Privacy and Background Knowledge

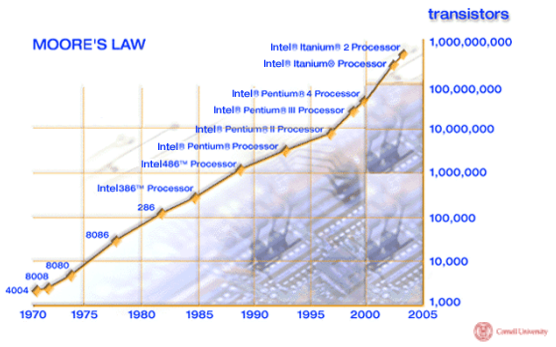
Johannes Gehrke
Department of Computer Science

An Abundance of Data

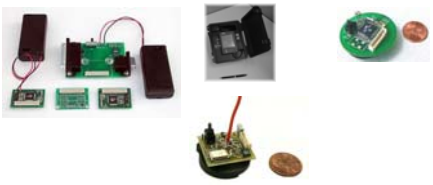
- Supermarket scanners
- Credit card transactions
- Direct mail response
- Call center records
- ATM machines
- Web server logs
- Customer web site trails
- Podcasts
- Blogs
- Scientific experiments
- Sensors
- Cameras
- Interactions in social networks
- Newswires
- Speech-to-text translation
- Email
- Closed caption

•Print, film, optical, and magnetic storage: 5 Exabytes (EB) of new information in 2002
•Doubled in the last three years

Driving Factors: A LARGE Hardware Revolution



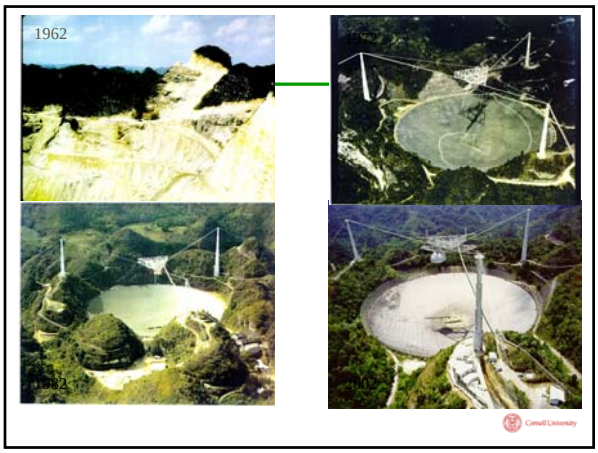
Driving Factors: A small Hardware Revolution



- Experts on ants estimate that there are 10^{16} to 10^{17} ants on earth. In the year 1997, we produced one transistor per ant.

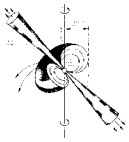
Other Driving Factors

- Gilder's law (bandwidth doubles every 6 months)
- Metcalf's law (network usefulness increases squared with the number of users)



Pulsars

- Pulsars are rotating stars
- Of interest are
  - Millisecond pulsars
  - Compact binaries
- Example:
  - Hulse-Taylor binary
  - Used to infer gravitational waves in support of Einstein's General Theory of Relativity
  - Nobel price in physics in 1993



Pulsar Surveys

- Most demanding of the ALFA surveys
  - ~ 100 MB/s to disk
  - ~ 1 PB for entire survey (3-5 yr @ 6-10% duty cycle)
- Requires coarsely parallel processing of raw data in discrete, local data chunks
  - processing time ~ 50-200x data acquisition time on single processor (Intel 2.5 GHz 512k cached with 1GB ram)
  - depends on data set details, algorithms, code
  - Distributed initial processing (Cornell + 5 sites)
- Requires meta-analysis of data products of the initial analysis
  - Database and data mining research problems

Project Requirements

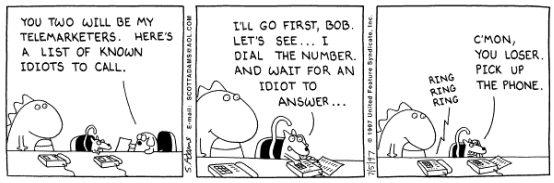
- Data
  - 14 TB every 2 weeks
  - Shipped on USB-2 disk drives
  - Need to archive raw data 5+ years
  - Need to make data products to the astronomy research community
- Processing
  - Extremely processor intensive
    - Currently just exhaustive search over a large parameter space (periodicity, dispersion, time)
  - Find new pulsars --- and other *interesting* phenomena

Driving Factors: Analysis Capabilities

Data mining is the exploration and analysis of large quantities of data in order to discover valid, novel, potentially useful, and ultimately understandable patterns in data.

Example pattern (Census Bureau Data):
If (relationship = husband), then (gender = male). 99.6%

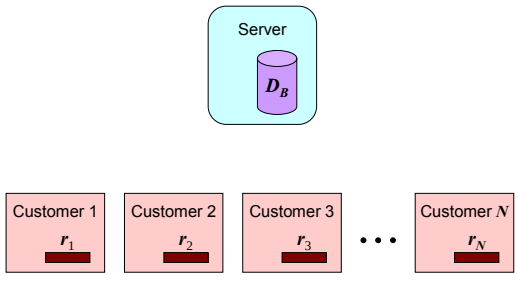
And Even the Popular Press Caught On



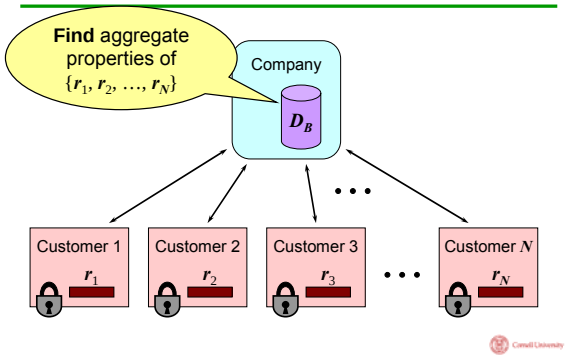
Concerns About Privacy

- S. Garfinkel, "Database Nation: The Death of Privacy in 21st Century", O' Reilly, Jan 2000

The Setup



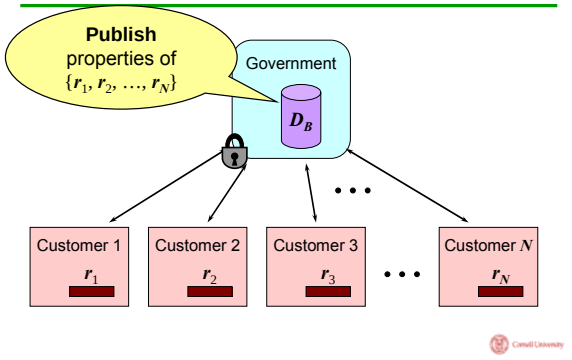
Model I: Untrusted Data Collector



Minimal Information Sharing

- Ideally, we want an algorithm that discloses only the query result, and only to the requesting party. (In practice, we need some extra disclosure.)
- How do we design algorithms that compute queries while preserving data privacy?
- How do we measure privacy (this extra disclosure)?

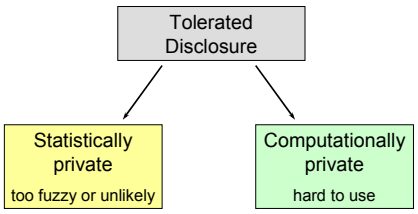
Model II: Trusted Data Collector



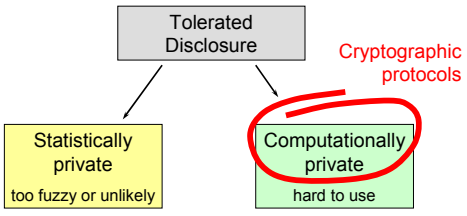
Disclosure Limitations

- Ideally, we want a solution that discloses as much statistical information as possible while preserving privacy of the individuals who contributed data.
- How do we design algorithms that allow the "largest" set of queries that can be disclosed while preserving data privacy?
- How do we measure privacy?

Types of Disclosure

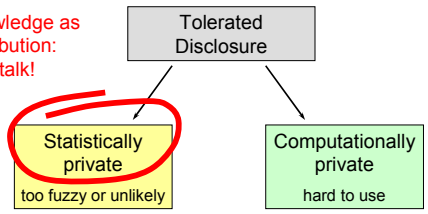


Types of Disclosure



Types of Disclosure

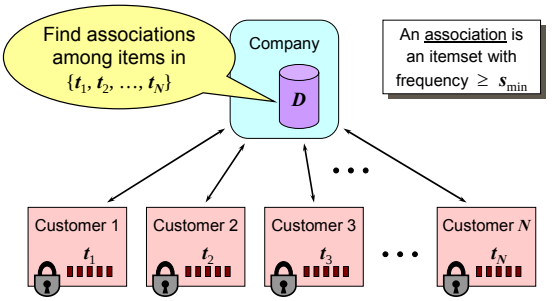
Knowledge as distribution: This talk!



Talk Outline

- Introduction
- Privacy-preserving data mining
  - Association rules
  - Problem definition
  - Privacy breaches
  - Select-A-Size randomization
  - Itemset compression
  - Experimental results
- Privacy-preserving data publishing
- Conclusions

Privacy Preserving Associations



Problem Introduction

Abstract:
- A set of items {1,2,…,k}
- A dabase of transactions (itemsets) D={t_1, t_2, …, t_n}, t_j subset {1,2,…,k}

GOAL:
Find all itemsets that appear in at least s_{min} transactions

("appear in" == "are subsets of")
$I \subseteq t$: t supports I

For an itemset I, the number of transactions it appears in is called the support of I.
s_{min} is called the minimum support.

Concrete:
- I = {milk, bread, cheese, …}
- D = { {milk,bread,cheese}, {bread,cheese,juice}, …}

GOAL:
Find all itemsets that appear in at least 1000 transactions

Transaction {milk,bread,cheese} supports itemset {milk,bread}

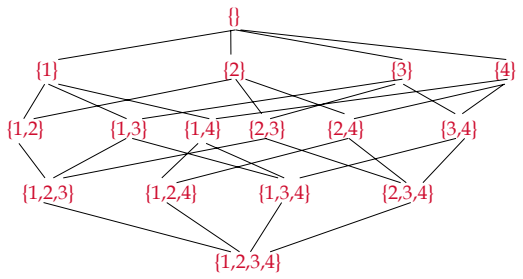
Example

Example:
- I = {1,2,3,4}
- D = {{1,2,3}, {1,2,3,4}, {1,4}, {1,2}, {3}, {1,2,3}}

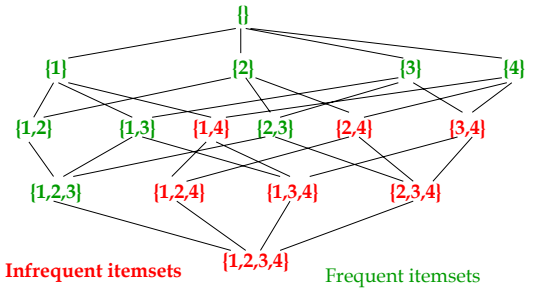
Questions:
- What is the support of {1}? {1,2}?
- Given a minimum support of 5, what is the output? 4? 3?

Observations:
- Support({1,2}) <= Support({1})
- Support({1,2}) <= Support({2})

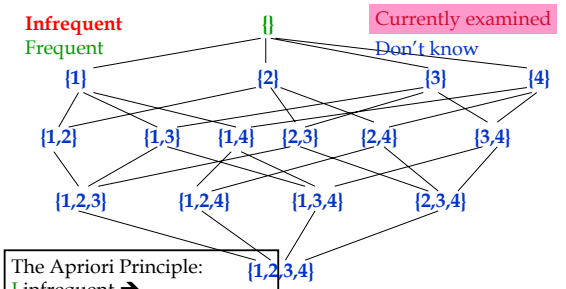
The Itemset Lattice



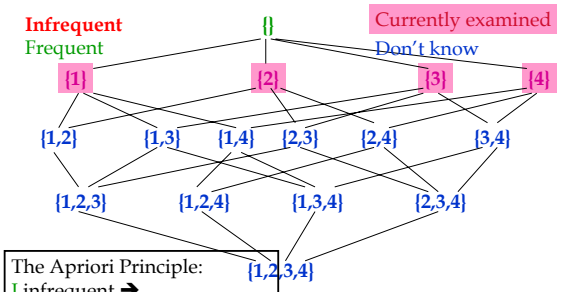
Frequent Itemsets



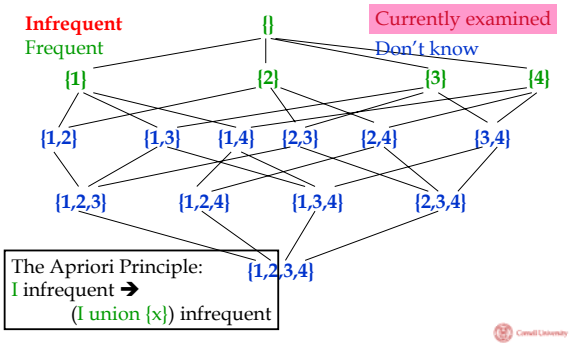
Breath First Search: 1-Itemsets



Breath First Search: 1-Itemsets

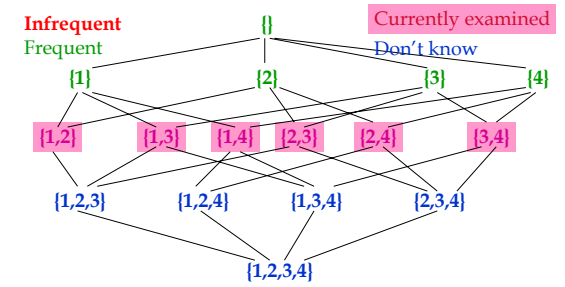


Breath First Search: 1-Itemsets

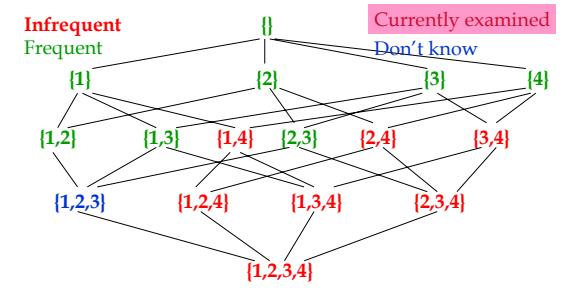


The Apriori Principle:
I infrequent ➔
(I union {x}) infrequent

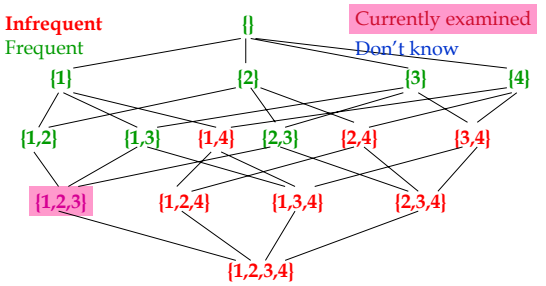
Breath First Search: 2-Itemsets



Breath First Search: 3-Itemsets



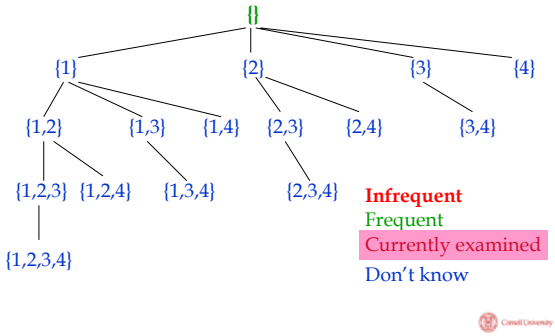
Breath First Search: 3-Itemsets



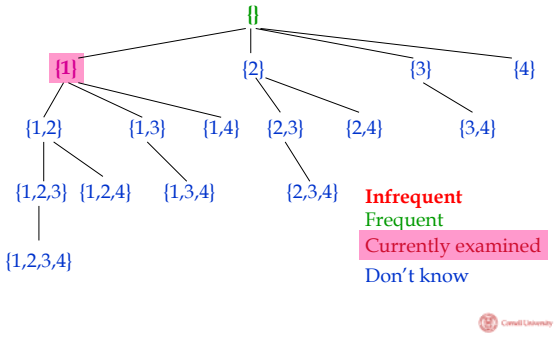
Breadth First Search: Remarks

- We prune infrequent itemsets and avoid to count them
- To find an itemset with k items, we first need to count all 2^k subsets

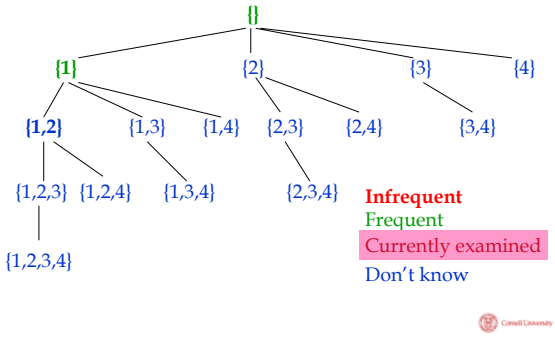
Depth First Search (1): Start



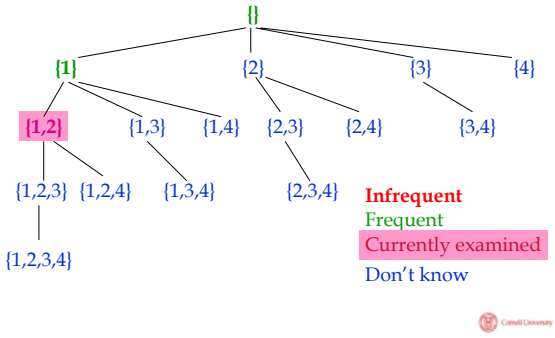
Depth First Search (2)



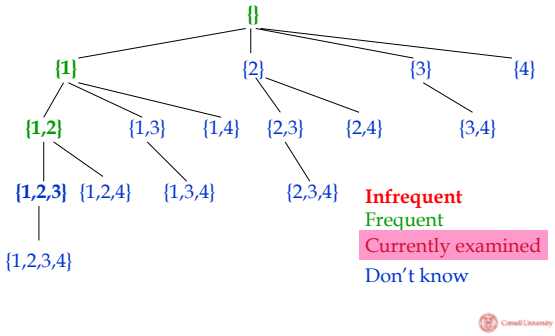
Depth First Search (3)



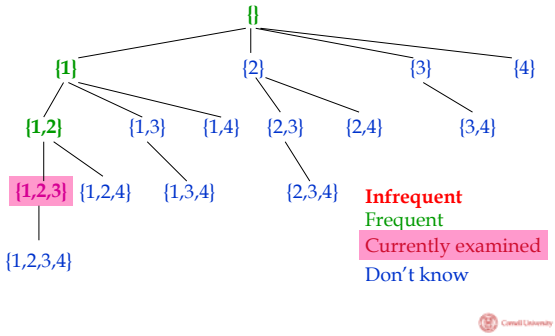
Depth First Search (4)



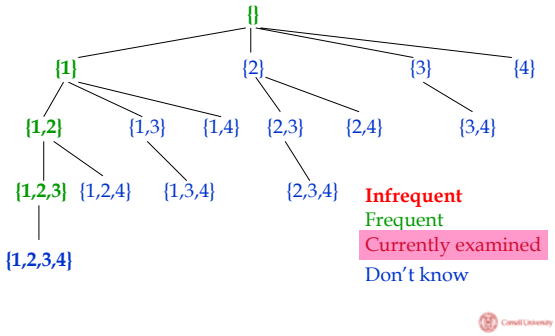
Depth First Search (5)



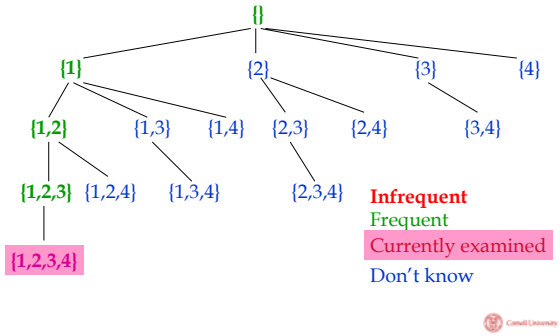
Depth First Search (6)



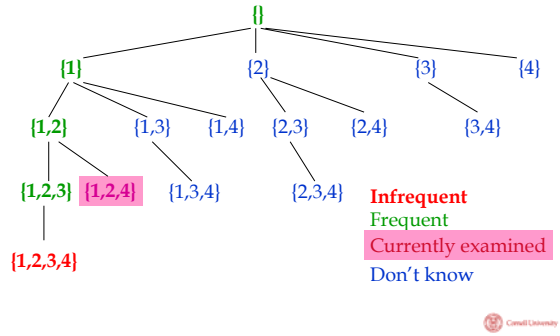
Depth First Search (7)



Depth First Search (8)



Depth First Search (9)



Depth First Search: Remarks

- We prune frequent itemsets and avoid counting them
- To find an itemset with k items, we need to count k prefixes

Associations Recap

- A transaction t is a set of items
- All transactions form a set T of transactions
- Any itemset A has support s in T if

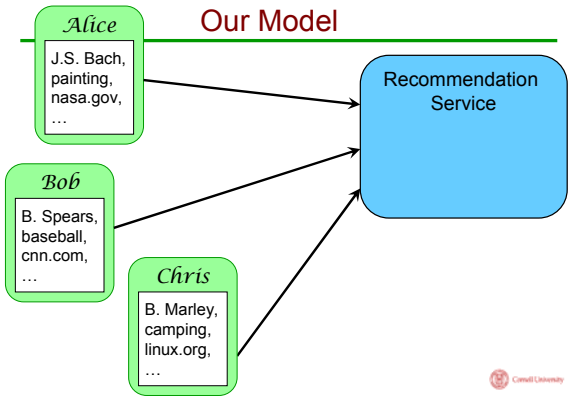
$$s = \text{supp}(A) = \frac{\#\{t \in T \mid A \subseteq t\}}{|T|}$$

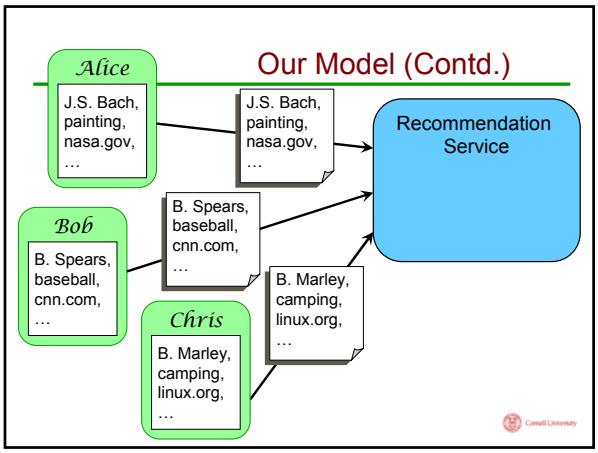
- Itemset A is frequent if $s \geq s_{min}$
- If $A \subseteq B$, then supp (A) ≥ supp (B).
- Association rule: $A \Rightarrow B$ holds when the union $A \cup B$ is frequent and: supp $(A \cup B) \geq$ supp $(A) \cdot \text{conf}_{min}$

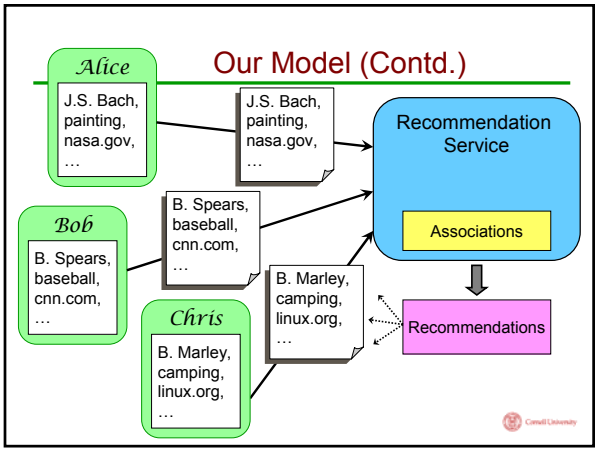
Talk Outline

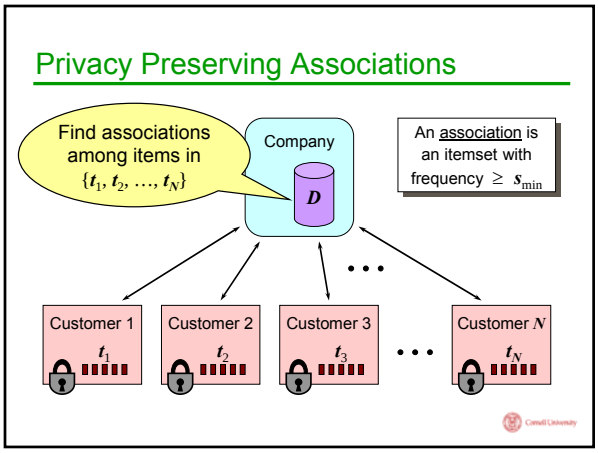
- Introduction
- Privacy-preserving data mining
  - Association rules
  - Problem definition
  - Privacy breaches
  - Select-A-Size randomization
  - Itemset compression
  - Experimental results
- Privacy-preserving data publishing
- Conclusions

Our Model




Our Model (Contd.)
Alice
J.S. Bach, painting, nasa.gov, …
J.S. Bach, painting, nasa.gov, …
Recommendation Service
Bob
B. Spears, baseball, cnn.com, …
B. Spears, baseball, cnn.com, …
B. Marley, camping, linux.org, …
Chris
B. Marley, camping, linux.org, …

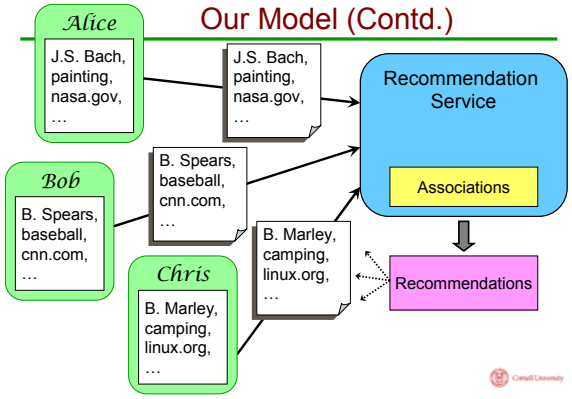

Our Model (Contd.)
Alice
J.S. Bach, painting, nasa.gov, …
J.S. Bach, painting, nasa.gov, …
Recommendation Service
Associations
Bob
B. Spears, baseball, cnn.com, …
B. Spears, baseball, cnn.com, …
B. Marley, camping, linux.org, …
Recommendations
Chris
B. Marley, camping, linux.org, …


Privacy Preserving Associations
Find associations among items in $\{t_1, t_2, \ldots, t_N\}$
Company
D
An association is an itemset with frequency ≥ s_{min}
Customer 1
t_1
Customer 2
t_2
Customer 3
t_3
Customer N
t_N

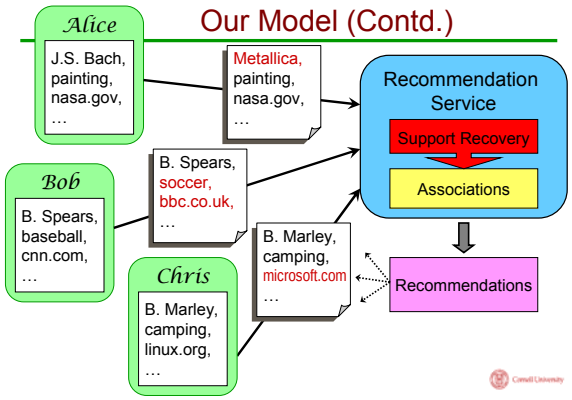
Minimal Information Sharing

- Ideally, we want an algorithm that discloses only the association rules
- However, in practice, we need some extra disclosure.

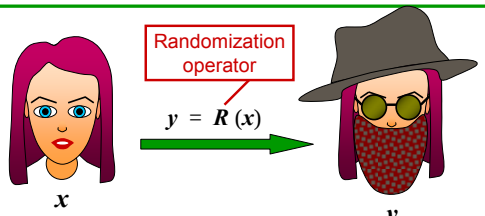
Our Model (Contd.)



Our Model (Contd.)



Our Model: Another View



Original (private) data

Assumptions:

- Described by a random variable X.
- Each client is independent.

Randomized data

Described by a random variable $Y = R(X)$.

The Problem

- How to randomize transactions so that
  - we can find frequent itemsets
  - while preserving privacy at transaction level?

The Randomized Response Model

[Stanley Warner, JASA 1965]

- Respondents are given:
  1. A source of randomness for YES and NO answers (a biased coin)
  2. A statement: I am teaching database systems with R+G DBMS, 3rd edition.
- The procedure:
  - Respondent flips the coin
  - Answers YES iff coin gives correct answer, answers NO otherwise

Another View: Two Questions

- Respondents are given:
  1. The coin
  2. Two logically opposite statements:
     - S: I am teaching database systems with R+G DBMS.
     - S^{bar}: I am not teaching database systems with R+G DBMS.
- The procedure:
  - Respondent flips the coin
  - Answers either statement S1 or S2.

Analysis

π = the true probability of S in the population.
p = the probability that the coin says YES.

Y_i = 1 if the i^{th} respondent says 'yes'.
0 if the i^{th} respondent reports 'no'.

- $P(Y_i=1) = \pi p + (1-\pi)(1-p) = p_{YES}$
- $P(Y_i=0) = (1-\pi)p + \pi(1-p) = p_{NO}$

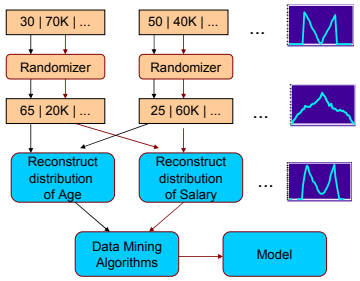
Analysis (Contd.)

- Assume a sample with n records, n1 say YES, (n-n1) say NO
- Likelihood of this sample:
  - $L = p_{YES}^{n1}\ p_{NO}^{(n-n1)}$
  - This gives a maximum likelihood estimate of $\pi^{hat} = (p-1)/(2p-1) + n1/n(2p-1)$
- Easy to show:
  - $E(\pi^{hat}) = \pi$
  - $Var(\pi^{hat}) = \pi(1-\pi)]/n + [1/[16(p-0.5)^2]-0.25]/n$

Source of Variance Sampling Coin Flips

Interval Privacy

- Agrawal & Srikant, SIGMOD 2000



Interval Privacy: Quantifying Privacy

- Add a random value between -30 and +30 to age.
- If randomized value is 60
  - know with 90% confidence that age is between 33 and 87.
- Interval width = amount of privacy.
  - Example:
    - Interval Width 54 with 90% confidence
    - Interval Width 60 with 100% confidence

Talk Outline

- Introduction
- Privacy-preserving data mining
  - Association rules
  - Problem definition
  - Privacy breaches
  - Select-A-Size randomization
  - Itemset compression
  - Experimental results
- Privacy-preserving data publishing
- Conclusions

Background Knowledge

A randomization may "look strong" but sometimes fail to hide some items of an individual transaction.

- Simple randomization example: Given a transaction
  - Keep item with 20% probability,
  - Replace with a new random item with 80% probability.

Example: $\{a, b, c\}$

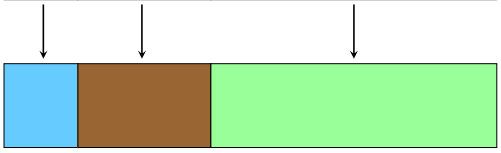
10 M transactions of size 10 with 10 K items:

1% have $\{a, b, c\}$	5% have $\{a, b\}$, $\{a, c\}$, or $\{b, c\}$ only	94% have one or zero items of $\{a, b, c\}$

Example: $\{a, b, c\}$

10 M transactions of size 10 with 10 K items:

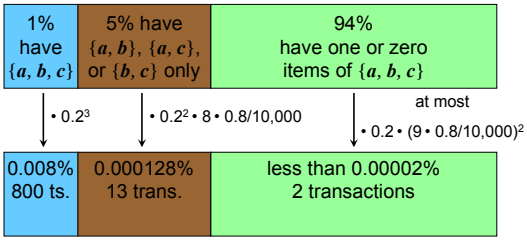
1% have $\{a, b, c\}$	5% have $\{a, b\}$, $\{a, c\}$, or $\{b, c\}$ only	94% have one or zero items of $\{a, b, c\}$



After randomization: How many have $\{a, b, c\}$?

Example: $\{a, b, c\}$

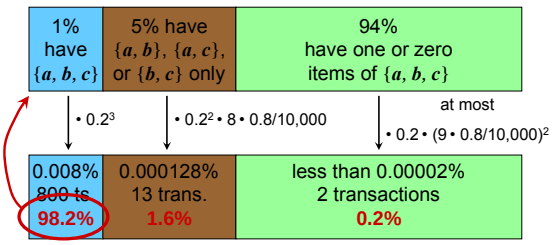
10 M transactions of size 10 with 10 K items:



After randomization: How many have $\{a, b, c\}$?

Example: $\{a, b, c\}$

10 M transactions of size 10 with 10 K items:



After randomization: How many have $\{a, b, c\}$?

Example: {a, b, c}

- Given nothing, we have only 1% probability that {a, b, c} occurs in the original transaction
- Given {a, b, c} in the randomized transaction, we have about 98% certainty of {a, b, c} in the original transaction.
- This is what we call a privacy breach.
- The example randomization preserves privacy "on average," but not "in the worst case."

Privacy Breaches

- A randomization may "look strong" but sometimes fails to hide properties of an individual transaction.
- Note: Interval privacy has the same problem
  - Assume interval privacy [-30,30]
  - Assume you see an age value y = 130

Simple Privacy Breaches

- Suppose the "adversary" wants to know if $z \in t$, where
  - t is an original transaction;
  - t′ is the corresponding randomized transaction;
  - A is an itemset
- Itemset A causes a privacy breach of level β (e.g. 50%) if:

$$\text{Prob}\left[z \in t \mid A \subseteq t'\right] \geq \beta$$

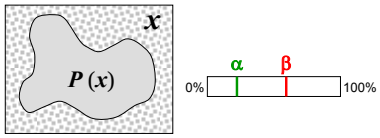
- Knowledge of $A \subseteq t'$ makes a jump from Prob [$z \in t$] to Prob [$z \in t \mid A \subseteq t'$] (in the adversary's viewpoint).

Privacy Breaches: Goals

- We want a bound for all privacy breaches
  - not only for: item $\in$ t versus itemset $\subseteq$ t′
- No knowledge of data distribution is required in advance
  - We should not need to know Prob [item $\in$ t]
- Applicable to numerical data as well
- Easy to work with, even for complex randomizations

α-to-β Privacy Breach

Let $P(x)$ be any property of client's private data;
Let $0 < \alpha < \beta < 1$ be two probability thresholds.



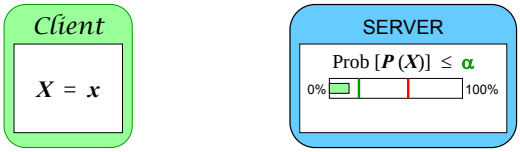
Example:

$P(x)$ = "transaction x contains $\{a, b, c\}$"

$\alpha = 1\%$ and $\beta = 50\%$

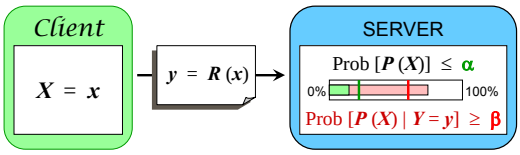
α-to-β Privacy Breach

Let $P(x)$ be any property of client's private data;
Let $0 < \alpha < \beta < 1$ be two probability thresholds.



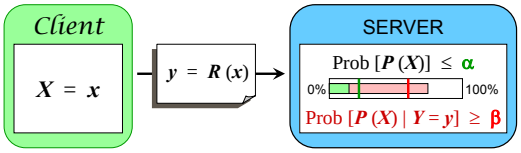
α-to-β Privacy Breach

Let $P(x)$ be any property of client's private data;
Let $0 < \alpha < \beta < 1$ be two probability thresholds.



α-to-β Privacy Breach

Let $P(x)$ be any property of client's private data;
Let $0 < \alpha < \beta < 1$ be two probability thresholds.



Disclosure of y causes an α-to-β privacy breach w.r.t. property $P(x)$.

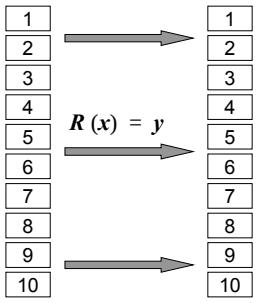
α-to-β Privacy Breach

Checking for α-to-β privacy breaches:

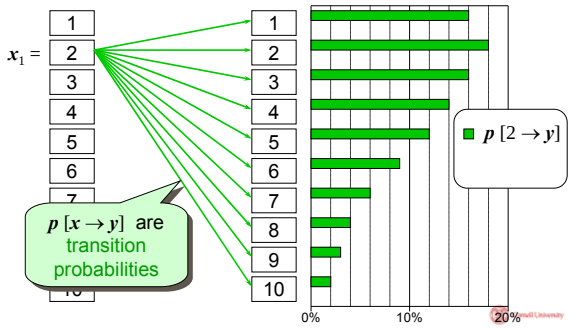
- There are exponentially many properties $P(x)$;
- We have to know the data distribution in order to check whether $\text{Prob}[P(X)] \leq \alpha$ and $\text{Prob}[P(X) \mid Y = y] \geq \beta$

Is there a simple property of randomization operator R that limits privacy breaches?

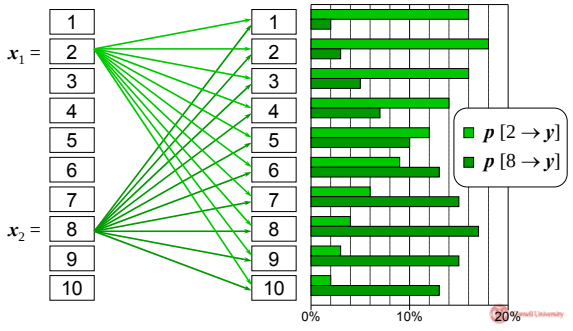
Amplification Condition



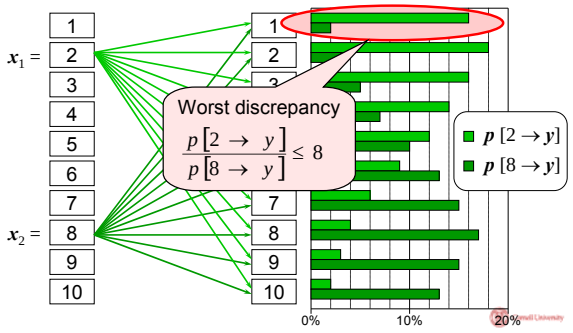
Amplification Condition



Amplification Condition



Amplification Condition



Amplification Condition

Definition:

- Randomization operator $\boldsymbol{R}$ is called "at most γ-amplifying" if:

$$\max_{x_1, x_2} \max_{y} \frac{p[x_1 \to y]}{p[x_2 \to y]} \leq \gamma$$

  - Transition probabilities $\boldsymbol{p}[\boldsymbol{x} \to \boldsymbol{y}] = \text{Prob}[\boldsymbol{R}(\boldsymbol{x}) = \boldsymbol{y}]$ depend only on the operator $\boldsymbol{R}$ and not on data.
  - We assume that all $\boldsymbol{y}$ have a nonzero probability.
- The bigger γ is, the more may be revealed about $\boldsymbol{x}$.

The Bound on α-to-β Breaches

Theorem:

- If randomization operator $\boldsymbol{R}$ is at most γ-amplifying, and if:

$$\gamma < \frac{\beta}{\alpha} \cdot \frac{1-\alpha}{1-\beta}$$

- Then, revealing $\boldsymbol{R}(\boldsymbol{X})$ to the server will never cause an α-to-β privacy breach.

The Bound on α-to-β Breaches

Examples:

- 5%-to-50% privacy breaches do not occur for $\gamma < 19$:

$$\frac{0.5}{0.05} \cdot \frac{1-0.05}{1-0.5} = 19$$

- 1%-to-98% privacy breaches do not occur for $\gamma < 4851$:

$$\frac{0.98}{0.01} \cdot \frac{1-0.01}{1-0.98} = 4851$$

- 50%-to-100% privacy breaches do not occur for any finite γ.

Amplification: Summary

- An α-to-β privacy breach w.r.t. property P (x) occurs when
  - Prob [P is true] $\leq \alpha$
  - Prob [P is true | Y = y] $\geq \beta$.
- Amplification methodology limits privacy breaches by just looking at transitional probabilities of randomization.
  - Does not use data distribution:

$$\max_{x_1, x_2} \max_{y} \frac{p[x_1 \rightarrow y]}{p[x_2 \rightarrow y]} \leq \gamma$$

Talk Outline

- Introduction
- Privacy-preserving data mining
  - Association rules
  - Problem definition
  - Privacy breaches
  - Select-A-Size randomization
  - Itemset compression
  - Experimental results
- Privacy-preserving data publishing
- Conclusions

Definition of select-a-size

- Given transaction t of size m, construct $t' = R(t)$:

t = $a, b, c, d, e, f, u, v, w$

t' =

Definition of select-a-size

- Given transaction t of size m, construct $t' = R(t)$:
  - Choose a number $j \in \{0, 1, \ldots, m\}$ with distribution $\{p[j]\}_{0..m}$;

t = $a, b, c, d, e, f, u, v, w$

t' = [ ]

$j = 4$

Definition of select-a-size

- Given transaction t of size m, construct $t' = R(t)$:
  - Choose a number $j \in \{0, 1, \ldots, m\}$ with distribution $\{p[j]\}_{0..m}$;
  - Include exactly j items of t into t';

t = $a, b, c, d, e, f, u, v, w$

t' = b, e, u, w

$j = 4$

Definition of select-a-size

- Given transaction t of size m, construct $t' = R(t)$:
  - Choose a number $j \in \{0, 1, \ldots, m\}$ with distribution $\{p[j]\}_{0..m}$;
  - Include exactly j items of t into t';
  - Each other item (not from t) goes into t' with probability ρ.

The choice of $\{p[j]\}_{0..m}$ and ρ is based on the desired privacy level.

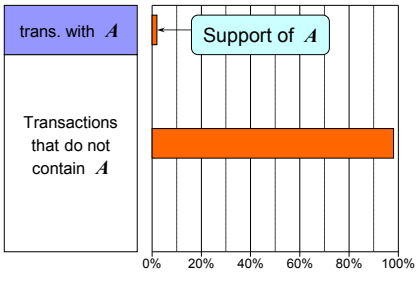
t = $a, b, c, d, e, f, u, v, w$

t' = b, e, u, w | œ, å, ß, §, ψ, €, ϰ, ъ, ђ, …

$j = 4$ items inserted with prob. ρ

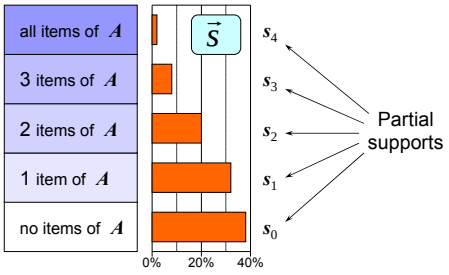
Support Recovery

Let itemset A have four items ($k = 4$).



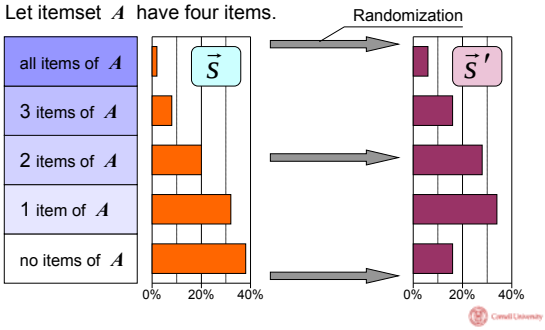
Support Recovery

Let itemset A have four items ($k = 4$).



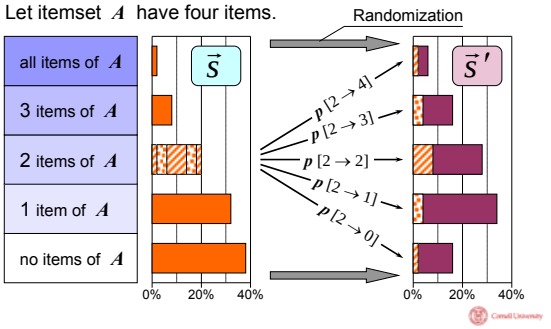
Support Recovery

Let itemset A have four items.



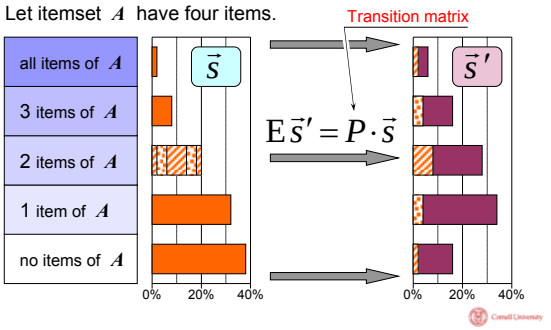
Support Recovery

Let itemset A have four items.



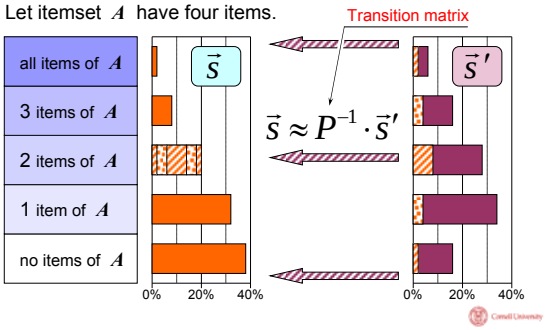
Support Recovery

Let itemset A have four items.



Support Recovery

Let itemset A have four items.



The Unbiased Estimators

- Given randomized partial supports, we can estimate original partial supports:

$$\vec{s}_{\text{est}} = Q \cdot \vec{s}', \quad \text{where} \quad Q = P^{-1}$$

- Covariance matrix for this estimator:

$$\text{Cov}\,\vec{s}_{\text{est}} = \frac{1}{|T|}\sum_{l=0}^{k} s_l \cdot Q\, D[l] Q^T,$$

$$\text{where} \quad D[l]_{i,j} = P_{i,l} \cdot \delta_{i=j} - P_{i,l} \cdot P_{j,l}$$

- To estimate it, substitute s_l with $(s_{\text{est}})_l$.
  - Special case: estimators for support and its variance
- [RH02] reconstruct statistics similarly

Apriori [AS94]

Let $k = 1$, candidate sets = all 1-itemsets.

Repeat:

1. Count support for all candidate sets
2. Output the candidate sets with support $\geq s_{\min}$
3. New candidate sets = all $(k+1)$-itemsets s.t. all their k-subsets are candidate sets with support $\geq s_{\min}$
4. Let $k = k + 1$

Stop when there are no more candidate sets.

The Modified Apriori

Let $k = 1$, candidate sets = all 1-itemsets.

Repeat:

1. Estimate support and variance (σ^2) for all candidate sets
2. Output the candidate sets with support $\geq s_{\min}$
3. New candidate sets = all $(k+1)$-itemsets s.t. all their k-subsets are candidate sets with support $\geq s_{\min} - \sigma$
4. Let $k = k + 1$

Stop when there are no more candidate sets, or the estimator's precision becomes unsatisfactory.

Talk Outline

- Introduction
- Privacy-preserving data mining
  - Association rules
  - Problem definition
  - Privacy breaches
  - Select-A-Size randomization
  - Itemset compression
  - Experimental results
- Privacy-preserving data publishing
- Conclusions

Select-A-Size Revisited

- Given transaction x of size m, construct $y = R(x)$:
  - Choose a number $j \in \{0, 1, \ldots, m\}$ with distribution $\{p[j]\}_{0..m}$;
  - Include exactly j items of x into y;
  - Each other item (not from x) goes into y with probability ρ.

The choice of $\{p[j]\}_{0..m}$ and ρ is based on the desired privacy level.

x = a, b, c, d, e, f, u, v, w

y = b, e, u, w | œ, å, ß, §, ψ, €, ϰ, ъ, ђ, …

j items — items inserted with prob. ρ

Select-A-Size Revisited

- Given transaction x of size m, construct $y = R(x)$:
  - Choose a number $j \in \{0, 1, \ldots, m\}$ with distribution $\{p[j]\}_{0..m}$;
  - Include exactly j items of x into y;
  - Each other item (not from x) goes into y with probability ρ.

The choice of $\{p[j]\}_{0..m}$ and ρ is b… level.

Often $\rho \approx 0.5$ and $n \approx 10\ldots100$ K items, making y HUGE!

x = a, b, c, d, e, f, u, v, w

y = b, e, u, w | œ, å, ß, §, ψ, €, ϰ, ъ, ђ, …

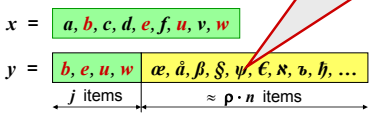
j items — $\approx \rho \cdot n$ items

Select-A-Size Revisited

- Given transaction x of size m, construct $y = R(x)$:
  - Choose a number $j \in \{0, 1, \ldots, m\}$ with distribution $\{p[j]\}_{0..m}$;
  - Include exactly j items of x int
  - Each other item (not from x) goe

The choice of $\{p[j]\}_{0..m}$ and ρ is ba
level.

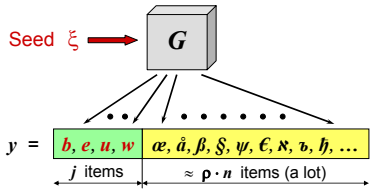
Often $\rho \approx 0.5$ and $n \approx 10\ldots100$ K items, making y HUGE!



- Let $m = 10$, $n = 100\,000$, mining itemsets of size ≤ 5.
- For $\rho = 0.5$: 100 000 bits per transaction;

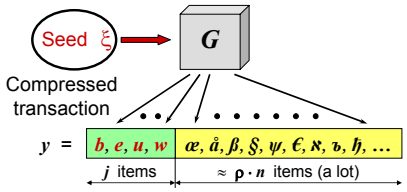
The Idea of Compression

- The idea: Let the items in y be computed by a pseudorandom number generator.



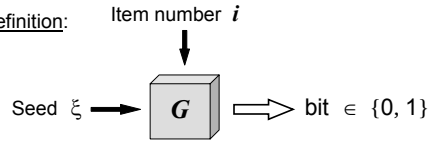
The Idea of Compression

- The idea: Let the items in y be computed by a pseudorandom number generator.



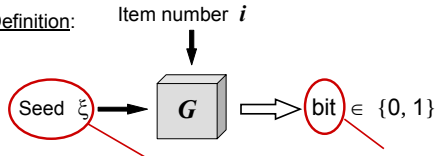
Pseudorandom Generator

- Definition:



Pseudorandom Generator

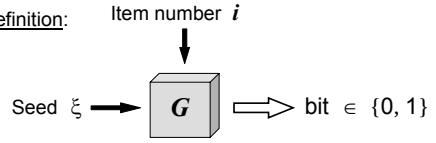
- Definition:



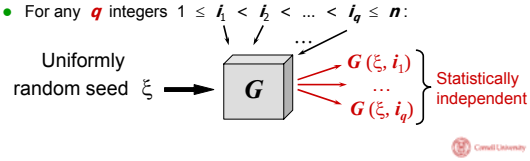
- If seed ξ is uniformly random, Prob [$G(\xi, i) = 1$] = ρ

Pseudorandom Generator

- Definition:



- If seed ξ is uniformly random, Prob [$G(\xi, i) = 1$] = ρ
- For any q integers $1 \le i_1 < i_2 < \ldots < i_q \le n$:



Itemset Compression

- Given transaction x of size m, construct $y = R(x)$:
  - (as before) Choose a number $j \in \{0, 1, ..., m\}$ with distribution $\{p[j]\}_{0..m}$;
  - (as before) Choose exactly j items of x (to include into y);

x = $a, b, c, d, e, f, u, v, w$

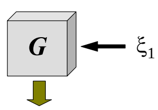
b, e, u, w

y =

Itemset Compression

- Given transaction x of size m, construct $y = R(x)$:
  - (as before) Choose a number $j \in \{0, 1, ..., m\}$ with distribution $\{p[j]\}_{0..m}$;
  - (as before) Choose exactly j items of x (to include into y);
  - Choose a seed ξ uniformly at random

x = $a, b, c, d, e, f, u, v, w$



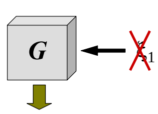
b, e, u, w

y = a, e, f, v, w | ŋ, ë, њ, چ, ώ, ¥, ב, ŭ, …

Itemset Compression

- Given transaction x of size m, construct $y = R(x)$:
  - (as before) Choose a number $j \in \{0, 1, ..., m\}$ with distribution $\{p[j]\}_{0..m}$;
  - (as before) Choose exactly j items of x (to include into y);
  - Choose a seed ξ uniformly at random conditioned by:
    - For all items in X, $G(\xi, \text{item}) = 1$ iff the item is chosen above.

x = $a, b, c, d, e, f, u, v, w$

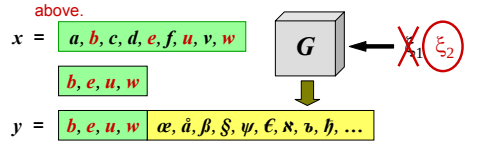


b, e, u, w

y = ~~a, e, f, v, w~~ | ŋ, ë, њ, چ, ώ, ¥, ב, ŭ, …

Itemset Compression

- Given transaction x of size m, construct $y = R(x)$:
  - as before:
    - Choose a number $j \in \{0, 1, \ldots, m\}$ with distribution $\{p[j]\}_{0..m}$;
    - Choose exactly j items of x (to include into y);
  - Choose a seed ξ uniformly at random conditioned by:
    - For all items in x, $G(\xi, \text{item}) = 1$ iff the item is chosen above.



"Transparency" of Compression

New transactions — the same old algorithms:

- We can do support recovery the same way as if there is no compression (for small itemsets);
- We can check amplification condition and select randomization parameters the same way as if there is no compression.

The bits produced by the pseudorandom generator must be q-wise independent, where

q = max. transaction size + max. association size

Compression in Practice

- Suppose we use Bose-Chaudhuri-Hocquenghem (BCH) error-correcting codes for pseudorandom generators.
- Let $m = 10$, $n = 100\,000$, mining itemsets of size ≤ 5.
- For $\rho = 0.5$:
  - "Ordinary" way: 100 000 bits per transaction;
  - "Compressed" way: 136 bits per transaction.
- For $\rho = 1/16$:
  - "Ordinary" way: $100\,000 \cdot H(1/16) \approx 33\,729$ bits per transaction;
  - "Compressed" way: 570 bits per transaction.

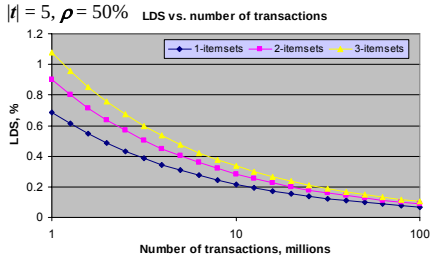
Talk Outline

- Introduction
- Privacy-preserving data mining
  - Association rules
  - Problem definition
  - Privacy breaches
  - Select-A-Size randomization
  - Itemset compression
  - Experimental results
- Privacy-preserving data publishing
- Conclusions

Lowest Discoverable Support

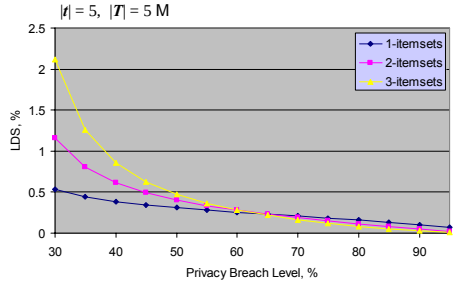
- LDS is s.t., when predicted, it is 4σ away from zero.
- Roughly, LDS is proportional to $1/\sqrt{|T|}$

$|t| = 5$, $\rho = 50\%$



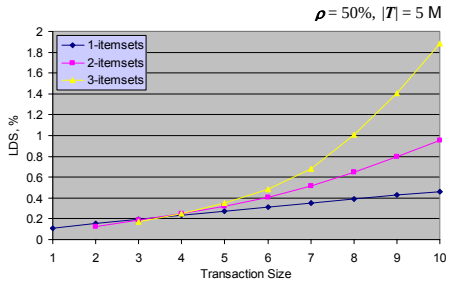
LDS vs. Breach Level

$|t| = 5$, $|T| = 5$ M



- Reminder: breach level is the limit on $\Pr[z \in t \mid A \subseteq t']$

LDS vs. Transaction Size



- Very long transactions cannot be used for prediction

Real datasets: Soccer, Mailorder

- Soccer is the clickstream log of WorldCup'98 web site, split into sessions of HTML requests.
  - 11 K items (HTMLs), 6.5 M transactions
  - Available at http://www.acm.org/sigcomm/ITA/
- Mailorder is a purchase dataset from an on-line store
  - Products are replaced with their categories
  - 96 items (categories), 2.9 M transactions

A small fraction of transactions are discarded as too long.

- longer than 10 (for soccer) or 7 (for mailorder)

Modified Apriori on Real Data

Breach level = 50%. Inserted 20-50% items to each transaction.

Soccer:

$s_{min} = 0.2\%$

$\sigma \approx 0.07\%$ for 3-itemsets

Itemset Size	True Itemsets	True Positives	False Drops	False Positives
1	266	254	12	31
2	217	195	22	45
3	48	43	5	26

Mailorder:

$s_{min} = 0.2\%$

$\sigma \approx 0.05\%$ for 3-itemsets

Itemset Size	True Itemsets	True Positives	False Drops	False Positives
1	65	65	0	0
2	228	212	16	28
3	22	18	4	5

False Drops False Positives

Soccer

Pred. supp%, when true supp ≥ 0.2%

Size	< 0.1	0.1-0.15	0.15-0.2	≥0.2
1	0	2	10	254
2	0	5	17	195
3	0	1	4	43

True supp%, when pred. supp ≥ 0.2%

Size	< 0.1	0.1-0.15	0.15-0.2	≥0.2
1	0	7	24	254
2	7	10	28	195
3	5	13	8	43

Mailorder

Pred. supp%, when true supp ≥ 0.2%

Size	< 0.1	0.1-0.15	0.15-0.2	≥0.2
1	0	0	0	65
2	0	1	15	212
3	0	1	3	18

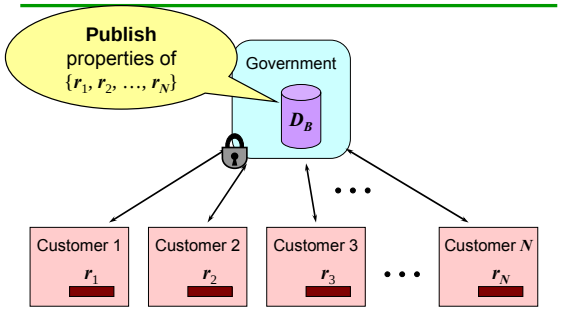
True supp%, when pred. supp ≥ 0.2%

Size	< 0.1	0.1-0.15	0.15-0.2	≥0.2
1	0	0	0	65
2	0	0	28	212
3	1	2	2	18

Talk Outline

- Introduction
- Privacy-preserving data mining
- Privacy-preserving data publishing
  - K-Anonymity
  - Attacks
  - L-Diversity
- Conclusions

Trusted Data Collector



Disclosure Limitations

- Ideally, we want a solution that discloses as much statistical information as possible while preserving privacy of the individuals who contributed data.
- How do we design algorithms that compute the "largest" set of queries that can be disclosed while preserving data privacy?

Sample Microdata

SSN	Zip	Age	Nationality	Disease
631-35-1210	13053	28	Russian	Heart
051-34-1430	13068	29	American	Heart
120-30-1243	13068	21	Japanese	Viral
070-97-2432	13053	23	American	Viral
238-50-0890	14853	50	Indian	Cancer
265-04-1275	14853	55	Russian	Heart
574-22-0242	14850	47	American	Viral
388-32-1539	14850	59	American	Viral
005-24-3424	13053	31	American	Cancer
248-223-2956	13053	37	Indian	Cancer
221-22-9713	13068	36	Japanese	Cancer
615-84-1924	13068	32	American	Cancer

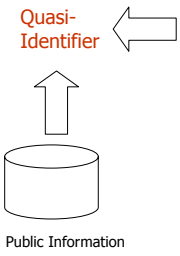
Removing SSN …

Zip	Age	Nationality	Disease
13053	28	Russian	Heart
13068	29	American	Heart
13068	21	Japanese	Viral
13053	23	American	Viral
14853	50	Indian	Cancer
14853	55	Russian	Heart
14850	47	American	Viral
14850	59	American	Viral
13053	31	American	Cancer
13053	37	Indian	Cancer
13068	36	Japanese	Cancer
13068	32	American	Cancer

Medical Records of a hospital near Ithaca serving patients from

- Freeville (13068)
- Dryden (13053)
- Ithaca (14850, 14853)

Linkage Attacks

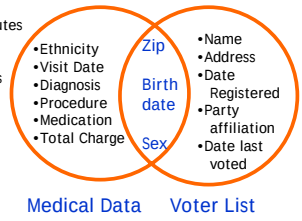


Zip	Age	Nationality	Disease
13053	28	Russian	Heart
13068	29	American	Heart
13068	21	Japanese	Viral
13053	23	American	Viral
14853	50	Indian	Cancer
14853	55	Russian	Heart
14850	47	American	Viral
14850	59	American	Viral
13053	31	American	Cancer
13053	37	Indian	Cancer
13068	36	Japanese	Cancer
13068	32	American	Cancer

Linkage Attacks (Contd.)

- Medical Data was considered anonymous, since identifying attributes were removed.
- Governor of Massachusetts, was uniquely identified by the attributes Zip, Birth Date, Sex
- Hence, his private medical records were out in the open

- {Zip, Birth Date, Sex} *Quasi-Identifier*
- 87 percent of US population uniquely identified using the above Quasi Identifier [S02]



Quasi-Identifiers and Sensitive Attributes

Zip	Age	Nationality	Disease
13053	28	Russian	Heart
13068	29	American	Heart
13068	21	Japanese	Viral
13053	23	American	Viral
14853	50	Indian	Cancer
14853	55	Russian	Heart
14850	47	American	Viral
14850	59	American	Viral
13053	31	American	Cancer
13053	37	Indian	Cancer
13068	36	Japanese	Cancer
13068	32	American	Cancer

- Base Table: Medical Records of a hospital near Ithaca serving patients from Freeville (13068), Dryden (13053), and Ithaca (14850, 14853)
- The combination {Zip, Age, Nationality} is the **quasi-identifier**
- Disease is the **sensitive attribute**

K-Anonymity [Sweeney02]

- Generalize, modify, or distort quasi-identifier values so that no individual is uniquely identifiable from a group of *k*
- In SQL, table T is k-anonymous if each

```
SELECT COUNT(*)
FROM T
GROUP BY Quasi-Identifier
```

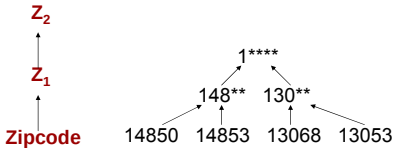
  is ≥ k
- Parameter k indicates the "degree" of anonymity

K-Anonymity

- There are at least k tuples sharing the same values for each combination of the quasi-identifiers.
- Techniques
  - Generalizing non-sensitive attributes
  - Tuple Suppression
  - Data Swapping
  - Randomization

K-Anonymity Through Generalization

- Generalization functions induce value generalization hierarchies
- Corresponding domain generalization hierarchies



Example Microdata

Zip	Age	Nationality	Disease
13053	28	Russian	Heart
13068	29	American	Heart
13068	21	Japanese	Viral
13053	23	American	Viral
14853	50	Indian	Cancer
14853	55	Russian	Heart
14850	47	American	Viral
14850	59	American	Viral
13053	31	American	Cancer
13053	37	Indian	Cancer
13068	36	Japanese	Cancer
13068	32	American	Cancer

4-Anonymous Microdata

Zip	Age	Nationality	Disease
130**	<30	*	Heart
130**	<30	*	Heart
130**	<30	*	Viral
130**	<30	*	Viral
1485*	>40	*	Cancer
1485*	>40	*	Heart
1485*	>40	*	Viral
1485*	>40	*	Viral
130**	30-40	*	Cancer
130**	30-40	*	Cancer
130**	30-40	*	Cancer
130**	30-40	*	Cancer

K-Anonymity Algorithms

- Optimal Full-Domain Algorithms
  - Binary Search [Sa01] of the lattice finds solution of minimum height
- Optimal Algorithms:
  - Bayardo-Agrawal [BA05]
  - Levefre-DeWitt-Ramakrishnan [LDR05]
- Heuristic Algorithms
  - Greedy Heuristic Search [Sw02-2, FWY05, WYC04]
  - No guarantees about optimality
- Stochastic Search
  - Genetic Algorithms [Iy02]
  - Simulated Annealing [Wi02]
  - Llong run times to convergence; do not guarantee optimality
- Approximation Algorithms
  - Cell-suppression [MW04, AFKM+05]
  - Have not been implemented

Talk Outline

- Introduction
- Privacy-preserving data mining
- Privacy-preserving data publishing
  - K-Anonymity
  - Attacks
  - L-Diversity
- Conclusions

Example Microdata

Zip	Age	Nationality	Disease
13053	28	Russian	Heart
13068	29	American	Heart
13068	21	Japanese	Viral
13053	23	American	Viral
14853	50	Indian	Cancer
14853	55	Russian	Heart
14850	47	American	Viral
14850	59	American	Viral
13053	31	American	Cancer
13053	37	Indian	Cancer
13068	36	Japanese	Cancer
13068	32	American	Cancer

4-Anonymous Microdata

Zip	Age	Nationality	Disease
130**	<30	*	Heart
130**	<30	*	Heart
130**	<30	*	Viral
130**	<30	*	Viral
1485*	>40	*	Cancer
1485*	>40	*	Heart
1485*	>40	*	Viral
1485*	>40	*	Viral
130**	30-40	*	Cancer
130**	30-40	*	Cancer
130**	30-40	*	Cancer
130**	30-40	*	Cancer

Homogeneity Attack

Zip	Age	Nationality	Disease
130**	<30	*	Heart
130**	<30	*	Heart
130**	<30	*	Viral
130**	<30	*	Viral
1485*	>40	*	Cancer
1485*	>40	*	Heart
1485*	>40	*	Viral
1485*	>40	*	Viral
130**	30-40	*	Cancer
130**	30-40	*	Cancer
130**	30-40	*	Cancer
130**	30-40	*	Cancer

- Alice's neighbor Bob is in the hospital.
- Alice knows Bob is 35 years old and is from Dryden (13053).
- Alice learns that Bob has cancer.

Alice

Background Knowledge Attack

Zip	Age	Occupation	Salary
130**	<30	*	~~Heart~~
130**	<30	*	~~Heart~~
130**	<30	*	Viral
130**	<30	*	Viral
1485*	>40	*	Cancer
1485*	>40	*	Heart
1485*	>40	*	Viral
1485*	>40	*	Viral
130**	30-40	*	Cancer
130**	30-40	*	Cancer
130**	30-40	*	Cancer
130**	30-40	*	Cancer

Alice

- Alice's friend Umeko is in the table.
- Alice knows Umeko is 24, a Japanese, living in Freeville (13068)
- Japanese have *extremely low incidence* of heart disease

Alice learns Umeko has a viral infection

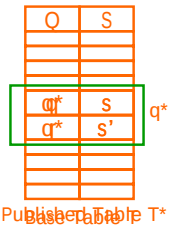
Data Publishing Desiderata

- Need to defend against attacks based on background knowledge
- Need to permit efficient sanitization algorithms
- Guarantee understood by a lay person

Talk Outline

- Introduction
- Privacy-preserving data mining
- Privacy-preserving data publishing
  - K-Anonymity
  - Attacks
  - L-Diversity
- Conclusions

Incorporating Background Knowledge



- Worst-case assumption: Adversary has *full knowledge of the joint distribution of the attributes.*
- Prior Belief:
  P[t[S] = s | t[Q] = q] = f(s|q)
- Posterior Belief:
  P[t[S] = s | t[Q] = q & t* ε T*]

$$= \frac{n_{sq^*}\frac{f(s,q)}{f(s,q^*)}}{\sum_{s'} n_{s'q^*}\frac{f(s',q)}{f(s',q^*)}}$$

Privacy Definition (1)

- Positive Disclosure: Posterior Belief > 1-δ
- Negative Disclosure: Posterior Belief < δ

BUT:

- Not all positive disclosures are bad
  - OK to disclose Bob is healthy
- Not all negative disclosures are bad
  - OK to disclose Bob does not have Ebola

Privacy Definition (2)

- Bayes-optimal privacy: After publishing we have
  Posterior belief ~ prior belief
- Example instantiation: α-to-β privacy breach definition
  Prior Belief < α and posterior Belief > β **OR**
  Prior Belief >1- α and posterior Belief <1-β
- Automatically eliminates homogeneity attack
  - Homogeneity → Posterior belief = 1

Bayes-Optimal Privacy– Drawbacks

- Insufficient knowledge
  - Nobody knows the complete joint distribution
- Adversary's knowledge unknown
  - Data publisher does not know how much the adversary knows
- Computational intractability
  - Checking for every (q,s) pair ...

Towards A Practical Definition (1)

- Posterior belief = $$\frac{n_{sq^*}\frac{f(s,q)}{f(s,q^*)}}{\sum_{s'} n_{s'q^*}\frac{f(s',q)}{f(s',q^*)}}$$
- Homegeneity attack

Q	S
q*	s
q*	s
q*	s
q*	s
q*	s'

$$\forall s' \neq s, n_{sq^*} >> n_{s'q^*}$$

Towards A Practical Definition (2)

- Posterior belief = $$\frac{n_{sq^*}\frac{f(s,q)}{f(s,q^*)}}{\sum_{s'} n_{s'q^*}\frac{f(s',q)}{f(s',q^*)}}$$
- Background knowledge attack

Q	S
q*	s
q*	~~s1~~
q*	~~s2~~
q*	~~s3~~
q*	~~s4~~

$$\forall s' \neq s, \frac{f(s',q)}{f(s',q^*)} \approx 0$$

Ensuring Diversity

- L-Diversity: Ensure that every group has at least L *well represented* groups of sensitive values"
  - "well represented" = roughly equal, non-negligible proportions

Two instantiations:

- Entropy l-diversity: Entropy(group) > log(l)

$$-\sum_{s\in S} p_{sq^*}\log(p_{sq^*}) \geq \log(\ell), \qquad p_{sq^*} = \frac{n_{sq^*}}{\sum_{s'\in S} n_{s'q^*}}$$

- Recursive (c,l)-diversity

3-Diverse Microdata

Zip	Age	Nationality	Disease
1306*	<=40	*	~~Heart~~
1306*	<=40	*	Viral
1306*	<=40	*	Cancer
1306*	<=40	*	Cancer
1485*	>40	*	Cancer
1485*	>40	*	Heart
1485*	>40	*	Viral
1485*	>40	*	Viral
1305*	<=40	*	Heart
1305*	<=40	*	Viral
1305*	<=40	*	Cancer
1305*	<=40	*	Cancer

- Bob is 35 years old and is from Dryden (13053).
- Umeko is 24, a Japanese from Freeville (13068)
- Japanese have *extremely low incidence* of heart disease

L-Diversity Revisited

- L -Diversity: Every group has at least L *well represented* groups
- Note: L-diversity does not protect against adversaries having arbitrary background knowledge.
- But: L-diversity increases the bar.

Q	S
q*	s
q*	~~s1~~
q*	~~s2~~
q*	~~s3~~
q*	~~s4~~

L-Diversity: Summary

- Defends against background knowledge attacks and homogeneity attacks
  - L-Diversity ensures diversity
  - Gives guarantees against "unknown" background knowledge
  - Can model don't care values ("person is healthy")
- Guarantee understood by a lay person
  - "At least L different values"
- Permits efficient sanitization algorithms
  - Bayes-optimal definition is not monotone
  - L-Diversity and (c,k)-recursive L-Diversity are monotone
- Experiments show that little utility is lost

Talk Outline

- Introduction
- Privacy-preserving data mining
- Privacy-preserving data publishing
- Conclusions

What I Talked About

- Privacy-preserving association rule mining
  - α-to-β privacy breaches
  - Amplification condition
  - Select-a-size randomization, itemset compression
- Privacy-preserving data publishing
  - Attacks due to background knowledge
  - L-diversity

What I Talked About (Only Useful Stuff)

The primary purpose of the DATA statement is to give names to constants; instead of referring to pi as 3.141592653589793 at every appearance, the variable PI can be given that value with a DATA statement and used instead of the longer form of the constant. This also simplifies modifying the program, should the value of pi change.

-- FORTRAN manual for Xerox Computers

Open Problems

- We only scratched the surface
- Selected future topics:
  - Tradeoff of utility versus privacy
  - Re-publication
  - Theory of learning from summaries
  - Multi-round protocols
  - Combination of randomization with other techniques (secure-multiparty computation, sketching, etc.)
  - Formalization of classes of background knowledge

Modeling Belief

- We model background knowledge (i.e. prior belief) as a background distribution plus simple existential statements about tuples (e.g. Person X is in the original table)

Knowing a Tuple-Level Pattern

- Person X is a 35 yr old Male living in 14850.
- Males do not usually have miscarriages, or ovarian or breast cancer.

Zip	Age	Sex	Malady
1485*	3*	*	Miscarriage
1485*	3*	*	Ovarian cancer
1485*	3*	*	Breast cancer
1485*	3*	*	Miscarriage
1485*	3*	*	Malaria

Knowing a Table-Level Pattern

- Person X is a 38 yr old Female living in 14850.
- Person Y is a 42 yr old Male living in 14850.
- X and Y are married.
- Viruses often attack spouses together.

Zip	Age	Sex	Malady
1485*	3*	*	Miscarriage
1485*	3*	*	Ovarian cancer
1485*	3*	*	Breast cancer
1485*	3*	*	Miscarriage
1485*	3*	*	Influenza
14850	4*	*	Malaria
14850	4*	*	Malaria
14850	4*	*	Malaria
14850	4*	*	Influenza
14850	4*	*	Malaria

Issue

- Representing prior knowledge by a distribution on tuples plus simple tuple-level existential statements is not sufficient → Distribution over tables.
- Distribution over distributions? Hidden variables? Encompassing framework?

Thanks

Current Students:
- Ashwin Machanavajjhala
- Daniel Kifer (graduating summer 2006)
- David Martin
- Muthuramakrishnan Venkitasubramaniam

Former students:
- Alexandre Evfimievski (now IBM Almaden)

Collaborators:
- Rakesh Agrawal (IBM Almaden)
- Ramakrishnan Srikant (IBM Almaden)

But Of Course We Have More Confidence Than Scott Adams …

Questions?

http://www.cs.cornell.edu/johannes
johannes@cs.cornell.edu


Cornell University

Generalization

- Originally defined by Samarati and Sweeney [Sa01, Sw02-1, Sw02-2]
- Each attribute has a domain of values
- Many-to-one (user-defined) generalization functions map the domains of each quasi-identifier attribute to successively more general domains