Sequential Patterns, Trends and Privacy

Ramakrishnan Srikant

IBM Almaden Research Center

www.almaden.ibm.com/cs/people/srikant/

Talk Overview

- Sequential Patterns & Trends
- Privacy Preserving Data Mining
- The Research Challenge

Sequential Patterns

- Given:
  - a set of data-sequences
  - data-sequence : list of transactions
  - transaction : set of items + transaction-time
- Example: 10% of customers bought "Foundation" and "Ringworld" in one transaction, followed by "Ringworld Engineers" in another transaction.
  - 10% is called the *support* of the pattern
- Find all sequential patterns supported by more than a user-specified percentage of data-sequences.
- R. Agrawal and R. Srikant, "Mining Sequential Patterns", ICDE '95.

Sequential Patterns Rules

- $\langle$ (F, R) (RE) $\Rightarrow$ (RT) $\rangle$ with 3% support and 40% confidence.
  - Confidence: 40% of occurrences of $\langle$ (F, R) (RE) $\rangle$ are followed by (RT).
- Problem Decomposition:
  - Find all sequential patterns with minimum support.
  - Use the sequential patterns to generate rules.

Applications

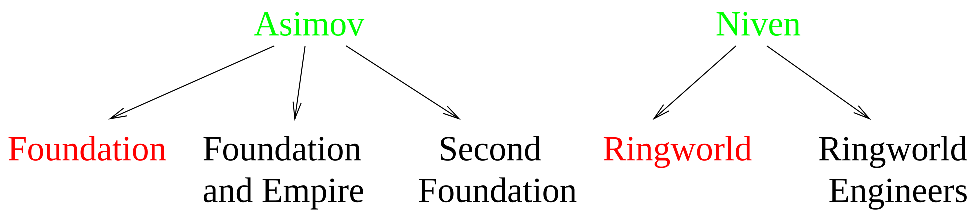
- Attached mailing, e.g., customized mailings for a book club.
- Customer satisfaction/retention
- Web log analysis
- Medical research

Generalizations

- Time Constraints:
  - Don't care if someone bought "Ringworld Engineers" 3 years after buying "Ringworld".
  - Maximum/minimum time-gap between adjacent elements.
- Flexible definition of transaction:
  - Allow all items bought within a user-specified time interval to be considered a "transaction".
  - *Sliding window* transactions.

Generalizations (cont.)

- Taxonomies:



  - find patterns between items at any level of the taxonomy
  - a data sequence ‘ “Foundation”, followed by “Ringworld” ’ would support the sequential patterns
    “Foundation”, followed by “Ringworld”,
    “Foundation”, followed by “Niven”,
    “Asimov”, followed by “Niven”, etc.

- R. Srikant and R. Agrawal, “Sequential Patterns, Generalizations & Performance Improvements”, EDBT ’96.

GSP Algorithm: Overview

- L_k : Set of frequent sequences of size k (those with minimum support).
- C_k : Set of candidate sequences of size k (potentially frequent sequences)

L_1 = {frequent items};
for ($k = 1$; $L_k \neq \emptyset$; $k{+}{+}$) **do**
  begin
  C_{k+1} = New candidates generated from L_k;
  foreach data-sequence s in the database **do**
    Increment the count of all candidates in C_{k+1} that are supported by s.
  L_{k+1} = Candidates in C_{k+1} with minimum support.
  end
Answer $= \bigcup_k L_k$;

Candidate Generation

Given a sequence $s = \langle s_1 s_2 ... s_n \rangle$ and a subsequence $c = \langle c_1 c_2 ... c_m \rangle$, c is a *contiguous* subsequence of s if there exists an integer k such that $c_i \subseteq s_{i+k}, 1 \leq i \leq m$.

Example: Let $s = \langle$ (1, 2) (3, 4) (5) (6) $\rangle$.
Contiguous subsequence: $\langle$ (3) (5) $\rangle$.
Non-contiguous: $\langle$ (3, 4) (6) $\rangle$

Lemma: If a data-sequence d supports a sequence s, d will also support any contiguous subsequence of s. If there is no max-gap constraint, d will support any subsequences of s.

d: $\langle$ (11) (1 2 15) (17) (3) (4 12) $\rangle$

s: $\langle$ (1 2) (3) (4) $\rangle$

c: $\langle$ (2) (3) $\rangle$

What about $\langle$ (2) (4) $\rangle$?

All *contiguous* subsequences of a frequent subsequence are frequent.

Candidate Generation (cont.)

Join Phase:

s_1': result of dropping the first item of s_1
s_2': result of dropping the last item of s_2

Join condition: s_1 joins with s_2 if $s'_1 = s'_2$
Result: s_1 extended with the last item in s_2

L_3		C_4
⟨ (1, 2) (3) ⟩	s_1' = ⟨ (2) (3) ⟩	
⟨ (1, 2) (4) ⟩		
⟨ (1) (3, 4) ⟩		
⟨ (2) (3, 4) ⟩	s_2' = ⟨ (2) (3) ⟩	⟨ (1, 2) (3, 4) ⟩
⟨ (2) (3) (5) ⟩	s_2' = ⟨ (2) (3) ⟩	⟨ (1, 2) (3) (5) ⟩

Prune Phase: Drop all sequences that have a non-frequent contiguous subsequence.

⟨ (1,2) (3) (5) ⟩ is dropped since ⟨ (1) (3) (5) ⟩ is not in L_3.

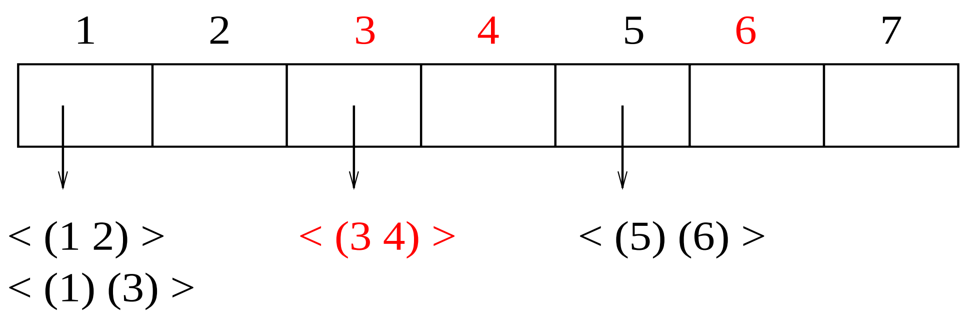
Counting Support

Given

- a data-sequence T and
- a set of candidates C_k,

find all members of C_k which are supported by T.

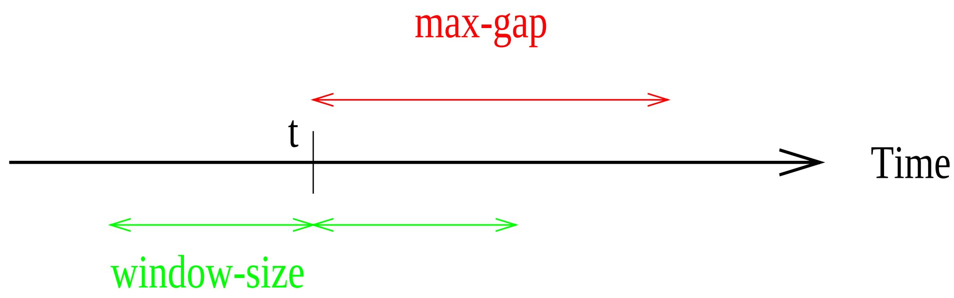
C_2 : $\{\ \langle$ (1 2) $\rangle$, $\langle$ (1) (3) $\rangle$, $\langle$ (3 4) $\rangle$, $\langle$ (5) (6) $\rangle\ \}$
T : $\langle$ (3 4) (6) $\rangle$



- Only check candidates in buckets corresponding to 3, 4, and 6.
- avg. number of items in data-sequence $\ll$ total number of items
- generalized into a *hash-tree*

Counting Support (cont.)

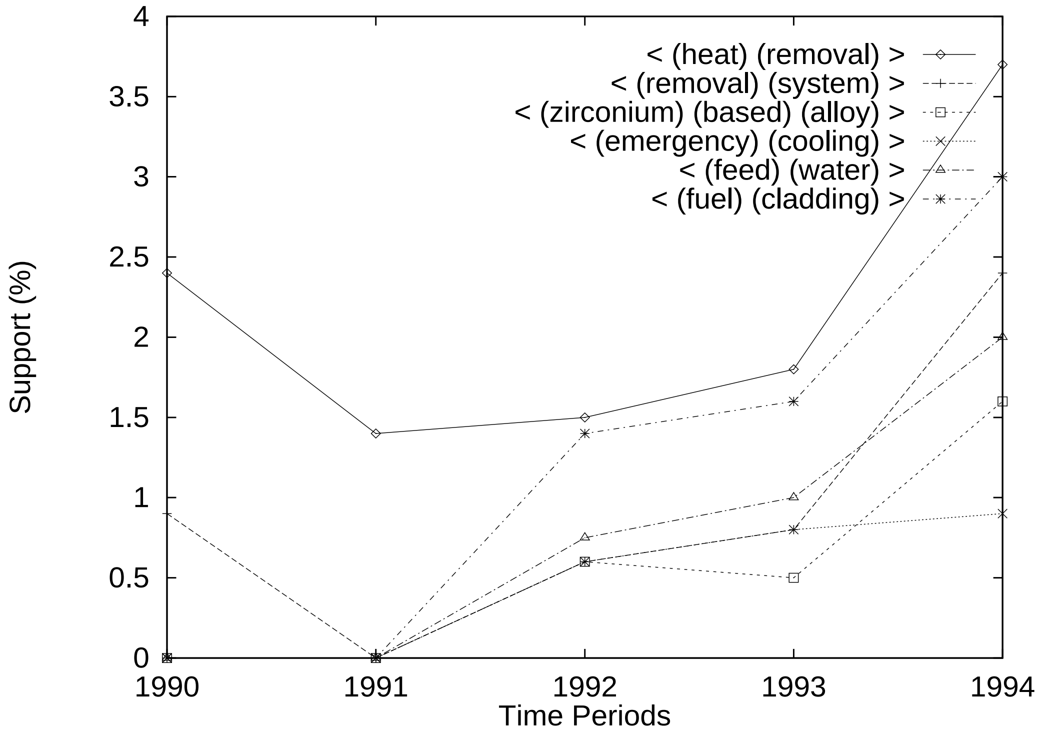
If we reach a node by hashing on an item x whose transaction-time is t, only check items in the data-sequence whose time is in $[t - \text{window-size}, t + \max(\text{window-size}, \text{max-gap})]$



Discovering Trends in Text Databases

- Identify frequent phrases using sequential patterns.
  - Sequential patterns allow considerable latitude in definition of "phrase".
- Generate histories of phrases.
  - Partition data by time period, e.g., years.
  - Find support in each time period.
- Identify phrases that satisfy a specified trend.
  - SDL Query language (Agrawal et al., VLDB '95)
  - GUI to generate queries.
- B. Lent, R. Agrawal and R. Srikant, "Discovering Trends in Text Databases", KDD '97.

Upward Trends in Patent Data



Talk Overview

- Sequential Patterns & Trends
- Privacy Preserving Data Mining
- The Research Challenge

Growing Concern for Privacy of Digital Information

- Popular Press:
  - Economist: The End of Privacy (May 99)
  - Time: How to Protect Your Privacy Online (July 2001)
- Govt directives/commissions:
  - European directive on privacy protection (Oct 98)
  - Information and privacy commissioner, Ontario (Jan 98)
- Special issue on internet privacy, CACM, Feb 99
- S. Garfinkel, "Database Nation: The Death of Privacy in 21st Century", O' Reilly, Jan 2000

Privacy Surveys

- [CRA99b] survey of web users:
  - 17% privacy fundamentalists
  - 56% pragmatic majority
  - 27% marginally concerned
- [Wes99] survey of web users:
  - 82% : privacy policy matters
  - 14% don't care
- Not equally protective of every field
  - may not divulge at all certain fields;
  - may not mind giving true values of certain fields;
  - may be willing to give not true values but modified values of certain fields.

Technical Question

- The primary task in data mining: development of models about aggregated data.
- Can we develop accurate models without access to precise information in individual data records?

- R. Agrawal and R. Srikant, “Privacy Preserving Data Mining”, SIGMOD 2000.

Talk Overview

- Sequential Patterns & Trends
- Privacy Preserving Data Mining
  - Randomization protects information at the individual level.
  - Algorithm to reconstruct the distribution of values.
  - Use reconstructed distributions in data mining algorithms, e.g. to build decision-tree classifier.
  - How well does it work?
- The Research Challenge

Using Randomization to protect Privacy

- Return $x_i + r$ instead of x_i, where r is a random value drawn from a distribution.
  - Uniform
  - Gaussian
- Fixed perturbation – not possible to improve estimates by repeating queries.
- Algorithm knows parameters of r's distribution.

Reconstruction Problem

- Original values $x_1, x_2, \ldots, x_n$
  - realizations of iid random variables $X_1, X_2, \ldots, X_n$,
  - each with the same distribution as random variable X.
- To hide these values, we use $y_1, y_2, \ldots, y_n$
  - realizations of iid random variables $Y_1, Y_2, \ldots, Y_n$,
  - each with the same distribution as random variable Y.

Given

- $x_1+y_1, x_2+y_2, \ldots, x_n+y_n$
- the density function f_Y for Y,

estimate the density function f_X for X.

Using Bayes' Rule

- Assume we know both f_X and f_Y.
- Let $w_i \equiv x_i + y_i$.

$$
\begin{aligned}
& f_{X_1}(a \mid X_1 + Y_1 = w_1) \\
&= \frac{f_{X_1+Y_1}(w_1 \mid X_1 = a)\, f_{X_1}(a)}{f_{X_1+Y_1}(w_1)} \\
&\quad \text{(using Bayes' rule for density functions)} \\
&= \frac{f_{X_1+Y_1}(w_1 \mid X_1 = a)\, f_{X_1}(a)}{\int_{-\infty}^{\infty} f_{X_1+Y_1}(w_1 \mid X_1 = z)\, f_{X_1}(z)\, dz} \\
&= \frac{f_{Y_1}(w_1 - a)\, f_{X_1}(a)}{\int_{-\infty}^{\infty} f_{Y_1}(w_1 - z)\, f_{X_1}(z)\, dz} \qquad (Y_1 \text{ independent of } X_1) \\
&= \frac{f_Y(w_1 - a)\, f_X(a)}{\int_{-\infty}^{\infty} f_Y(w_1 - z)\, f_X(z)\, dz} \qquad (f_{X_1} \equiv f_X,\ f_{Y_1} \equiv f_Y)
\end{aligned}
$$

$$
\begin{aligned}
f'_X(a) &\approx \frac{1}{n} \sum_{i=1}^{n} f_{X_i}(a \mid X_i + Y_i = w_i) \\
&= \frac{1}{n} \sum_{i=1}^{n} \frac{f_Y(w_i - a)\, f_X(a)}{\int_{-\infty}^{\infty} f_Y(w_i - z)\, f_X(z)\, dz}
\end{aligned}
$$

Reconstruction Method: Algorithm

f_X^0 := Uniform distribution
$j := 0$ // Iteration number
`repeat`
    Use equation to compute a new estimate f_X^{j+1}.
    $j := j + 1$
`until` (stopping criterion met)

Stopping Criterion: Stop when difference between successive estimates of the original distribution becomes very small (1% of the threshold of the χ^2 test).

Using Partitioning to Speed Computation

- distance(z, w_i) $\approx$ distance between the mid-points of the intervals in which they lie, and
- density function $f_X(a)$ $\approx$ the average of the density function over the interval in which a lies.

$$f'_X(a) = \frac{1}{n} \sum_{i=1}^{n} \frac{f_Y(w_i - a)\, f_X(a)}{\int_{-\infty}^{\infty} f_Y(w_i - z)\, f_X(z)\, dz}$$

becomes

$$\Pr{}'(X \in I_p) =$$

$$\frac{1}{n} \sum_{s=1}^{k} N(I_s) \times \frac{f_Y(m(I_s) - m(I_p)) \Pr(X \in I_p)}{\sum_{t=1}^{k} f_Y(m(I_s) - m(I_t)) \Pr(X \in I_t)}$$

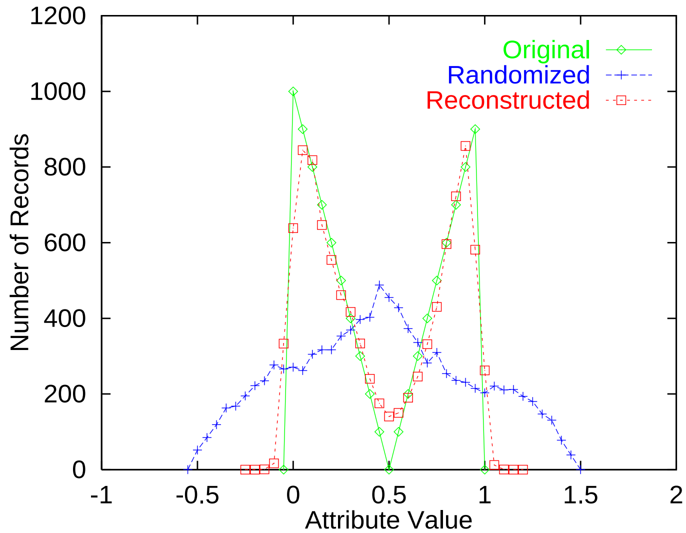
- Can be computed in O(k^2) time, where k is the number of intervals.

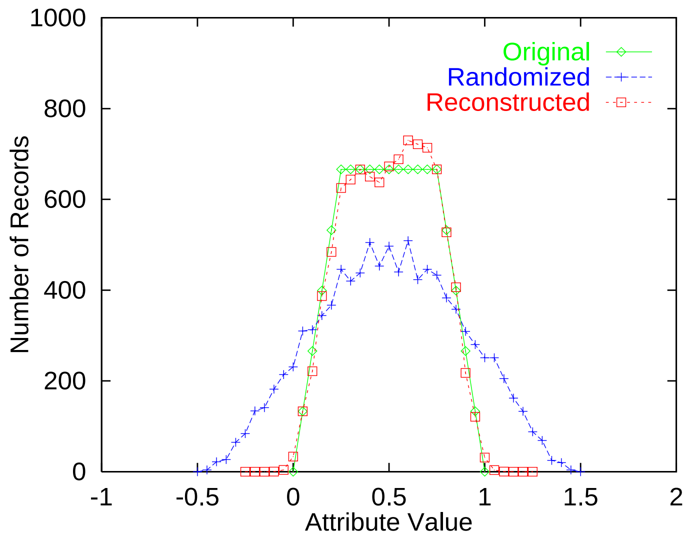
Maximum Likelihood Estimate

- The above algorithm (minus the interval approximation) converges to the maximum likelihood estimate.
  - D. Agrawal and C.C. Aggarwal, "On the Design and Quantification of Privacy Preserving Data Mining Algorithms", PODS 2001.

How well does this work?

- Uniform random variable [-0.5, 0.5]





Talk Overview

- Sequential Patterns & Trends
- Privacy Preserving Data Mining
  - Randomization protects information at the individual level.
  - Algorithm to reconstruct the distribution of values.
  - Use reconstructed distributions to build decision-tree classifier.
  - How well does it work?
- The Research Challenge

Algorithms

Global:

- Reconstruct for each attribute once at the beginning.
- Induce decision tree using reconstructed data.

ByClass:

- For each attribute, first split by class, then reconstruct separately for each class.
- Induce decision tree using reconstructed data.

Local:

- As in ByClass, split by class and reconstruct separately for each class.
- However, reconstruct at each node (not just once).

Methodology

- Compare accuracy of Global, ByClass and Local against
  - Original: unperturbed data without randomization.
  - Randomized: perturbed data but without making any corrections for randomization.
- Synthetic data generator from [AGI+92].
- Training set of 100,000 records, split equally between the two classes.

Quantifying Privacy

If it can be estimated with $c\%$ confidence that a value x lies in the interval $[x_1,\ x_2]$, then the interval width $(x_2 - x_1)$ defines the amount of privacy at $c\%$ confidence level.

- Example: Randomization Level for Age[10,90]
  - Given a perturbed value 40
  - 95% confidence that true value lies in [30,50]
  - $\frac{\text{Interval Width : 20}}{\text{Range : 80}} \Rightarrow$ 25% randomization level

- Uniform: between $[-\alpha,\ +\alpha]$
- Gaussian: mean $\mu = 0$ and standard deviation σ

	Confidence		
	50%	95%	99.9%
Uniform	$0.5 \times 2\alpha$	$0.95 \times 2\alpha$	$0.999 \times 2\alpha$
Gaussian	$1.34 \times \sigma$	$3.92 \times \sigma$	$6.8 \times \sigma$

Synthetic Data Functions

- Class A if function is true, Class B otherwise.

F1 $(\text{age} < 40) \vee ((60 \leq \text{age})$

F2 $((\text{age} < 40) \wedge (50K \leq \texttt{salary} \leq 100K)) \vee$
$((40 \leq \text{age} < 60) \wedge (75K \leq \texttt{salary} \geq 125K)) \vee$
$((\text{age} \geq 60) \wedge (25K \leq \texttt{salary} \leq 75K))$

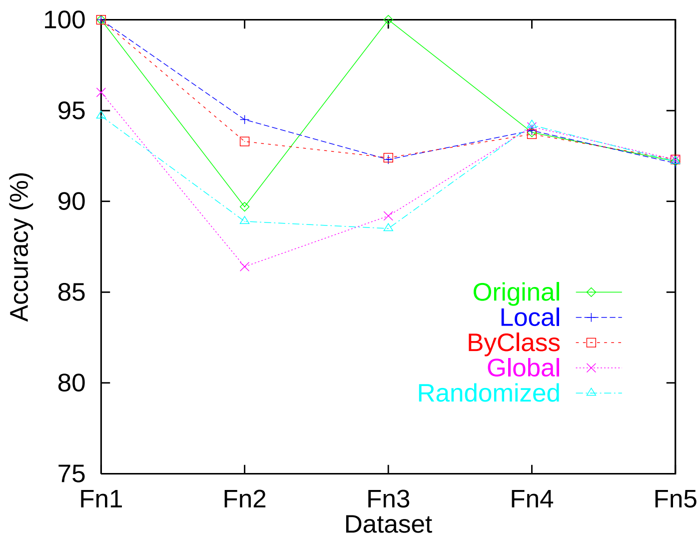
F3 $((\text{age} < 40) \wedge$
$(((\texttt{elevel} \in [0..1]) \wedge (25K \leq \texttt{salary} \leq 75K)) \vee$
$((\texttt{elevel} \in [2..3]) \wedge (50K \leq \texttt{salary} \leq 100K)))) \vee$
$((40 \leq \text{age} < 60) \wedge$
$(((\texttt{elevel} \in [1..3]) \wedge (50K \leq \texttt{salary} \leq 100K)) \vee$
$(((\texttt{elevel} = 4)) \wedge (75K \leq \texttt{salary} \leq 125K)))) \vee$
$((\text{age} \geq 60) \wedge$
$(((\texttt{elevel} \in [2..4]) \wedge (50K \leq \texttt{salary} \leq 100K)) \vee$
$((\texttt{elevel} = 1)) \wedge (25K \leq \texttt{salary} \leq 75K))))$

F4 $(0.67 \times (\texttt{salary} + \texttt{commission}) - 0.2 \times \texttt{loan} - 10K) > 0$

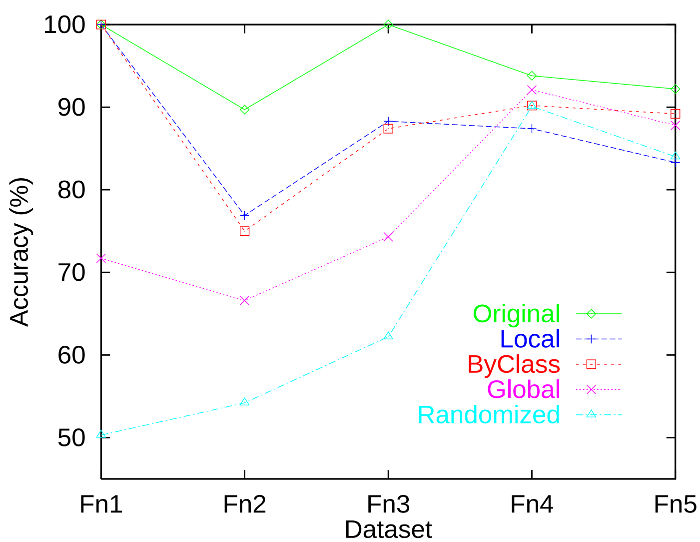
F5 $(0.67 \times (\texttt{salary} + \texttt{commission}) - 0.2 \times \texttt{loan} +$
$0.2 \times \text{equity} - 10K) > 0$
where $\text{equity} = 0.1 \times \texttt{hvalue} \times \max(\texttt{hyears} - 20, 0)$

Classification Accuracy

Randomization Level: 25% of Attribute Range

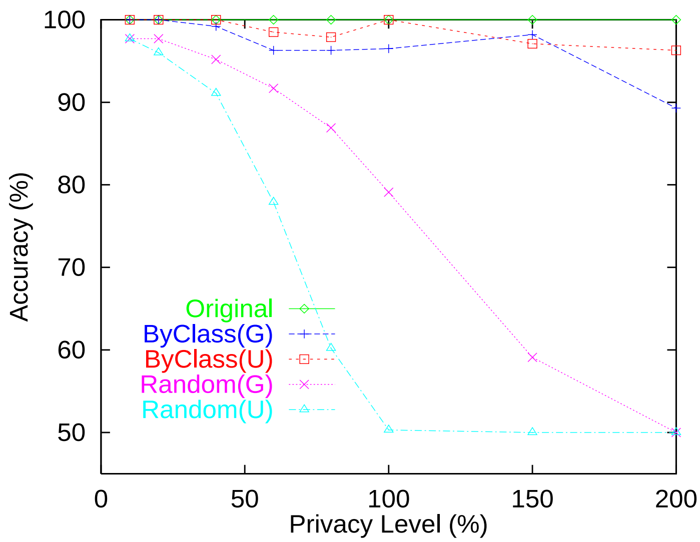


Randomization Level: 100% of Attribute Range

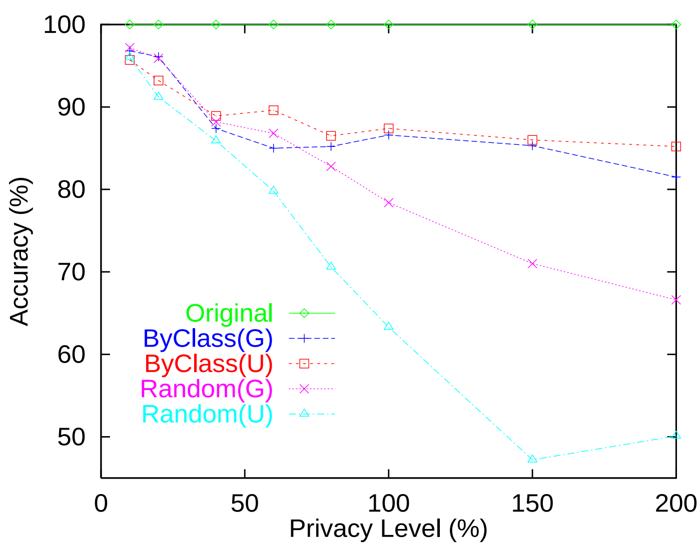


Change in Accuracy with Privacy

Fn 1



Fn 3



Potential Privacy Breaches

- Distribution is a spike.
  - Example: Everyone is of age 40.
- Some randomized values are only possible from a given range.
  - Example: Add U[-50,+50] to age and get 125 $\Rightarrow$ True age is $\geq$ 75.
  - Not an issue with Gaussian.

Potential Privacy Breaches (cont.)

- Most randomized values in a given interval come from a given interval.
  - Example: 60% of the people whose randomized value is in [120,130] have their true age in [70,80].
  - Implication: Higher levels of randomization will be required.
- Correlations can make previous effect worse.
  - Example: 80% of the people whose randomized value of age is in [120,130] and whose randomized value of income is [...] have their true age in [70,80].
- Given a dataset, we can search for privacy breaches.
  - But how do we do it in advance?

Cryptographic Approach

- Y. Lindell and B. Pinkas, "Privacy Preserving Data Mining", *Crypto 2000*, August 2000.
- Problem: Two parties owning confidential databases wish to build a decision-tree classifier on the union of their databases, without revealing any unnecessary information.
- Malicious adversary: can alter its input, e.g., define input to be the empty database.
- Semi-honest (or passive) adversary: Correctly follows the protocol specification, yet attempts to learn additional information by analyzing the messages.

Private Distributed ID3

- Key problem: find attribute with highest information gain.
- We can then split on this attribute and recurse.
- Information Gain: Need to compute
  - $\sum_j \sum_i |T(a_j, c_i)| \log |T(a_j, c_i)|$
  - $\sum_j |T(a_j)| \log |T(a_j)|$.
  - $T(c_i, a_j) =$ set of records in class c_i with attribute $A = a_j$.
- Given v_1 known to party 1 and v_2 known to party 2, compute $(v_1 + v_2) \log(v_1 + v_2)$ and output random shares.
- Given random shares for each attribute, use Yao's protocol to compute information gain.

Cryptographic Approach (Summary)

- Solves different problem (vs. randomization)
- Efficient with semi-honest adversary and small number of parties.
- Gives (almost) the same solution as the non-privacy-preserving computation (unlike randomization).
- Will not scale to individual user data.

Talk Overview

- Sequential Patterns & Trends
- Privacy Preserving Data Mining
- The Research Challenge

Randomizing Time Values

- Similar to randomizing age or salary.
- But what if we want to find trends at different levels of granularity?
  - People who visit the website on a Saturday ...
  - People who visit the website in March ...

Randomizing a Boolean Attribute

- Warner, "Randomized response: A survey technique for eliminating evasive answer bias", J. Am. Stat. Assoc. 1965.
- Boolean variables, e.g., "drug addiction = yes/no".
- Keep the value with probability p, and flip it with probability $1 - p$.
- Let f_y be fraction of records with true "yes", f'_y fraction of records with "yes" after randomization:

  $f'_y = f_y p + (1 - f_y)(1 - p)$

  $f_y = (f'_y - (1 - p))/(2p - 1)$

Randomizing Transaction Data

- For each (unique) item in the transaction, keep item with probability p and replace item with a random item with probability $1 - p$.

- Can (probably) compute formulae for support and variance.

Privacy Breaches with Sequential Patterns

- Replace item with 80% probability.
- 10 million transactions, $\langle$(F, R) (RE)$\rangle$ has 1% support.
- Prob. of retaining pattern $= 0.2^3 =$ 0.8%
- 805 occurrences of $\langle$(F, R) (RE)$\rangle$ in randomized data.
  - 800 of these were in the original data-sequence.
  - 5 of these were generated from replaced items.
- Estimate with 99% confidence that pattern was originally present!
- Ack: Alexandre Evfimievski

The Research Challenge

- Goal: Have your cake and mine it too!
  - Preserve privacy at the individual level, but still build accurate models.
- Can we discover sequential patterns and trends, while avoiding privacy breaches?

Related Work: Statistical Databases

- Statistical Databases : provide statistical information without compromising sensitive information about individuals (surveys: [AW89] [Sho82])
- Query Restriction
  - restrict the size of query result (e.g. [FEL72][DDS79])
  - control overlap among successive queries (e.g. [DJL79])
  - keep audit trail of all answered queries (e.g. [CO82])
  - suppress small data cells (e.g. [Cox80])
  - cluster entities into mutually exclusive atomic populations (e.g. [YC77])
- Data Perturbation
  - replace the original database by a sample from the same distribution (e.g. [LST83][LCL85][Rei84])
  - sample the result of a query (e.g. [Den80])
  - swap values between records (e.g. [Den82])
  - add noise to the query result (e.g. [Bec80])
  - add noise to the values (e.g. [TYW84][War65])

Related Work: Statistical Databases (cont.)

- Negative results: cannot give high quality statistics and simultaneously prevent partial disclosure of individual information [AW89]

- Negative results not directly applicable to privacy-preserving data mining.
  - Also want to prevent disclosure of confidential information
  - But sufficient to reconstruct original distribution of data values, i.e. not interested in high quality point estimates