Topics in Probabilistic and Statistical Databases

Dan Suciu
University of Washington

Lecture #1: Overview

This Course

- Advanced course on a special research topic
- I will give all lectures
- There will be recommended readings
- Your immediate goal: think, ask, discuss
- Your longer term goal: find great new research topics
- Credits: attendance + discussions

Communication

- http://www.cs.washington.edu/education/courses/cse599t/CurrentQtr/

- Need a volunteer TA for:
  - Setting up the mailing list
  - Regular updates to the Website

Prerequisites

I will assume the following:

- Some background in databases
- Some background in probability theory

But I will review, when details are needed

What you will see:

- Some out of order presentation

Motivation

- CS is increasingly dominated by data
- The new data has two trends:
  - Too large to process in traditional way
    - Data from the Web, physical world, science
  - Too imprecise to model in traditional way
    - Data extraction, measurement errors

Motivation

- Data management needs to produce the techniques to manage large, imprecise data
- Has been doing this for a while:
  - Data statistics, data sketches
  - Ranking query results
  - Approximate query answering
  - Data anonymization
  - *Probabilistic databases*

My 1^{st} Goal for This Course

- Comprehensive treatment of the technical material in probabilistic databases; resource for teaching probdb
  - Should be able to achieve goal 1 better than 2 and 3
- What you will see:
  - Technical detail on the slides
  - But: difficult to prepare about 80 slides / course, may have to use whiteboard extensively

My 2nd Goal for This Course

- Position probabilistic databases as a common foundation for a heterogeneous collection of techniques
  - Warning: I probably wont achieve this goal…

- What you will see:
  - I will discuss some related topics in an attempt to show how they fit under the probdb umbrella

My 3rd Goal for This Course

- Place probabilistic databases in the right context
  - Intellectual roots: probability theory and statistics, finite model theory, random graphs
  - Many neighboring areas in databases

- What you will see:
  - Digressions into other topics

Course Outline

1. Overview
2. Representation of Probabilistic Databases

3-4-5. Query Evaluation, Ranking

6. Query evaluation in Random Graphs
7. Probabilistic logic, Conditional logic

8-9. Approximate query processing

10. Review, discussions

Today’s Lecture

- Definition of a probabilistic database
- Three classes of applications

Probabilistic Databases

Notations:

- **R** = a relational schema
- D = a finite domain
- **Inst** = the (finite) set of all **R**-instances on D

Background: Relational Data

- Relational schema **R** = set of relation names with attributes

Likes(Drinker, Beer), Frequents(Drinker, Bar), Serves(Bar, Beer)

- Relational instance I over schema **R** = set of relations

Drinker	Beer

Drinker	Bar

Bar	Beer

The Definition

The set of all possible database instances:

$$\mathbf{Inst} = \{I_1, I_2, I_3, \ldots, I_N\}$$

Sample space (Ω)

Definition A *probabilistic database* PDB = (**Inst**, Pr) is a discrete probability distribution:

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

Definition A *possible world* is I s.t. Pr(I) > 0

A *possible tuple* is a tuple $t \in I$, for a possible world I

PDB =

Example

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

$Pr(I_1) = 1/3$

Customer	Address	Product
John	Boston	Gadget
Sue	Denver	Gizmo

$Pr(I_2) = 1/12$

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

$Pr(I_3) = 1/2$

Customer	Address	Product
John	Boston	Gadget
Sue	Seattle	Camera

$Pr(I_4) = 1/12$

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

Tuples as Events

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

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

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

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

Tuple Correlation

Disjoint	$Pr(t_1\ t_2) = 0$	--
Negatively correlated	$Pr(t_1\ t_2) < Pr(t_1)\ Pr(t_2)$	-
Independent	$Pr(t_1\ t_2) = Pr(t_1)\ Pr(t_2)$	0
Positively correlated	$Pr(t_1\ t_2) > Pr(t_1)\ Pr(t_2)$	+
Identical	$Pr(t_1\ t_2) = Pr(t_1) = Pr(t_2)$	++

PDB = Example

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

$Pr(I_1) = 1/3$

Customer	Address	Product
John	Boston	Gadget
Sue	Denver	Gizmo

$Pr(I_2) = 1/12$

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

$Pr(I_3) = 1/2$

Customer	Address	Product
John	Boston	Gadget
Sue	Seattle	Camera

$Pr(I_4) = 1/12$

Example: Disjoint-Independent Databases

<u>Definition</u> A **PDB** is disjoint-independent if for any set T of possible tuples one of the following holds:

- T is an independent set, or
- T contains two disjoint tuples

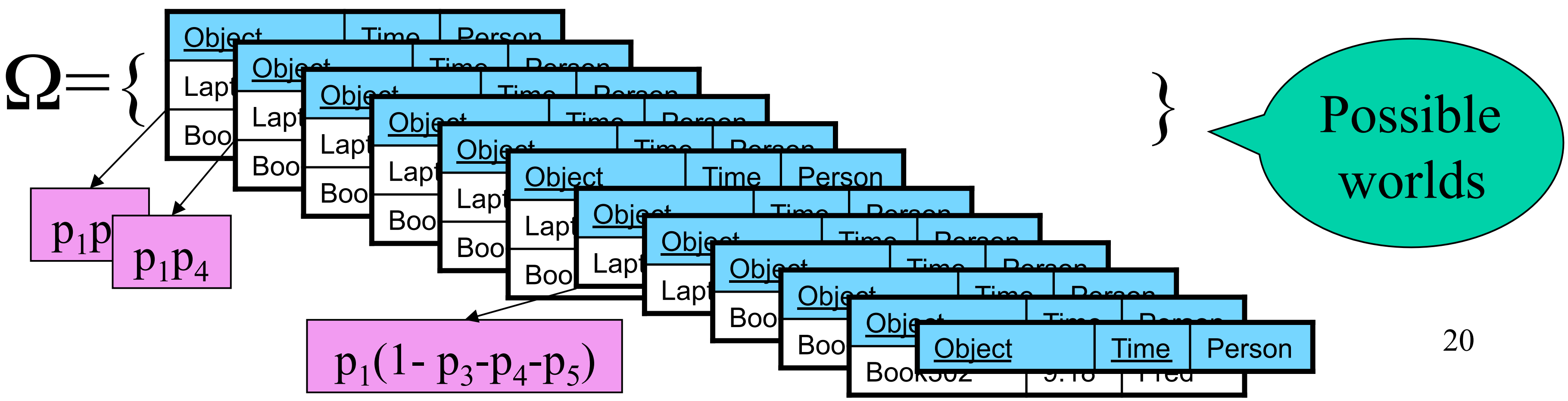
A disjoint-independent database can be fully specified by:

- all marginal tuple probabilities
- an indication of which tuples are disjoint or independent

Example: Disjoint-Independent Databases

Object	Time	Person	P
LaptopX77	9:07	John	p_1
		Jim	p_2
Book302	9:18	Mary	p_3
		John	p_4
		Fred	p_5

PDB



Background: Queries

- Relational queries = formulas in FO
- Conjunctive query:

  $\exists y_1 \exists y_2 \ldots \exists y_k.(g_1 \wedge g_2 \wedge \ldots \wedge g_m)$
- Conjunctive query notation:

  $q(x_1,\ldots,x_n) \text{ :- } g_1, g_2, \ldots, g_m$
- Boolean query = closed formula
- Boolean conjunctive query:

  $q \text{ :- } g_1, g_2, \ldots, g_m$

Examples

Likes(Drinker, Beer), Frequents(Drinker, Bar), Serves(Bar, Beer)

What are these queries and what do they return ?

$\forall y.\ (\text{Frequents}(x,y) \rightarrow \forall z.\ (\text{Likes}(x,z) \rightarrow \text{not}\ (\text{Serves}(y,z))))$

$\exists y.\ (\text{Frequents}(\text{Fred},x)\ \text{and}\ \text{Likes}(\text{Fred},y)\ \text{and}\ \text{Serves}(x,y))$

q(x) :- Frequents(Fred,x), Likes(Fred,y), Serves(x,y))

q :- Frequents(Fred,x), Likes(Fred,y), Serves(x,y))

Query Semantics

Given a query Q and a probabilistic database PDB, what is the meaning of Q(PDB) ?

Query Semantics

Semantics 1: Possible Sets of Answers

A probability distributions on <u>*sets of tuples*</u>

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

Semantics 2: Possible Tuples

A probability function on <u>*tuples*</u>

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

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 Purchasep x, Purchasep y
WHERE x.name = 'John'
        and x.product = y.product
        and y.name = 'Sue'
```

Possible answers semantics:

Answer set	Probability	
Gizmo, Camera	1/3	$Pr(I_1)$
Gizmo	1/12	$Pr(I_2)$
Camera	7/12	$P(I_3) + P(I_4)$

Possible tuples semantics:

Tuple	Probability	
Camera	11/12	$Pr(I_1)+P(I_3) + P(I_4)$
Gizmo	5/12	$Pr(I_1)+Pr(I_2)$

Query Semantics

- If q is a boolean query, then the possible answers and the possible tuples are essentially the same

Why ?

Factoid

- In traditional database theory, it suffices to study only boolean queries

Why ?

- But over probabilistic databases that reduction no longer works

Why ?

- We study first boolean query evaluation (=simpler) and discuss top-k semantics later

Special Case

Tuple independent probabilistic database

$\boxed{\mathbf{Tup} = \{t_1, t_2, \ldots, t_M\}}$ = all possible tuples

$\boxed{pr : \mathbf{Tup} \to (0,1]}$

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

Tuple Prob. ⇒ 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	$1-(1-p_2(1-(1-q_5)(1-q_6)))$ $\times(1-p_3 q_7)$

Three Classes of Application

- Uncertain data
- Information Disclosure
- Approximate query answering

1. Uncertain Data

We'll discuss three, many more exists

- Ranking query answers
- Information extraction
- Fuzzy joins

Most work on probdb has focused on this class of apps

Questions to Ponder

- Is there a ground truth (max likelihood world) ?
- What do we gain by keeping multiple worlds ?
- Are the confidence scores indeed probabilities ?

[Agrawal,Chaudhuri,Das,Gionis 2003]

Ranking Query Answers

The *Empty Answers* problem:

```
SELECT *
FROM Houses
WHERE bedrooms = 4
    AND  style = ‘craftsman’
    AND  district = ‘View Ridge’
    AND  price < 400000
```

No Matches !

[Agrawal,Chaudhuri,Das,Gionis 2003]

Ranking:

Compute a similarity score between a tuple and the query

```
Q = SELECT *
    FROM   R
    WHERE A1=v1 AND … AND Am=vm
```

Query is a vector: $Q = (v_1, \ldots, v_m)$

Tuple is a vector: $T = (u_1, \ldots, u_m)$

Rank tuples by their TF/IDF similarity to the query Q

Includes partial matches

Ranking Query Answers

```
SELECT *
FROM Houses
WHERE bedrooms = 4
    AND  style = 'craftsman'
    AND  district = 'View Ridge'
    AND  price < 400000
```

Are similarities probabilities ?

Address	Bedrooms	Style	District	Price	SimScore
…	5	Craftsman	Ravenna	300000	0.8
…	2	Craftsman	View ridge	500000	0.4
…	4	Ranch	U District	400000	0.7
…					

[Dalvi&S:2004]

Adding Similarity Predicates to 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 <u>probabilities</u>

[Dalvi&S:2004]

Evaluation using a ProbDB

```
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
```

A	B	S	D	P	x.Sim

A	B	S	D	P	y.Sim

[Dalvi&S:2004]

Evaluation using a ProbDB

Finally, evaluate the “rest of the query” (w/o ~) on the ProbDB

```
SELECT *
FROM Houses1 x
WHERE NOT EXISTS
      (SELECT *
       FROM Houses2 y
       WHERE x.district = y.district AND x.ID != y.ID)
```

A	B	S	D	P	x.Sim

A	B	S	D	P	y.Sim

Answer these for query ranking (in class)

- Is there a ground truth (max likelihood world) ?
- What do we gain by keeping multiple worlds ?
- Are the confidence scores indeed probabilities ?

[Gupta and Sarawagi, VLDB'2006]

ProbDB for IE Models

- Input:
  - Text = a collection of independent text records
  - Record = sequence of tokens x1,…,xn
- Set of labels $A = \{A_1, \ldots, A_K\}$
- Ouput:
  - Segmentation = $s_1, \ldots, s_p$
    where $s_i = (start_i, end_i, label_i)$

[Gupta and Sarawagi, VLDB'2006]

ProbDB for IE Models

Conditional Random Fields (CRF):

- = probability space on all segementations

Text record

52-A Goregaon West Mumbai 400 076

A possible segmentation (prob = 0.5)

[Gupta and Sarawagi, VLDB'2006]

ProbDB for IE Models

Conditional Random Fields (CRF):

- = probability space on all segementations

Text record

52-A Goregaon West Mumbai 400 076

All segmentations

House_no	Area	City	Pincode	Prob
52	Goregaon West	Mumbai	400 076	0.1
52-A	Goregaon	West Mumbai	400 076	0.2
52-A	Goregaon West	Mumbai	400 076	0.5
52	Goregaon	West Mumbai	400 076	0.2

[Gupta and Sarawagi, VLDB'2006]

ProbDB for IE Models

$$\Pr(s \mid x, \Lambda) = 1/Z(x) \exp(\Lambda \cdot \textstyle\sum_j f(j,x,s))$$

Where:

$\Lambda = (\lambda_i,\ldots,\lambda_N) =$ feature weights

$f = (f_i,\ldots,f_N) =$ feature function

$j = 1, \ldots, |s| =$ segment index

$Z =$ normalization factor

A feature $f_i(j,x,s)$ depends only on s_{j-1}, s_j and the corresponding x

$f_8(j,\ x,\ (2,5,y_{j-1}),(6,12,y_j)) =$
$[y_{j-1}= \text{House\_no}]\cdot[y_j=\text{Area}]\cdot[x_6x_7\ldots x_{12}$ appears in a list of areas$]$

[Gupta and Sarawagi, VLDB'2006]

ProbDB for IE Models

Traditional IE keeps the ***maximum likelihood segmentation***, which can be computed using dynamic programming (Viterbi):

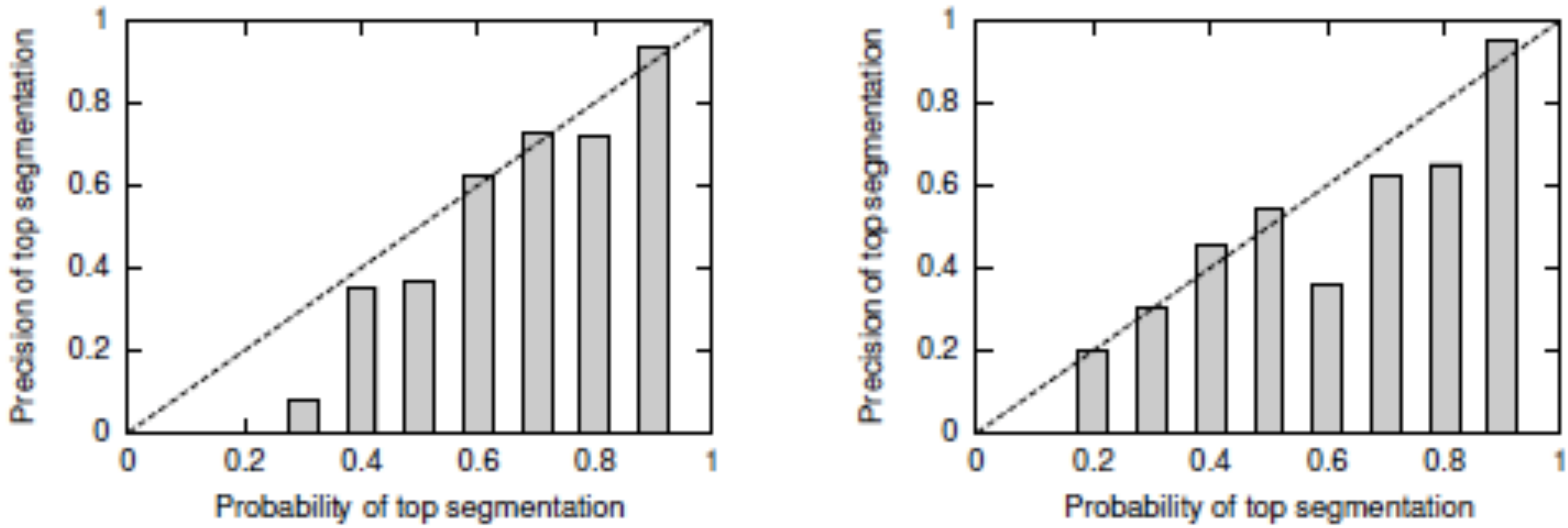
$$\boxed{\mathrm{argmax}_s \ \Pr(s \mid x, \Lambda)}$$

But this results in low recall, e.g. for the query:

```
SELECT DISTINCT x.name
FROM  Person x,   Address^p y
WHERE x.ID = y.ID and y.city = 'West Mumbai'
```

[Gupta and Sarawagi, VLDB'2006]

ProbDB for IE Models



(a) Cora

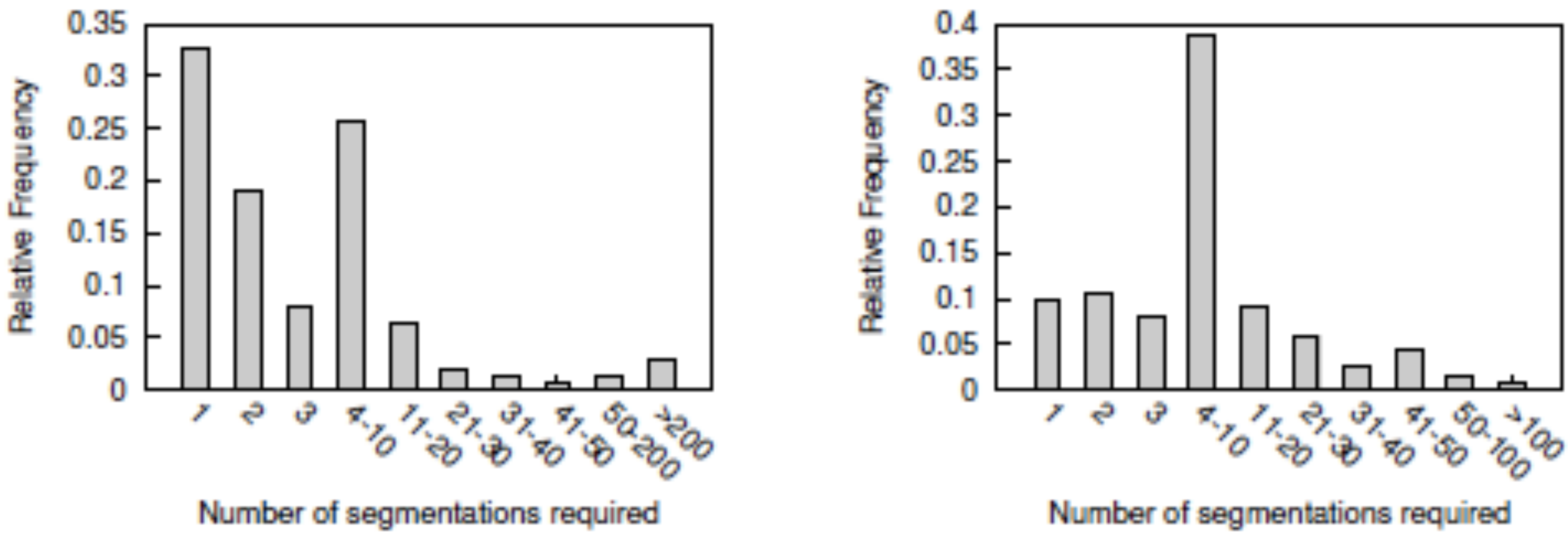
(b) Address

[Gupta and Sarawagi, VLDB’2006]

ProbDB for IE Models

Next idea: try to keep the top 2 segmentations (or top k…)

But k needs to be large to cover significant probability mass:



(a) Cora, p=0.95

(b) Address, p=0.9

[Gupta and Sarawagi, VLDB'2006]

ProbDB for IE Models

Keep all ! ➔ a probabilistic database:

ID	House-No	Street	City	P
1	52	Goregaon West	Mumbai	0.1
1	52-A	Goregaon West	Mumbai	0.4
1	52	Goregaon	West Mumbai	0.2
1	52-A	Goregaon	West Mumbai	0.2
2	. . . .	. . . .	. . . .	. . . .
2	. . . .			

```
SELECT DISTINCT x.name
FROM  Person x,   Address^p y
WHERE x.ID = y.ID and y.city = 'West Mumbai'
```

[Gupta and Sarawagi, VLDB'2006]

ProbDB for IE Models

The rest of their paper:

- Considers a compact representation of the probabilistic database

We will discuss representations next time

Answer these for IE (in class)

- Is there a ground truth (max likelihood world) ?
- What do we gain by keeping multiple worlds ?
- Are the confidence scores indeed probabilities ?

[Arasu, Ganti, Kaushik, VLDB'2006]

Similarity Joins

- Same object represented in different ways
- Why ?
  - Typos: "Woshington" v.s. "Washington"
  - Different naming conventions: "IBM" v.s. "International Business Machines Corporation"

[Arasu, Ganti, Kaushik, VLDB'2006]

Example

Company1

CName1	. . . Other attributes
Microsoft Corp	
Apple Computer	
Apples, Pears, and More	
. . .	

Company1

CName2	. . . Other attributes
Microsoft Inc	
Apple Corporation	
Apples and Pears Farm	
. . .	

[Arasu, Ganti, Kaushik, VLDB'2006]

Similarity Join

```
SELECT *
FROM Company1, Company2
WHERE cname1 ≈ cname2
```

We want the strings to be "similar"

[Arasu, Ganti, Kaushik, VLDB'2006]

What is "Similar" ?

- Similarity function sim(s1,s2):
  - Sim(s1,s2) > c means s1, s2 are similar

- Distance function dist(s1,s2):
  - Dist(s1,s2) < c means s1, s2 are similar

```
SELECT *
FROM Company1, Company2
WHERE cname1 ≈ cname2
```

→

```
SELECT *
FROM Company1, Company2
WHERE Sim(cname1, cname2) > c
```

[Arasu, Ganti, Kaushik, VLDB'2006]

Q-Grams

- Given a string s, a q-gram is a substring of length q
- Usually q = 3

washington woshington

s1 = {was, ash, shi, hin, ing, ngt, gto, ton}

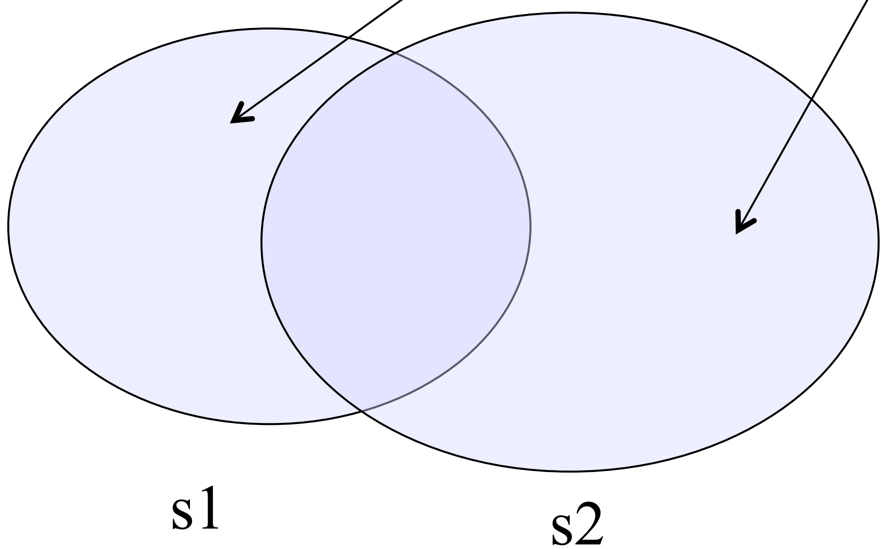
s2 = {wos, osh, shi, hin, ing, ngt, gto, ton}

Variation: may include beginning and end: ##w, #wa, on$, n$$

[Arasu, Ganti, Kaushik, VLDB'2006]

Hamming Distance

- $H(s1, s2) = |s1 \Delta s2| = |s1 - s2| + |s2 - s1|$

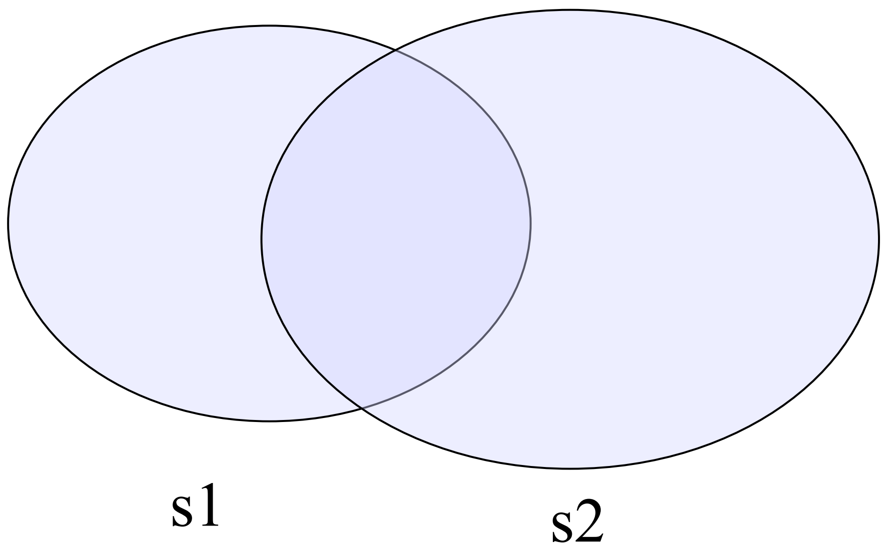


"s1 is similar to s2" iff $H(s1, s2) < k$

[Arasu, Ganti, Kaushik, VLDB'2006]

Jaccard Similarity

- $J(s1,s2) = |s1 \cap s2| / |s1 \cup s2|$



"s1 is similar to s2" iff $J(s1, s2) > c$

[Arasu, Ganti, Kaushik, VLDB'2006]

They are related !

- Suppose $|s1| = |s2| = L$
- Denote $I(s1,s2) = |s1 \cap s2|$

Then:

- $J(s1,s2) > c$ iff $I(s1, s2) > 2cL/(1+c)$

- $H(s1,s2) < k$ iff $I(s1,s2) > 2L - k$

Why ?

[Arasu, Ganti, Kaushik, VLDB'2006]

Representing q-Grams

Company (id, name, …)

Id	Name	. . .
1	Washington	. . .
2	. . .	. . .

CQ(id, qgram)

Id	Qgram
1	was
1	ash
1	shi
1	hin
. . .	. . .
2	. . .

```
SELECT x.id, y.id
FROM Company x, Company y
WHERE J(x.name, y.name) > c
```

```
SELECT x.id, y.id
FROM Company x, Company y
WHERE H(x.name, y.name) < k
```

```
SELECT x.id, y.id
FROM CQ x, CQ y
WHERE x.Qgram=y.Qgram
GROUP BY x.id, y.id
HAVING count(*) < k
```

[Arasu, Ganti, Kaushik, VLDB'2006]

ProbDBs for SS-Joins

- Most of the work is focused on computing it *efficiently* – we won't discuss here
- Main point: the semantics of the ss-join depends on the threshold c
- The threshold is hard to choose !

Better: keep several ss-joins and their similarities → probabilistic database

Answer these for SS-join (in class)

- Is there a ground truth (max likelihood world) ?
- What do we gain by keeping multiple worlds ?
- Are the confidence scores indeed probabilities ?

Other Types of Uncertain Data

- Deduplication
- Data integration
- Sensor readings
- RFID data
- Scientific data management
- Social networks

For most, the probabilistic data has not been studied seriously

Summary of Uncertain Data

- Data is almost like a traditional database, but certain values or tuples are uncertain
- We assume uncertainties are quantified as probabilities (but this may be difficult in some cases)
- The research questions are:
  - Representation (Lecture 2)
  - Query processing (Lectures 3,4,5)

[Rastogi, S., Hong, VLDB'2007]

2. Information Disclosure

Have private data

- Want to make available for *statistical analysis*
- But want to hide the private details

Two conflicting goals:

- Strong privacy
- Good utiliy

[Rastogi, S., Hong, VLDB'2007]

Example

Instance I

Name	Age	Nationality	Score
Fred	17	British	62
Alice	18	Czech	95
Mary	22	Indian	99
Joe	21	British	42
Bob	22	Czech	92

Statistical queries OK:

- average score of British students ?
- how many Czechs scored better than Indians ?

Private queries not OK

- what is Alice's score ?

[Rastogi, S., Hong, VLDB'2007]

K-Anonymization

Instance I

Name	Age	Nationality	Score
Fred	17	British	62
Alice	18	Czech	95
Mary	22	Indian	99
Joe	21	British	42
Bob	22	Czech	92

View V

Age	Nationality	Score
15-19	*	62
15-19	*	95
20-24	*	99
20-24	*	42
20-24	*	92

Mallory the Eavesdropper…

- Wants to find out Alice’s score…

Mallory's Cabal

View V

Age	Score
15-19	62
15-19	95
20-24	99
20-24	42
20-24	92

Case 1: Mallory Jr. knows that Alice is 18 years old. What is Alice's score ?

Case 2: Mallory Sr. knows that Alice is 18 years old, and is smarter than her brother, Bob, who is 22. What is Alice's score ?

Age	Nationality	Score
25	British	99
27	British	97
21	Indian	82
32	Indian	90
33	American	94
36	American	94

(a) Test Scores

Age	Nationality	Score
21-30	*	99
21-30	*	97
21-30	*	82
31-40	*	90
31-40	*	94
31-40	*	94

(b) 2-diversity and 3-anonymity [13]

There are better anonymizations…

Age	Nationality	Score
25	**British**	**99**
28	Indian	99
29	American	81
32	**Indian**	**90**
39	American	84
32	Indian	89

(c) FRAPP [3]

Age	Nationality	Score
25	**British**	**99**
21	British	99
22	Indian	89
32	**Indian**	**90**
28	Indian	99
29	American	81
33	**American**	**94**
27	American	94
32	British	83
36	**American**	**94**
26	American	99
39	Indian	94

(d) $\alpha\beta$ algorithm

[Rastogi, S., Hong, VLDB'2007]

[Rastogi, S., Hong, VLDB'2007]

Definition of Privacy

The adversary has a *prior*. This is ***a probabilistic database P***:

P(s) = the adversary's prior belief of the secret s

After seeing V, the adversary adjusts its belief to the *a posterior*

P(s | V) = the adversary's a posterior belief of the secret s

$$P(I \mid V) = 0 \qquad \text{if } V(I) \neq V$$
$$P(I \mid V) = P(I) / \sum_{J:V(J)=V} P(J) \qquad \text{if } V(I) = V$$

Definition The algorithm computing I→V is private for s if $P(s) \approx P(s \mid V)$

What are Prior and a Posteriori ?

View V

Age	Score
15-19	62
15-19	95
20-24	99
20-24	42
20-24	92

Case 1: Mallory Jr. knows
that Alice is 18 years old.
What is Alice's score ?

Case 2: Mallory Sr. knows
that Alice is 18 years old,
and is smarter than her brother,
Bob, who is 22.
What is Alice's score ?

Discussion of Information Disclosure

- The technical problem is computing $P(s \mid V)$
- But the probabilistic database P is very different from those in uncertain data !
  - We don't have the set of possible tuples t
  - We don't have their marginal probabilities $P(t)$
  - In fact, we may not even know Malory's prior !
  - Reasonable assumption: $P(t)$ is "small", $\forall t$

Discussion of Information Disclosure

- Approach 1: quantify over ALL P's
  - "perfect security"
- Approach 2: fix one particular "small" P
  - Random graphs
- Approach 3: quantify over all "small" P's
  - Conditional logic ?? Differential privacy ??
  - This approach is an open research question
- Lectures 6 and 7

3. Approximate Query Answering

- Have data I, want to compute a query q(I)
  - Usually q is a count(*), or avg( )

- But I is too big: will use some statistics S to estimate q

  - Goal: rewrite q to q0 s.t. q(I) $\approx$ q0(S)

Two Examples

- Consistent cardinality estimation
- Robust query optimization

[Markl et al. VLDB'2005]

Example

```
SELECT count(*)
FROM R
WHERE R.A=10 and R.B=20 and R.C=30
```

Think of this query as being issued during query optimization:
Optimizer wants to find out the size of a subplan

Assume |R| = 1,000,000,000
Can't scan R. Will use statistics instead

[Markl et al. VLDB'2005]

Histograms to the Rescue !

R.A =	…	9	10	11	…
count =	…	…	100,000,000	…	…

R.B =	…	19	20	21	…
count =	…	…	200,000,000	…	…

R.C =	…	29	30	31	…
count =	…	…	250,000,000	…	…

[Markl et al. VLDB'2005]

Normalized Histograms

Replace counts with frequencies, by dividing by |R|=1,000,000,000:

R.A =	…	10	…
s_1 =	…	0.1	…

R.B =	…	20	…
s_2 =	…	0.2	…

R.C =	…	30	…
s_3 =	…	0.25	…

```
SELECT count(*)
FROM R
WHERE R.A=10 and R.B=20 and R.C=30
```

What's your estimate ?

[Markl et al. VLDB'2005]

2-Dimensional Histograms

We have two more histograms for the same |R|=1,000,000,000:

R.A =	…	10	…
s_1 =	…	0.1	…

R.B =	…	20	…
s_2 =	…	0.2	…

R.C =	…	30	…
s_3 =	…	0.25	…

R.AB	…	10,20	…
s_{12} =	…	0.05	…

R.AC	…	20,30	…
s_{13} =	…	0.03	…

```
SELECT count(*)
FROM R
WHERE R.A=10 and R.B=20 and R.C=30
```

What's your estimate now ?

[Markl et al. VLDB'2005]

The Estimation Problem

- We have several statistics
  - Here: five histograms
- Want to estimate a query q

- Problem:
  - There are different ways to use the histograms, and result in inconsistent answers
  - We want a consistent estimate

[Markl et al. VLDB'2005]

Histograms as Probabilities

- Probability space on $\{(x,y,z) \mid (x,y,z) \in \{0,1\}^3\}$ defined as follows:

- Randomly select a tuple t from R
  - If t.A=10 then set x=1; otherwise x=0
  - If t.B=20 then set y=1; otherwise y=0
  - If t.C=30 then set z=1; otherwise z=0

Is this a probabilistic database ??

[Markl et al. VLDB'2005]

Modeling Histograms as ProbDB

- There are eight possible worlds, need their probs
- The five histograms lead to 5+1 = 6 constraints:

x	y	z	P
0	0	0	p_{000}
0	0	1	p_{001}
0	1	0	p_{010}
0	1	1	p_{011}
1	0	0	p_{100}
1	0	1	p_{101}
1	1	0	p_{110}
1	1	1	p_{111}

$p_{000}+p_{001}+p_{010}+p_{011}+p_{100}+p_{101}+p_{110}+p_{111}= 1$
$p_{100}+p_{101}+p_{110}+p_{111}=s_1$
$p_{010}+p_{011}+p_{110}+p_{111}=s_2$
$p_{001}+p_{011}+p_{101}+p_{111}=s_3$
$p_{110}+p_{111}=s_{12}$
$p_{101}+p_{111}=s_{13}$

But underdetermined.
How do we choose ?

[Markl et al. VLDB'2005]

Maximum Entropy Principle

- Equivalent to the principle of indifference
- The entropy: $H = - \sum p_i \log(p_i)$
- There is a unique solution to the previous system that maximizes H, which is obtained by solving a non-linear system of equations
  - IN CLASS
- It turns out: $p_{111} = 0.015$, hence $q(I) \approx 15{,}000{,}000$

A Much Simpler Approach: Sampling

- R has N=1,000,000,000 tuples
- Compute (offline) a sample of size n =500

```
SELECT count(*)
FROM R
WHERE R.A=10 and R.B=20 and R.C=30
```

- Evaluate the query on the sample ➔ 8 tuples

What is your estimate ?

[Babock et al. SIGMOD'2005]

Robust Query Optimization

Traditional optimization:

- Plan 1: use index
- Plan 2: sequential scan

- The choice between 1 and 2 depends on the estimated selectivity
- E.g. for $p < 0.26$ the Plan 1 is better

[Babock et al. SIGMOD'2005]

Robust Query Optimization

The performance/predictability tradeoff:

- Plan 1: use index
  - If it is right ➔ ☺
  - If it is wrong ➔ ☹ MUST AVOID THIS !!
- Plan 2: sequential scan ➔ 😐

Optimizing performance may result in significant penalty, with some probabililty

[Babock et al. SIGMOD’2005]

Query Plan Cost

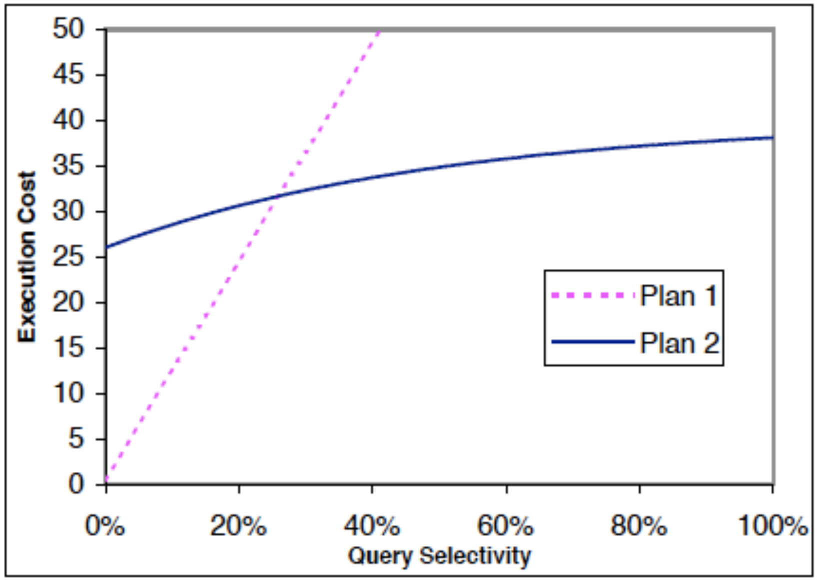


Figure 1: Execution Costs for Two Hypothetical Plans

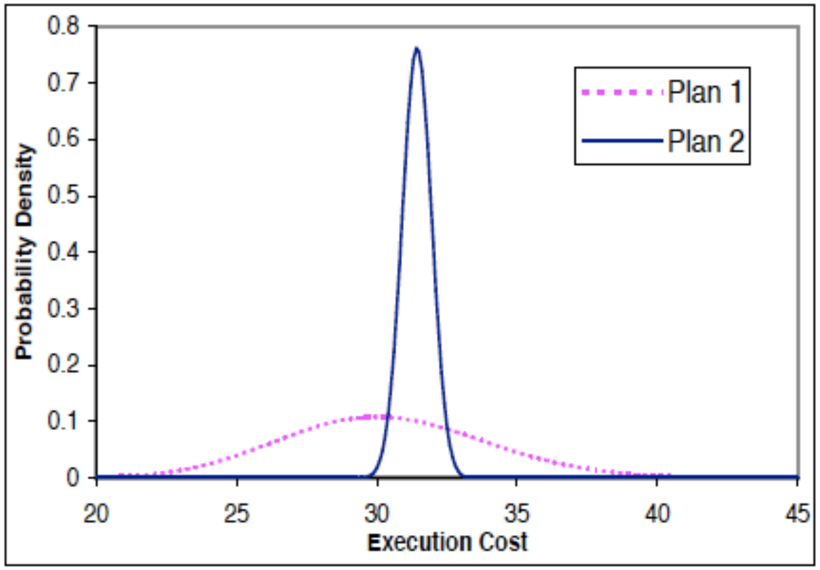
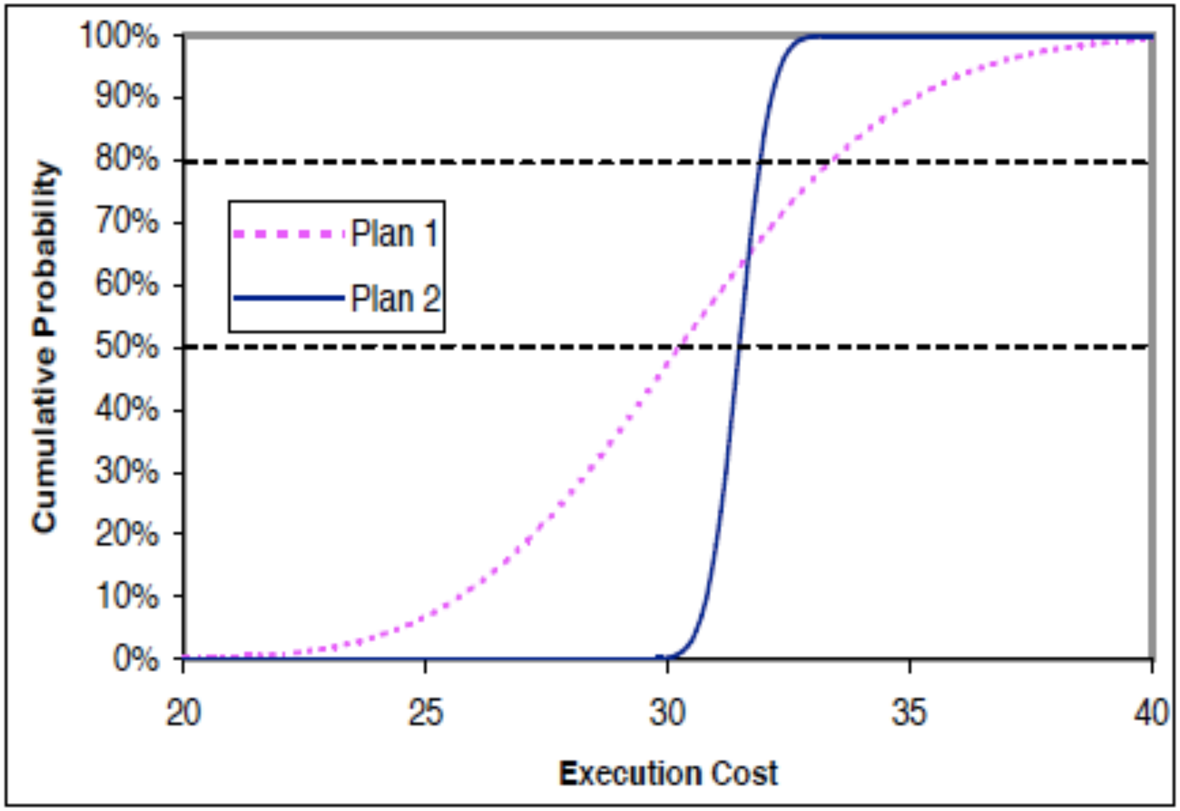


Figure 2: Probability Density Function for Execution Cost

[Babock et al. SIGMOD'2005]

Cumulative Distribution

User chooses confidence level T%.



T%=50% ➔ plans are chosen by expected cost;

T%=80% ➔ plans chosen by their cost at cumulative prob of 80%

[Babock et al. SIGMOD'2005]

The Probabilistic Database

- R has N=1,000,000,000 tuples
- Compute (offline) a sample X of size n =500

```
SELECT count(*)
FROM R
WHERE R.A=10 and R.B=20 and R.C=30
```

- Evaluate the query on the sample ➔ 8 tuples
- Thus E[p] = 8/500 = 0.0016

But what is the distribution of p ??

[Babock et al. SIGMOD'2005]

The Probabilistic Database

- R has N=1,000,000,000 tuples
- Compute (offline) a sample X of size n =500
- A fraction k = 8 of X satisfy the predicate
- An unknown fraction p of R satisfy the pred.
- Denote f(z) = density function for p:

$$Pr[(a \le p \le b)|X] = \int_a^b f(z|X)dz.$$

[Babock et al. SIGMOD'2005]

The Probabilistic Database

- Bayes' rule:

$$f(z|X) = \frac{\mathbf{Pr}[X|p = z]f(z)}{\int_0^1 \mathbf{Pr}[X|p = y]f(y)dy}$$

Next, compute each term (in class)

- What is Pr[X | p=z] ? Assume X= w/ replacement
- Whas is "the prior" f(z) ?

[Babock et al. SIGMOD’2005]

The Probabilistic Database

$$f(z|X) = \frac{z^{k-1/2}(1-z)^{n-k-1/2}}{\int_0^1 y^{k-1/2}(y-z)^{n-k-1/2}dy}$$

[Babock et al. SIGMOD’2005]

The Probabilistic Database

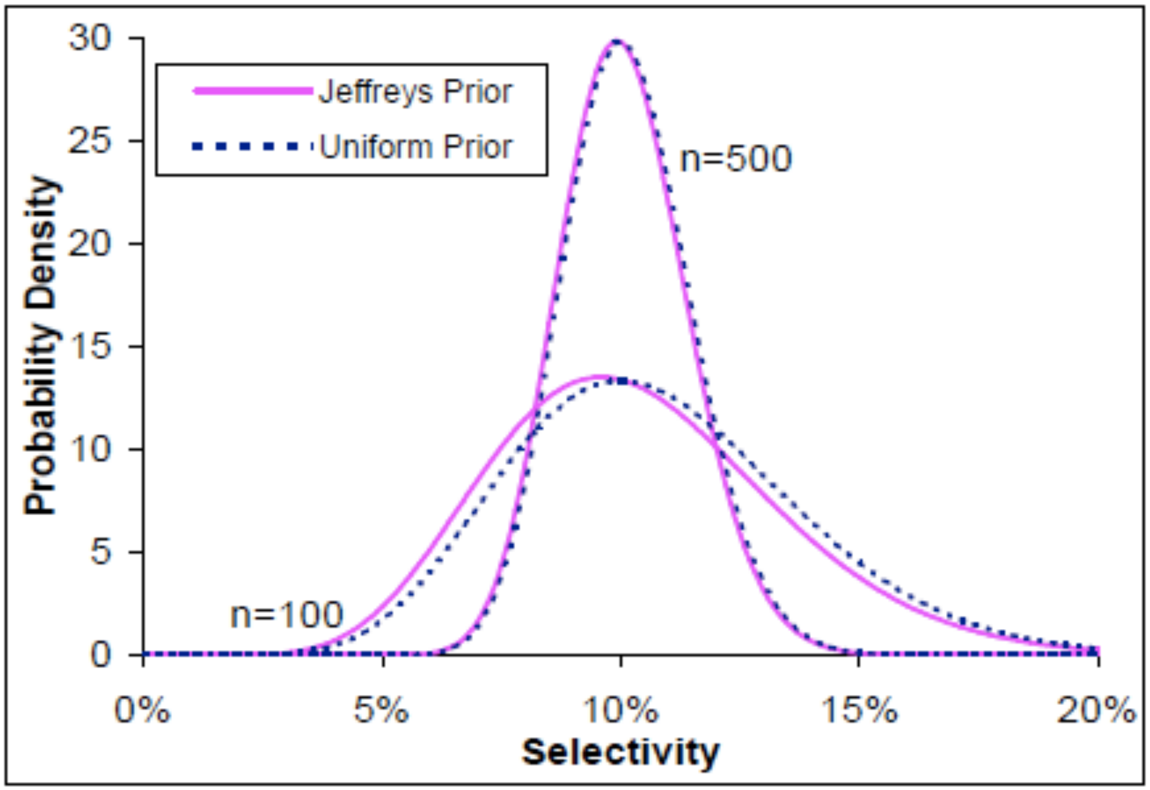


Figure 4: Sample Size Matters, Prior Doesn’t

Summary on Approx. Query Answering

- Will become increasingly relevant in the near future
- A large collection of techniques exists:
  - Sketches, samples, statistics, …
- My thesis: there exists a foundation on probabilistic databases that still has to be discovered
- Lectures 8, 9: will try to find out together…