_2, I_3, I_4\}$

40

Tuples as Events

One tuple $t \Rightarrow$ event $t \in I$

$\text{Pr}(t) = \sum_{I: t \in I} \text{Pr}(I)$

Two tuples $t_1, t_2 \Rightarrow$ event $t_1 \in I \wedge t_2 \in I$

$\text{Pr}(t_1, t_2) = \sum_{I: t_1 \in I \wedge t_2 \in I} \text{Pr}(I)$

41

Tuple Correlation

Disjoint

$\text{Pr}(t_1, t_2) = 0$

--

Negatively correlated

$\text{Pr}(t_1, t_2) < \text{Pr}(t_1) \text{Pr}(t_2)$

-

Independent

$\text{Pr}(t_1, t_2) = \text{Pr}(t_1) \text{Pr}(t_2)$

0

Positively correlated

$\text{Pr}(t_1, t_2) > \text{Pr}(t_1) \text{Pr}(t_2)$

+

Identical

$\text{Pr}(t_1, t_2) = \text{Pr}(t_1) = \text{Pr}(t_2)$

++

42

$I^P =$ Example

Customer	Address	Product
John	Seattle	Gizmo
John	Seattle	Camera
Sue	Denver	Gizmo

$Pr(I_1) = 1/3$

Customer	Address	Product
John	Seattle	Gizmo
John	Seattle	Camera
Sue	Seattle	Camera

$Pr(I_3) = 1/2$

Customer	Address	Product
John	Boston	Gadget
Sue	Denver	Gizmo

$Pr(I_2) = 1/12$

Customer	Address	Product
John	Boston	Gadget
Sue	Seattle	Camera

$Pr(I_4) = 1/12$

43

Query Semantics

Given a query Q and a probabilistic database I^P , what is the meaning of $Q(I^P)$?

44

Query Semantics

Semantics 1: Possible Answers

A probability distributions on *sets of tuples*

$$\forall A. Pr(Q = A) = \sum_{I \in INST. Q(I)=A} Pr(I)$$

Semantics 2: Possible Tuples

A probability function on *tuples*

$$\forall t. Pr(t \in Q) = \sum_{I \in INST. t \in Q(I)} Pr(I)$$

45

Example: Query Semantics

Purchase^P

Name	City	Product
John	Seattle	Gizmo
John	Seattle	Camera
Sue	Denver	Gizmo
Sue	Denver	Camera

$Pr(I_1) = 1/3$

Name	City	Product
John	Boston	Gizmo
Sue	Denver	Gizmo
Sue	Seattle	Gadget

$Pr(I_2) = 1/12$

Name	City	Product
John	Seattle	Gizmo
John	Seattle	Camera
Sue	Seattle	Camera

$Pr(I_3) = 1/2$

Name	City	Product
John	Boston	Camera
Sue	Seattle	Camera

$Pr(I_4) = 1/12$

SELECT DISTINCT x.product
FROM Purchase^P x, Purchase^P y
WHERE x.name = 'John'
and x.product = y.product
and y.name = 'Sue'

Possible answers semantics:

Answer set	Probability
Gizmo, Camera	1/3
Gizmo	1/12
Camera	7/12

Possible tuples semantics:

Tuple	Probability
Camera	11/12
Gizmo	5/12

46

Special Case

Tuple independent probabilistic database

$INST = \mathcal{P}(TUP)$
 $N = 2^M$

$TUP = \{t_1, t_2, \dots, t_M\}$ = all tuples

$pr : TUP \rightarrow [0,1]$ No restrictions

$$Pr(I) = \prod_{t \in I} pr(t) \times \prod_{t \notin I} (1-pr(t))$$

47

Tuple Prob. \Rightarrow Possible Worlds

$J =$

Name	City	pr
John	Seattle	$p_1 = 0.8$
Sue	Boston	$p_2 = 0.6$
Fred	Boston	$p_3 = 0.9$

$E[size(I^P)] = 2.3$ tuples

$I^P =$

I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8
John	John	John	John	John	John	John	John
Seattle	Seattle	Seattle	Seattle	Seattle	Seattle	Seattle	Seattle
Sue	Sue	Sue	Sue	Sue	Sue	Sue	Sue
Bosto	Bosto	Bosto	Bosto	Bosto	Bosto	Bosto	Bosto
Fred	Fred	Fred	Fred	Fred	Fred	Fred	Fred
Bosto	Bosto	Bosto	Bosto	Bosto	Bosto	Bosto	Bosto

$\sum = 1$

48

Tuple Prob. \Rightarrow Query Evaluation

Name	City	pr
John	Seattle	p_1
Sue	Boston	p_2
Fred	Boston	p_3

Customer	Product	Date	pr
John	Gizmo	...	q_1
John	Gadget	...	q_2
John	Gadget	...	q_3
Sue	Camera	...	q_4
Sue	Gadget	...	q_5
Sue	Gadget	...	q_6
Fred	Gadget	...	q_7

```
SELECT DISTINCT x.city
FROM Person x, Purchase y
WHERE x.Name = y.Customer
and y.Product = 'Gadget'
```

Tuple	Probability
Seattle	$p_1(1-(1-q_2)(1-q_3))$
Boston	$1 - (1 - p_2(1-(1-q_5)(1-q_6))) \times (1 - p_3 q_7)$

50

Summary of Part II

Possible Worlds Semantics

- Very powerful model: *any* tuple correlations
- Needs separate representation formalism

Summary of Part II

Query semantics

- Very powerful: *every* SQL query has semantics
- Very intuitive: from standard semantics
- Two variations, both appear in the literature

51

Summary of Part II

Possible answers semantics

- Precise
- Can be used to compose queries
- Difficult user interface

Possible tuples semantics

- Less precise, but simple; sufficient for most apps
- Cannot be used to compose queries
- Simple user interface

52

Part III

Representation Formalisms

53

Representation Formalisms

Problem

Need a good representation formalism

- Will be interpreted as possible worlds
- Several formalisms exists, but no winner

Main open problem in probabilistic db

54

Evaluation of Formalisms

- What possible worlds can it represent ?
- What probability distributions on worlds ?
- Is it closed under query application ?

55

Outline

A complete formalism:

- Intensional Databases

Incomplete formalisms:

- Various expressibility/complexity tradeoffs

56

[Fuhr&Roellke:1997]

Intensional Database

Atomic event ids

e_1, e_2, e_3, \dots

Probabilities:

$p_1, p_2, p_3, \dots \in [0,1]$

Event expressions: \wedge, \vee, \neg

$e_3 \wedge (e_5 \vee \neg e_2)$

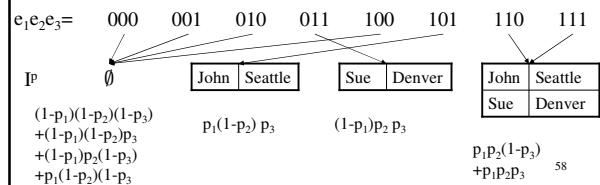
Intensional probabilistic database J:
each tuple t has an event attribute t.E

57

Intensional DB \Rightarrow Possible Worlds

J =

Name	Address	E
John	Seattle	$e_1 \wedge (e_2 \vee e_3)$
Sue	Denver	$(e_1 \wedge e_2) \vee (e_2 \wedge e_3)$



Possible Worlds \Rightarrow Intensional DB

Name	Address
John	Seattle
John	Boston
Sue	Seattle

$E_1 = e_1$ $\Pr(e_1) = p_1$
 $E_2 = \neg e_1 \wedge e_2$ $\Pr(e_2) = p_2/(1-p_1)$
 $E_3 = \neg e_1 \wedge \neg e_2 \wedge e_3$ $\Pr(e_3) = p_3/(1-p_1-p_2)$
 $E_4 = \neg e_1 \wedge \neg e_2 \wedge \neg e_3 \wedge e_4$ $\Pr(e_4) = p_4/(1-p_1-p_2-p_3)$

“Prefix code”

Name	Address
John	Seattle
Sue	Seattle

Name	Address
Sue	Seattle

Name	Address
John	Boston

\Rightarrow^P

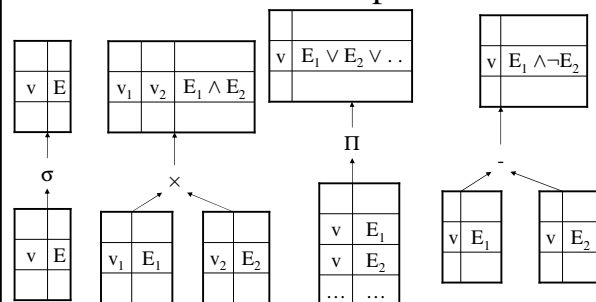
Name	Address	E
John	Seattle	$E_1 \vee E_2$
John	Boston	$E_1 \vee E_4$
Sue	Seattle	$E_1 \vee E_2 \vee E_3$

Intensional DBs are complete

59

[Fuhr&Roellke:1997]

Closure Under Operators



One still needs to compute probability of event expression

Summary on Intensional Databases

Event expression for each tuple

- Possible worlds: any subset
 - Probability distribution: any
- Complete (in some sense) ... but impractical

61

Restricted Formalisms

Explicit tuples

- Have a tuple template for every tuple that may appear in a possible world

Implicit tuples

- Specify tuples indirectly, e.g. by indicating how many there are

62

Explicit Tuples

Independent tuples

tuple = event

independent

Name	City	E	pr
John	Seattle	e_1	0.8
Sue	Boston	e_2	0.2
Fred	Boston	e_3	0.6

Atomic, distinct.
May use TIDs.

$E[\text{size}(\text{Customer})] = 1.6$ tuples

63

Application 1: Similarity Predicates

Name	City	Profession
John	Seattle	statistician
Sue	Boston	musician
Fred	Boston	physicist

Cust	Product	Category
John	Gizmo	dishware
John	Gadget	instrument
John	Gadget	instrument
Sue	Camera	musicware
Sue	Gadget	microphone
Sue	Gadget	instrument
Fred	Gadget	microphone

Step 1:
evaluate ~ predicates

```
SELECT DISTINCT x.city
FROM Person x, Purchase y
WHERE x.Name = y.Cust
and y.Product = 'Gadget'
and x.profession ~ 'scientist'
and y.category ~ 'music'
```

64

Application 1: Similarity Predicates

Name	City	Profession	pr
John	Seattle	statistician	$p_1=0.8$
Sue	Boston	musician	$p_2=0.2$
Fred	Boston	physicist	$p_3=0.9$

Step 1:
evaluate ~ predicates

```
SELECT DISTINCT x.city
FROM Personp x, Purchasep y
WHERE x.Name = y.Cust
and y.Product = 'Gadget'
and x.profession ~ 'scientist'
and y.category ~ 'music'
```

Step 2:
evaluate rest
of query

Cust	Product	Category	pr
John	Gizmo	dishware	$q_1=0.2$
John	Gadget	instrument	$q_2=0.6$
John	Gadget	instrument	$q_3=0.6$
Sue	Camera	musicware	$q_4=0.9$
Sue	Gadget	microphone	$q_5=0.7$
Sue	Gadget	instrument	$q_6=0.6$
Fred	Gadget	microphone	$q_7=0.7$

Tuple	Probability
Seattle	$p_1(1-(1-q_2)(1-q_3))$
Boston	$1-(1-p_2(1-(1-q_3)(1-q_6))) \times (1-p_3q_7)$

66

Explicit Tuples

Independent/disjoint tuples

Independent events: $e_1, e_2, \dots, e_i, \dots$

Split e_i into disjoint "shares" $e_i = e_{i1} \vee e_{i2} \vee e_{i3} \vee \dots$

$e_{34}, e_{37} \Rightarrow$ disjoint events

--

$e_{37}, e_{57} \Rightarrow$ independent events

0

Application 2: Inconsistent Data

Name	City	Product
John	Seattle	Gizmo
John	Seattle	Camera
John	Boston	Gadget
John	Huston	Gizmo
Sue	Denver	Gizmo
Sue	Seattle	Camera

```
SELECT DISTINCT Product
FROM Customer
WHERE City = 'Seattle'
```

Step 1:
resolve violations

Name → City (violated)

67

Application 2: Inconsistent Data

Name	City	Product	E	Pr
John	Seattle	Gizmo	e ₁₁	1/3
John	Seattle	Camera	e ₁₁	1/3
John	Boston	Gadget	e ₁₂	1/3
John	Huston	Gizmo	e ₁₃	1/3
Sue	Denver	Gizmo	e ₂₁	1/2
Sue	Seattle	Camera	e ₂₂	1/2



```
SELECT DISTINCT Product
FROM CustomerP
WHERE City = 'Seattle'
```

Step 2:
evaluate query

Step 1:
resolve violations

Tuple	Probability
Gizmo	$p_{11} = 1/3$
Camera	$1 - (1 - p_{11})(1 - p_{22}) = 2/3$

$E[\text{size}(\text{Customer})] = 2 \text{ tuples}$

68

[Barbara et al.92, Lakshmanan et al.97,Ross et al.05;Widom05]

Inaccurate Attribute Values

Name	Dept	Bonus	
John	Toy	Great	0.4
		Good	0.5
		Fair	0.1
Fred	Sales	Good	1.0

Name	Dept	Bonus	E	Pr
John	Toy	Great	e ₁₁	0.4
John	Toy	Good	e ₁₂	0.5
John	Toy	Fair	e ₁₃	0.1
Fred	Sales	Good	e ₂₁	1.0

Inaccurate attributes



Disjoint and/or independent events

69

Summary on Explicit Tuples

Independent or disjoint/independent tuples

- Possible worlds: subsets
- Probability distribution: restricted
- Closure: no

In KR:

- Bayesian networks: disjoint tuples
- Probabilistic relational models: correlated tuples

[Friedman,Getoor,Koller,Pfeffer: 1999] 70

Summary on Part III: Representation Formalism

- Intensional databases:
 - Complete (in some sense)
 - Impractical, but...
 - ...important practical restrictions
- Incomplete formalisms:
 - Explicit tuples
 - Implicit tuples
- We have not discussed query processing yet

71

Part IV

Algorithms, Implementation Techniques

72

Outline

- Probability of boolean expressions
- Query complexity
- Monte Carlo algorithms

73

Needed for query processing

Probability of Boolean Expressions

$$E = X_1X_3 \vee X_1X_4 \vee X_2X_5 \vee X_2X_6$$

Randomly make each variable **true** with the following probabilities

$$\Pr(X_1) = p_1, \Pr(X_2) = p_2, \dots, \Pr(X_6) = p_6$$

What is $\Pr(E)$???

Answer: re-group cleverly $E = X_1(X_3 \vee X_4) \vee X_2(X_5 \vee X_6)$

$$\Pr(E) = 1 - (1 - p_1(1 - (1 - p_3)(1 - p_4))) (1 - p_2(1 - (1 - p_5)(1 - p_6)))$$

74

Now let's try this:

$$E = X_1X_2 \vee X_1X_3 \vee X_2X_3$$

No clever grouping seems possible.
Brute force:

X_1	X_2	X_3	E	Pr
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$(1 - p_1)p_2p_3$
1	0	0	0	
1	0	1	1	$p_1(1 - p_2)p_3$
1	1	0	1	$p_1p_2(1 - p_3)$
1	1	1	1	$p_1p_2p_3$

$$\Pr(E) = (1 - p_1)p_2p_3 + p_1(1 - p_2)p_3 + p_1p_2(1 - p_3) + p_1p_2p_3$$

Seems inefficient in general...

75

[Valiant:1979]

Complexity of Boolean Expression Probability

Theorem [Valiant:1979]

For a boolean expression E, computing $\Pr(E)$ is #P-complete

NP = class of problems of the form "is there a witness ?" SAT

#P = class of problems of the form "how many witnesses ?" #SAT

The decision problem for 2CNF is in PTIME
The counting problem for 2CNF is #P-complete

76

Summary on Boolean Expression Probability

- #P-complete
- It's hard even in simple cases: 2DNF
- Can do Monte Carlo simulation (later)

77

Query Complexity

Data complexity of a query Q:

- Compute $Q(I^P)$, for probabilistic database I^P

Simplest scenario only:

- Possible tuples semantics for Q
- Independent tuples for I^P

78

[Fuhr&Roelke:1997,Dalvi&S:2004]

Extensional Query Evaluation

Relational ops compute probabilities

or: $p_1 + p_2 + \dots$

Data complexity: PTIME

79

[Dalvi&S:2004]

```
SELECT DISTINCT x.City
FROM Personp x, Purchasep y
WHERE x.Name = y.Cust
and y.Product = 'Gadget'
```

Wrong !

Correct

Depends on plan !!!

80

Summary on Query Complexity

Extensional query evaluation:

- Very popular
 - generalized to “strategies” [Lakshmanan et al.1997]
- However, result depends on query plan !

General query complexity

- #P complete (not surprising, given #SAT)
- Already #P hard for very simple query (Q_{bad})

Probabilistic database have high query complexity

Query Processing on a Probabilistic Database

82

[Karp,Luby&Madras:1989]

1. Monte Carlo Simulation

Naïve:

$$E = X_1 X_2 \vee X_1 X_3 \vee X_2 X_3$$

```

Cnt ← 0
repeat N times
  randomly choose  $X_1, X_2, X_3 \in \{0,1\}$ 
  if  $E(X_1, X_2, X_3) = 1$ 
    then Cnt = Cnt+1
P = Cnt/N
return P /*  $\approx \Pr(E)$  */

```

May be very big

0/1-estimator theorem

Works for any E
Not in PTIME

Theorem. If $N \geq (1/\Pr(E)) \times (4\ln(2/\delta)/\epsilon^2)$ then:
 $\Pr[|P/\Pr(E) - 1| > \epsilon] < \delta$

[Karp,Luby&Madras:1989]

Monte Carlo Simulation

Improved:

$$E = C_1 \vee C_2 \vee \dots \vee C_m$$

```

Cnt ← 0; S ← Pr(C1) + ... + Pr(Cm);
repeat N times
  randomly choose  $i \in \{1,2,\dots, m\}$ , with prob.  $\Pr(C_i) / S$ 
  randomly choose  $X_1, \dots, X_n \in \{0,1\}$  s.t.  $C_i = 1$ 
  if  $C_1=0$  and  $C_2=0$  and ... and  $C_{i-1} = 0$ 
    then Cnt = Cnt+1
P = Cnt/N * 1/
return P /*  $\approx \Pr(E)$  */

```

Now it's better

Theorem. If $N \geq (1/m) \times (4\ln(2/\delta)/\epsilon^2)$ then:
 $\Pr[|P/\Pr(E) - 1| > \epsilon] < \delta$

Only for E in DNF
In PTIME

Summary on Monte Carlo

Some form of simulation is needed in probabilistic databases, to cope with the #P-hardness bottleneck

- Naïve MC: works well when Prob is big
- Improved MC: needed when Prob is small

85

Conclusions

Probabilistic databases

- Possible worlds semantics
 - Simple
 - Every query has well defined semantics
- Need: expressive representation formalism
- Need: efficient query processing techniques

A great deal of research still required!!!

- Expressiveness vs. tractability of model (data model + query semantics)

86