Data Mining: A Powerful Tool for Data Cleaning

Jiawei Han
Department of Computer Science
University of Illinois at Urbana-Champaign

Nov. 4, 2003
Outline

- Data mining: A powerful tool for data cleaning
  - How can newer data mining methods help data quality assurance?
- PROM (Profile-based Object Matching): Identifying and merging objects by profile-based data analysis
- CoMine: Comparative correlation measure analysis
- CrossMine: Mining noisy data across multiple relations
- SecureClass: Effective document classification in the presence of substantial amount of noise
- Conclusions
Correlation, classification and cluster analysis for data cleaning
- Discovery of interesting data characteristics, models, outliers, etc.
- Mining database structures from contaminated, heterogeneous databases

A comprehensive overview on the theme

How can newer data mining methods help data quality assurance?
- Exploring several newer data mining tasks and their relationships to data cleaning
Where Are the Source of the Materials?

- Y.-K. Lee, W.-Y. Kim, Y. D. Cai, and J. Han, CoMine: Efficient mining of correlated patterns, Proc. 2003 Int. Conf. on Data Mining (ICDM'03), Melbourne, FL, Nov. 2003.
- X. Yin, J. Han, J. Yang, and P.S. Yu, CrossMine: Efficient classification across multiple database relations, Proc. 2004 Int. Conf. on Data Engineering, Boston, MA, March 2004
- X. Yin, J. Han, A. Mehta, SecureClass: Privacy-Preserving Classification of Text Documents, submitted for publication.
Object Matching for Data Cleaning

Object matching: Identifying and merging objects by data mining and statistical analysis

Decide if two objects refer to the same real-world entity

(Mike Smith, 235-2143) & (M. Smith, 217 235-2143)

Purposes: information integration & data cleaning

remove duplicates when merging data sources

consolidate information about entities

information extraction from text

join of string attributes in databases
PROM: Profile-based Object Matching

Key observations
- disjoint attributes are often correlated
- such correlations can be exploited to perform “sanity check”

Example
- (9, Mike Smith) & (Mike Smith, 200K)
- Match them?— because both names are “Mike Smith”?
- Sanity check using profiler:
  - Match? → Mike Smith: 9 years-old with salary 200K
  - Knowledge: the profile of a typical person
  - Conflict with the profile → two are unlikely to match
Example: Matching Movies

Step 1: check if two movie names are sufficiently similar
Step 2: sanity check using multiple profilers

- **review profiler:**
  - Production year (pyear) must not be after review year (ryear)
  - Roger Ebert (reviewer) never reviews movies with rating < 5

- **actor profiler:**
  - Certain actor has never played in action movies

- **movie profiler:**
  - Rating and rrating tend to be strongly correlated

**PROM combines profiler predictions to reach matching decision**
Profilers in Movie Example

- Contain knowledge about domain concepts
  - movies, reviews, actors, studios, etc.
- Constructed once, reused anywhere
  - as long as the new matching task involves same domain concepts
- Can be constructed in many ways
  - manually specified by experts and users
  - learned from data in the domain
    - all movies at Internet Movie Database imdb.com
    - text of reviews from the New York Times
  - learned from training data of a specific matching task
    - then transferred to related matching tasks
Architecture of PROM

Training data → Expert Knowledge → Domain Data → Previous Matching Tasks

Table T1 → Hard Profiler 1 → ... → Hard Profiler n → Soft Profiler 1 → ... → Soft Profiler m → Combiner → Matching Prediction

Table T2 → Similarity Estimator → Match Filter
Hard vs. Soft Profilers: Hard Profiler

Given a tuple pair

A profiler issues a confidence score on how well the pair fits the concept (i.e., how well their data mesh together)

Hard profiler

specifies constraints that any concept instance must satisfy

- review year ≥ production year of movie
- actor A has only played in action movies

can be constructed manually by domain experts and users

can be constructed from domain data if data is complete

- e.g., by examining all movies of actor A
Hard vs. Soft Profilers: Soft Profiler

Soft profiler

- Specifies “soft” constraints that most instances satisfy
- can be constructed manually, from domain data
  (e.g., learning a Bayesian network from imdb.com)
- from training data of a matching task
  (e.g., learning a classifier from training data)
Combining Profilers

Step 1: How to combine hard profilers?
- Any hard profiler says “no match”, declare “no match”

Step 2: How to combine soft profilers?
- Each soft profiler examines pair and issue a prediction “match” with a confidence score
- Combine profilers’ scores
  - currently use weighted sum (with weights set manually)
Empirical Evaluation: CiteSeer Name Match

CiteSeer: Popularly cited authors but may not match the correct homepages

Citation list: Highly cited researchers and their homepages
- The “Jim Gray” citeseer problem: cs.vt.edu/~gray, data.com/~jgray, microsoft.com/~gray
- Which homepage should be for the real J. Gray?

Created two data sources
- source 1: highly cited researchers, 200 tuples
  - (name, highly-cited)
- source 2: homepages, 254 tuples (manually created from text)
  - (name, title, institute, graduation-year, … )
## PROM Improves Matching Accuracy

<table>
<thead>
<tr>
<th></th>
<th>Baseline</th>
<th>PROM</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DT</td>
<td>Man+DT</td>
<td>Man+AR</td>
<td>Man+AR+DT</td>
</tr>
<tr>
<td>CiteSeer</td>
<td>Recall</td>
<td>0.99</td>
<td>0.95</td>
<td>0.96</td>
</tr>
<tr>
<td></td>
<td>Precision</td>
<td>0.67</td>
<td>0.78</td>
<td>0.82</td>
</tr>
<tr>
<td></td>
<td>F-Value</td>
<td>0.80</td>
<td>0.85</td>
<td>0.88</td>
</tr>
</tbody>
</table>

- **Baseline**: exploit only shared attributes
- **PROM**:
  - Used three soft profilers: DT (decision tree), Man (manual), and AR (association rules)
  - Adding profilers tends to improve accuracy
  - DT < Man+AR < Man+AR+DT
CoMine: Mining Strongly Correlated Patterns

Why CoMine is closely related to data cleaning?
- Correlation analysis: A powerful data cleaning tool
- Current association analysis: generate too many rules!
- Maybe the correlation rules are what we want

What should be a good correlation measure to handle large data sets?
- Find good correlation measure
- Find an efficient mining method
Why Mining Correlated Patterns?

- Association ≠ correlation
  - high min_support → commonsense knowledge
  - low minimum support → huge number of rules

- Association may not carry the right semantics
  - “Buy walnuts ⇒ buy milk [1%, 80%]” is misleading
    - if 85% of customers buy milk

- What should be a good measure?
  - Support and conf. alone are no good
  - Will lift or $\chi^2$ be better?
## A Comparative Analysis of 21 Interesting Measures

<table>
<thead>
<tr>
<th>symbol</th>
<th>measure</th>
<th>range</th>
<th>formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi$</td>
<td>$\phi$-coefficient</td>
<td>$-1 \ldots 1$</td>
<td>$\frac{P(A \cap B) - P(A)P(B)}{\sqrt{P(A)P(B)(1-P(A))(1-P(B))}}$</td>
</tr>
<tr>
<td>$Q$</td>
<td>Yule’s $Q$</td>
<td>$-1 \ldots 1$</td>
<td>$\frac{P(A \cap B)P(\overline{A})P(B) - P(A)P(\overline{B})P(B)}{P(A)P(B)P(\overline{A})P(B)}$</td>
</tr>
<tr>
<td>$Y$</td>
<td>Yule’s $Y$</td>
<td>$-1 \ldots 1$</td>
<td>$\sqrt{P(A,B)P(\overline{A},B) + P(A,B)P(\overline{A},B)} - \sqrt{P(A,B)P(\overline{A},B)P(\overline{A},B)}$</td>
</tr>
<tr>
<td>$k$</td>
<td>Cohen’s</td>
<td>$-1 \ldots 1$</td>
<td>$\frac{P(A,B) - P(A)P(B)}{1-P(A)P(B) - P(A)P(B)}$</td>
</tr>
<tr>
<td>$PS$</td>
<td>Piatetsky-Shapiro’s</td>
<td>$-0.25 \ldots 0.25$</td>
<td>$\max(\frac{P(B</td>
</tr>
<tr>
<td>$F$</td>
<td>Certainty factor</td>
<td>$-1 \ldots 1$</td>
<td>$\max(\frac{P(B</td>
</tr>
<tr>
<td>$AV$</td>
<td>added value</td>
<td>$-0.5 \ldots 1$</td>
<td>$\sqrt{P(A,B) \max(P(B</td>
</tr>
<tr>
<td>$K$</td>
<td>Klosgen’s $Q$</td>
<td>$-0.33 \ldots 0.38$</td>
<td>$\frac{\sum_j \frac{\max_k P(A_j</td>
</tr>
<tr>
<td>$g$</td>
<td>Goodman-kruskal’s</td>
<td>$0 \ldots 1$</td>
<td>$\frac{\min(\sum_i P(A_i</td>
</tr>
<tr>
<td>$M$</td>
<td>Mutual Information</td>
<td>$0 \ldots 1$</td>
<td>$\max(P(A,B) \log \frac{P(A</td>
</tr>
<tr>
<td>$J$</td>
<td>J-Measure</td>
<td>$0 \ldots 1$</td>
<td>$\max(P(A,B) \log \frac{P(A</td>
</tr>
<tr>
<td>$G$</td>
<td>Gini index</td>
<td>$0 \ldots 1$</td>
<td>$\frac{\max(P(A</td>
</tr>
<tr>
<td>$s$</td>
<td>support</td>
<td>$0 \ldots 1$</td>
<td>$\max(P(B</td>
</tr>
<tr>
<td>$c$</td>
<td>confidence</td>
<td>$0 \ldots 1$</td>
<td>$\max(P(B</td>
</tr>
<tr>
<td>$L$</td>
<td>Laplace</td>
<td>$0 \ldots 1$</td>
<td>$\max(\frac{NP(A,B)+1}{NP(A)+2}, \frac{NP(A,B)+1}{NP(B)+2})$</td>
</tr>
<tr>
<td>$IS$</td>
<td>Cosine</td>
<td>$0 \ldots 1$</td>
<td>$\frac{\sqrt{P(A)P(B)} - P(A,B)}{P(A,B)P(A,B)}\frac{P(\overline{A})P(B)}{P(\overline{A})P(\overline{A})}$</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>coherence (Jaccard)</td>
<td>$0 \ldots 1$</td>
<td>$\frac{P(A</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>all_confidence</td>
<td>$0 \ldots 1$</td>
<td>$\frac{\max(P(\overline{A}), P(B))}{P(\overline{A})P(B)}\frac{P(\overline{A})P(B)}{P(\overline{A})P(\overline{A})}$</td>
</tr>
<tr>
<td>$o$</td>
<td>odds ratio</td>
<td>$0 \ldots \infty$</td>
<td>$\frac{\max(P(\overline{A}), P(\overline{B}))}{P(\overline{A})P(\overline{B})}$</td>
</tr>
<tr>
<td>$V$</td>
<td>Conviction</td>
<td>$0.5 \ldots \infty$</td>
<td>$\max(\frac{P(A)P(B)}{P(AB)}, \frac{P(B)P(\overline{A})}{P(B)\overline{A}})$</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>lift</td>
<td>$0 \ldots \infty$</td>
<td>$\frac{P(A,B)P(B)}{P(A)P(B)}\frac{P(A,B)P(B)}{P(A)P(B)}\frac{1-P(A)P(B)-P(\overline{A})P(B)}{1-P(A,B)-P(AB)}$</td>
</tr>
<tr>
<td>$S$</td>
<td>Collective strength</td>
<td>$0 \ldots \infty$</td>
<td>$\frac{\sum_i \frac{(P(A_i)-E_i)^2}{E_i}}{\sum_i \frac{(P(A_i)-E_i)^2}{E_i}}$</td>
</tr>
<tr>
<td>$\chi^2$</td>
<td>$\chi^2$</td>
<td>$0 \ldots \infty$</td>
<td>$\sum_i \frac{(P(A_i)-E_i)^2}{E_i}$</td>
</tr>
</tbody>
</table>
Let’s Look Closely on a few Measures

\[
\lambda = \text{lift} = \frac{P(A \cup B)}{P(A)P(B)}
\]

\[
\chi^2 = \sum \frac{(\text{Observed} - \text{Expected})^2}{\text{Expected}}
\]

\[
\alpha = \text{all} \_\text{conf} = \frac{\text{sup}(X)}{\text{max}_\text{item}\_\text{sup}(X)}
\]

\[
\gamma (\text{Jaccard}_\text{Coeff}) = \text{coh} = \frac{\text{sup}(X)}{|\text{universe}(X)|}
\]
Comparison among $\lambda$, $\alpha$, $\gamma$, and $\chi^2$

The contingency table and the behavior of a few measures

<table>
<thead>
<tr>
<th>DB</th>
<th>mc</th>
<th>¬mc</th>
<th>m¬c</th>
<th>¬(mc)</th>
<th>$\lambda$</th>
<th>$\alpha$</th>
<th>$\gamma$</th>
<th>$\chi^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>1000</td>
<td>100</td>
<td>100</td>
<td>1000</td>
<td>83.64</td>
<td>0.91</td>
<td>0.83</td>
<td>83452</td>
</tr>
<tr>
<td>A2</td>
<td>1000</td>
<td>100</td>
<td>100</td>
<td>10000</td>
<td>9.26</td>
<td>0.91</td>
<td>0.83</td>
<td>9055</td>
</tr>
<tr>
<td>A3</td>
<td>1000</td>
<td>100</td>
<td>100</td>
<td>100000</td>
<td>1.82</td>
<td>0.91</td>
<td>0.83</td>
<td>1472</td>
</tr>
<tr>
<td>A4</td>
<td>100</td>
<td>1000</td>
<td>1000</td>
<td>100000</td>
<td>8.44</td>
<td>0.09</td>
<td>0.05</td>
<td>670</td>
</tr>
<tr>
<td>A5</td>
<td>1000</td>
<td>100</td>
<td>10000</td>
<td>100000</td>
<td>9.18</td>
<td>0.09</td>
<td>0.09</td>
<td>8172</td>
</tr>
<tr>
<td>A6</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1</td>
<td>0.5</td>
<td>0.33</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>milk</th>
<th>¬milk</th>
<th>coffee</th>
<th>mc</th>
<th>¬mc</th>
<th>¬(coffee)</th>
<th>m¬c</th>
<th>¬(mc)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
What Should Be a Good Correlation Measure?

- Disclose genuine correlation relationship
- Null Invariance Property (Tan, et al. 02)
  - Invariant by adding more null transactions (those not containing these items)
  - Useful in large sparse databases – co-presence is far less than co-absence
- Has the downward closure property
  - for efficient mining (Apriori like algorithms)
## Examining a larger set of Measures

<table>
<thead>
<tr>
<th>φ</th>
<th>φ-coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q</td>
<td>Yule’s Q</td>
</tr>
<tr>
<td>Y</td>
<td>Yule’s Y</td>
</tr>
<tr>
<td>k</td>
<td>Cohen’s</td>
</tr>
<tr>
<td>P</td>
<td>Piatetsky-Shapiro’s</td>
</tr>
<tr>
<td>S</td>
<td>Certainty factor</td>
</tr>
<tr>
<td>A</td>
<td>Added value</td>
</tr>
<tr>
<td>V</td>
<td>Klosgen’s Q</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>g</th>
<th>Goodman-kruskal’s</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>Mutual Information</td>
</tr>
<tr>
<td>J</td>
<td>J-Measure</td>
</tr>
<tr>
<td>G</td>
<td>Gini index</td>
</tr>
<tr>
<td>s</td>
<td>support</td>
</tr>
<tr>
<td>c</td>
<td>confidence</td>
</tr>
<tr>
<td>L</td>
<td>Laplace</td>
</tr>
<tr>
<td>IS</td>
<td>Cosine</td>
</tr>
<tr>
<td>γ</td>
<td>Coherence(Jaccard)</td>
</tr>
<tr>
<td>α</td>
<td>All_confidence</td>
</tr>
</tbody>
</table>

- o: odds ratio  \( \text{range from -1 to 1} \)
- V: Conviction  \( \text{range from 0 to 1} \)
- λ: lift  \( \text{range from 0 to } \infty \)
- S: Collective Strength
- \( \chi^2 \): Chi-squared
Effect of Null Transactions: Positively Correlated Cases

- Input parameters (symmetric data)
- Results

<table>
<thead>
<tr>
<th></th>
<th>B</th>
<th>~B</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1000</td>
<td>100</td>
</tr>
<tr>
<td>~A</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

Data mining for data quality assurance
Effect of Null Transactions: Negatively Correlated Cases

- Input parameters
  - | | B | –B |
  - | A | 100 | 1000 |
  - | –A | 1000 | |AB| |

- Results

Data mining for data quality assurance
Effect of Null Transactions: Independently Correlated Cases

- Input parameters

<table>
<thead>
<tr>
<th></th>
<th>B</th>
<th>¬B</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>¬A</td>
<td>1000</td>
<td></td>
</tr>
</tbody>
</table>

- Results
Correlations in Asymmetric Data

Input parameters (asymmetric data)

Results

<table>
<thead>
<tr>
<th></th>
<th>B</th>
<th>(~B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td>10000</td>
</tr>
<tr>
<td>(~A)</td>
<td>100</td>
<td>10000</td>
</tr>
</tbody>
</table>

\(\Rightarrow\) IS, \(\alpha\), and \(\gamma\) are good. However, IS doesn’t have downward close property.
CoMine: Efficient Correlation Mining

Utilize the downward close property

- Given a pattern $X$,
  - if $\text{all}_\text{conf}(X) \geq \text{min}_\alpha$, then $\forall Y \subseteq X$, $\text{all}_\text{conf}(Y) \geq \text{min}_\alpha$
  - if $\text{coh}(X) \geq \text{min}_\gamma$, then $\forall Y \subseteq X$, $\text{coh}(Y) \geq \text{min}_\gamma$.

Extend the FP-growth: Additional optimization techniques

- (for both) Counting space pruning
- (for $\gamma$) Efficient computing cardinality of the universe
- (for $\gamma$) Reducing the number of computations of the universe cardinality
How May CrossMine Help Data Quality?

- CrossMine: Efficient classification across multiple database relations
- Originally designed for efficient multi-relational data mining
- Data quality issue exists across multiple relations
  - Data quality assurance is more challenging in multi-relational environment
- Efficient and effective classification across multi-relations will help data cleans and data quality assurance
Multi-Relational Classification

Target relation:
Each tuple has a class label, indicating whether a loan is paid on time.

Example rules:
Loan(L, +) :- Loan (L, A, ?, ?, ?, ?), Account(A, ?, ’monthly’, ?).
Loan(L, +) :- Loan (L, A, ?, ?, ?, ’<1000’), Account(A, D, ?, ?),
District(D, ?, region = ‘northMoravia’, ?, ?, …).

Data mining for data quality assurance
Existing Approaches

Inductive Logic Programming (FOIL, Golem, …)
- Repeatedly find the best predicate.
- To evaluate a predicate $p$ on relation $R$, first join target relation with $R$, which is time consuming.
- Not scalable w.r.t. size of database schema, because of huge search space.

<table>
<thead>
<tr>
<th>loan-id</th>
<th>account-id</th>
<th>amount</th>
<th>duration</th>
<th>payment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>124</td>
<td>1000</td>
<td>12</td>
<td>120</td>
</tr>
<tr>
<td>2</td>
<td>124</td>
<td>4000</td>
<td>12</td>
<td>350</td>
</tr>
<tr>
<td>3</td>
<td>108</td>
<td>10000</td>
<td>24</td>
<td>500</td>
</tr>
<tr>
<td>4</td>
<td>45</td>
<td>12000</td>
<td>36</td>
<td>400</td>
</tr>
<tr>
<td>5</td>
<td>45</td>
<td>2000</td>
<td>24</td>
<td>90</td>
</tr>
</tbody>
</table>

Predicates on Account relation:
Loan $(L, A, ?, ?, ?, ?)$, Account$(A, ?, date<x (date>x))$. 

<table>
<thead>
<tr>
<th>account-id</th>
<th>frequency</th>
<th>date</th>
</tr>
</thead>
<tbody>
<tr>
<td>124</td>
<td>monthly</td>
<td>960227</td>
</tr>
<tr>
<td>108</td>
<td>weekly</td>
<td>950923</td>
</tr>
<tr>
<td>45</td>
<td>monthly</td>
<td>941209</td>
</tr>
<tr>
<td>67</td>
<td>weekly</td>
<td>950101</td>
</tr>
</tbody>
</table>
## Tuple ID Propagation

- Propagate the tuple IDs of the target relation to non-target relations

- Virtually join the relations, but avoid the high cost of physical joins

- Tuple IDs can be propagated freely among relations

- Search for good predicates in promising directions

### Loan

<table>
<thead>
<tr>
<th>loan-id</th>
<th>account-id</th>
<th>amount</th>
<th>duration</th>
<th>payment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>124</td>
<td>1000</td>
<td>12</td>
<td>120</td>
</tr>
<tr>
<td>2</td>
<td>124</td>
<td>4000</td>
<td>12</td>
<td>350</td>
</tr>
<tr>
<td>3</td>
<td>108</td>
<td>10000</td>
<td>24</td>
<td>500</td>
</tr>
<tr>
<td>4</td>
<td>45</td>
<td>12000</td>
<td>36</td>
<td>400</td>
</tr>
<tr>
<td>5</td>
<td>45</td>
<td>2000</td>
<td>24</td>
<td>90</td>
</tr>
</tbody>
</table>

### Account

<table>
<thead>
<tr>
<th>account-id</th>
<th>frequency</th>
<th>date</th>
<th>IDs</th>
<th>Class Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>124</td>
<td>monthly</td>
<td>960227</td>
<td>1, 2</td>
<td>2, 0−</td>
</tr>
<tr>
<td>108</td>
<td>weekly</td>
<td>950923</td>
<td>3</td>
<td>0, 1−</td>
</tr>
<tr>
<td>45</td>
<td>monthly</td>
<td>941209</td>
<td>4, 5</td>
<td>1+, 1−</td>
</tr>
<tr>
<td>67</td>
<td>weekly</td>
<td>950101</td>
<td>--</td>
<td>0, 0−</td>
</tr>
</tbody>
</table>
Algorithm for Finding the Best Predicate

Relations used in the current rule are called *Active Relations*

To compute foil gain of predicates:

- Predicates on active relations are computed directly
- Predicates on relations directly joinable to some active relation: Propagate tuple IDs, then compute
- Predicates on other relations: Do not compute
Algorithm for Finding the Best Predicate

Target relation

Loan
- loan-id
- account-id
- date
- amount
- duration
- payment

Account
- account-id
- district-id
- frequency
- date

Transaction
- trans-id
- account-id
- date
- type
- operation
- amount
- balance
- symbol

District
- district-id
- dist-name
- region
- #people
- #lt-500
- #lt-2000
- #lt-10000
- #lt-10000
- city
- ratio-urban
- avg-salary
- unemploy95
- unemploy96
- den-enter
- #crime95
- #crime96

Card
- card-id
- disp-id
- type
- issue-date

Disposition
- disp-id
- account-id
- client-id

Client
- client-id
- birth-date
- gender
- district-id

First predicate

Second predicate

Data mining for data quality assurance 32
Performance on Synthetic Datasets:

Scalability w.r.t. number of relations

Scalability w.r.t. number of tuples

Figure 9. Runtime on R*.T500.F2.

Figure 11. Runtime on R20.T*.F2.

Performance on Real data set: PKDD Cup 99 dataset

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOIL</td>
<td>74.0%</td>
<td>3338 sec</td>
</tr>
<tr>
<td>TILDE</td>
<td>81.3%</td>
<td>2429 sec</td>
</tr>
<tr>
<td>CrossMine</td>
<td>90.7%</td>
<td>15.3 sec</td>
</tr>
</tbody>
</table>

Data mining for data quality assurance
Privacy-Preserving Document Classification

Smashed documents

Document Owner

Sensitive documents

Document classifiers

Data Miner

Document mining

SecureClass: Privacy-Preserving Classification of Text Documents, by Xiaoxin Yin, Jiawei Han, Anish Mehta

Data mining for data quality assurance
Why Is SecureClass Related to DQ Issues?

Philosophy of SecureClass
- intentionally introduce noises to documents
  so that documents are not understandable
  but still preserves classifiable property

Real data is dirty, but we may still like to do effective classification

Can we explore privacy-preserving mining methodology for effective classification of documents or other kinds of data?

Efficient and effective classification despite of noise
Removing Privacy Information

Randomizing a document

- Remove sensitive words (names, locations, …), numerical data, dates, etc. Only common words are kept
- Smash the order of words
- Remove up to 40% of words and add up to 40% of noises

In regards to fractal compression, I have seen 2 fractal compressed "movies". They were both fairly impressive. The first one was a 64 gray scale "movie" of Casablanca, it was 1.3MB and had 11 minutes of 13 fps video. It was a little grainy but not bad at all. The second one I saw was only 3 minutes but it had 8 bit color with 10fps and measured in at 1.2MB.

I consider the fractal movies a practical thing to explore. But unlike many other formats out there, you do end up losing resolution. I don't know what kind of software/hardware was used for creating the "movies" I saw but the guy that showed them to me said it took 5-15 minutes per frame to generate. But as I said above playback was 10 or more frames per second. And how else could you put 11 minutes on one floppy disk?

davidr@rincon.ema.rockwell.com

My opinions are my own except where they are shared by others in which case I will probably change my mind.
Document Classification Process

- Build rules that predict for class labels with a sequential covering algorithm.
  - routine, polygon $\rightarrow$ computer graphics
  - a frequent pattern $\quad$ a class label

- Rules may come from noises. Use following constraints to rules:
  - Rules with high support are less likely to come from noises
  - Longer rules are less likely to come from noises
  - For each rule $r = \text{“}w_1, \ldots, w_k \rightarrow c\text{“}$
    - Make sure that $r$’s confidence is improved at most $\varepsilon$ by noises, with probability $(1-\delta)$. 

Data mining for data quality assurance 37
Experimental Results

Accuracy on newsgroup dataset

Accuracy on BankSearch dataset

SecureClass is more accurate than SVM, Naïve Bayes, and CMAR.
The accuracy of SecureClass is less affected than those three approaches.
The efficiency of SecureClass is similar to SVM, and is slower than Naïve Bayes but faster than CMAR.
Conclusions

Data Mining helps data quality assurance
  Not only by traditional statistical, machine learning, data mining methods
  But also potentially with newer techniques

Explore how to explore new data mining methods for data quality assurance
  Object matching using profilers, statistical analysis, etc.
  Correlation mining
  Cross-relational data mining
  Privacy-preserving data mining
  And potentially many others!
www.cs.uiuc.edu/~hanj

Thank you !!!