Schema Extraction

Divesh Srivastava (AT&T Labs – Research)

Joint work with Marios Hadjieleftheriou, Beng Chin Ooi, Cecilia M. Procopiuc, Xiaoyan Yang, Meihui Zhang
Motivation

♦ Information extraction
  – Extracting structure (e.g., tables) from unstructured data (e.g., text)

♦ Schema extraction
  – Extracting schema (σχήμα) from structured (e.g., tabular) data
  – Wealth of tabular data, e.g., spreadsheets, web tables, ...
  – Schema includes keys, foreign keys, table spaces, ...
  – Knowledge of database schema enables richer queries (e.g., joins), more sophisticated data analysis
Motivation: TPCE Schema Graph

TPCE

Broker
Customer
Market
Dimension
Outline

♦ Motivation
  – Extracting schema from tabular data

♦ Discovering good foreign keys from tabular data
  – Schema graph = nodes (tables, attributes) + edges (foreign keys)

♦ Discovering good table spaces
  – Clustering tables by topic, identifying important tables
Discovering Foreign Keys: Motivation

- Foreign/primary key relationship is an important constraint in relational databases

- Knowing foreign keys is often a crucial step in understanding and analyzing the data
Discovering Foreign Keys: Motivation

- In practice, foreign keys are often **NOT** specified in the schema

- Reasons
  - Associations not known to DB designers but inherent in data
  - Implicit relationships across multiple databases
  - Data inconsistencies (data integration, database evolution, ...)
  - ...
Existing Work

- Little previous work on discovering **multi-column** foreign keys

- Most focus mainly on identifying **inclusion dependencies** \([1,2,3]\)
  - The only formal requirement (a subset of primary key)
  - Not enough

→ a large number of false positives

Existing Work
Existing Work

- Heuristic rules to reduce the number of false positives [4]
  - The column names of foreign/primary keys should be similar
  - A foreign key should have significant cardinality
  - A foreign key should have good coverage of the primary key
  - The primary key should have only a small percentage of values outside the range of the foreign key
  - The average length of the values in foreign/primary key columns should be similar (mostly for strings)
  - ...

- Counter-examples exist for any rule!

Our Approach

♦ Randomness

– Measuring the likelihood that \((F, P)\) is a useful FK/PK constraint
– Thesis: values in \(F\) form a random sample of (ordered) values in \(P\)
– No correlation between the semantics of the table with the foreign key and the way the primary keys are generated
– In dynamic databases, the distributions change over time
Outline

♦ Motivation
  – Extracting schema from tabular data

♦ Discovering good foreign keys from tabular data
  – Schema graph = nodes (tables, attributes) + edges (foreign keys)
  – Inclusion, randomness

♦ Discovering good table spaces
  – Clustering tables by topic, identifying important tables
Inclusion

- Partial inclusion $\sigma(F,P)$ : user defined threshold
  - $\sigma(F,P) = \frac{|F \cap P|}{|F|}$
  - $\sigma(F,P) \geq 0.9$

- For efficiency
  - bottom-k sketch \[^5\]

\[^5\] Edith Cohen, Haim Kaplan: Leveraging discarded samples for tighter estimation of multiple-set aggregates. SIGMETRICS/Performance 2009
Inclusion

♦ Partial inclusion $\sigma(F,P)$ : user defined threshold
  - $\sigma(F,P) = |F \cap P|/|F|$
  - $\sigma(F,P) \geq 0.9$

♦ For efficiency
  – bottom-k sketch $[^5]$
  – $\sigma(F,P) = \text{Jacc}(F,P)/\text{Jacc}(F \cup P,F)$
Randomness

♦ Randomness test
  – Given F and P, test if the distinct values (tuples) in F have the same underlying distribution as the values (tuples) in P

♦ Domain order
  – Numerical order: numeric values
  – Lexicographic order: strings
Randomness Measure

- Earth Mover’s Distance (EMD)
  - Standard distance measure between probability distributions
  - EMD measures the amount of work needed to convert 1\textsuperscript{st} distribution into the 2\textsuperscript{nd}
Distance Function

- Normalized distance between ranks
  - Independent of the actual values in any column

- Single-column
  - Absolute difference between the ranks in the underlying ordered space (PK column)
  - Normalize by the number of values

- Multi-column
  - Manhattan distance
  - Normalize by dimensionality
Probability Distribution

- Exact distribution
  - let each value in $F (P)$ have a probability of $1/|F| \ (1/|P|)$

Computing EMD is **too expensive** over large $F (P)$
(Hungarian algorithm has cubic complexity)
Probability Distribution

- Quantile histogram
  - $\ell$-quantiles of PK

- One dimension
  - Equi-depth
Probability Distribution

- Quantile histogram
  - $\ell$-quantiles of PK
  - Probability distribution of FK is defined w.r.t quantiles of PK

- Multi-dimension
  - Compute quantiles separately on each dimension
  - Construct a grid
Overall Algorithm

1. **Inclusion**
   - bottom-k sketch

2. **Randomness**
   - quantile summary

<table>
<thead>
<tr>
<th>Function</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1.id -&gt; P1.cid</td>
<td>0.001</td>
</tr>
<tr>
<td>F2.id -&gt; P3.tid</td>
<td>0.002</td>
</tr>
<tr>
<td>F3.id -&gt; P1.cid</td>
<td>0.002</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Outline

♦ Motivation
  – Extracting schema from tabular data

♦ Discovering good foreign keys from tabular data
  – Schema graph = nodes (tables, attributes) + edges (foreign keys)
  – Inclusion, randomness
  – Experimental results

♦ Discovering good table spaces
  – Clustering tables by topic, identifying important tables
Experiments

♦ Datasets
  – Benchmark databases: TPC-E, TPC-H
  – Real databases: Wikipedia, IMDB

♦ Evaluation
  – Accuracy
  – Scalability
  – Comparison
Experiments

- Number of candidates after inclusion test

<table>
<thead>
<tr>
<th>Dataset</th>
<th>TPC-H</th>
<th>TPC-E</th>
<th>Wikipedia</th>
<th>IMDB</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC-FK</td>
<td>9</td>
<td>44</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>MC-FK</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>θ =</td>
<td>0.9</td>
<td>1</td>
<td>0.9</td>
<td>0.9</td>
</tr>
<tr>
<td>SC-Candidates</td>
<td>38</td>
<td>304</td>
<td>12</td>
<td>24</td>
</tr>
<tr>
<td>MC-Candidates</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

© 2010 AT&T Intellectual Property. All rights reserved. AT&T and the AT&T logo are trademarks of AT&T Intellectual Property.
Accuracy

TPC-H

Wikipedia
Accuracy

TPC-E

IMDB
Scalability

- TPC-H: 1M, 10M, 100M, 1G, 10G

![Graph showing scalability with TPC-H benchmarks and time in hours]
Comparison

♦ Machine learning approach [4]
  – Use 7 heuristic rules
  – Need learning phase to train 4 classifiers
  – Need known foreign/primary key pairs for training
  – Discover single-column keys only
  – TPC-H: F-measure = 0.95 (best classifier J48)
        F-measure = 0.915 (average all classifiers)

♦ Our approach
  – TPC-H: F-measure = 0.95

Summary

♦ Introduce randomness and show it can discover meaningful foreign keys, including multi-column foreign keys

♦ Provide efficient algorithm for evaluating randomness

♦ Present I/O efficient algorithm for discovering good foreign keys

♦ Experiments show the efficacy of our techniques
Outline

- Motivation
  - Extracting schema from tabular data

- Discovering good foreign keys from tabular data
  - Schema graph = nodes (tables, attributes) + edges (foreign keys)

- Discovering good table spaces
  - Clustering tables by topic, identifying important tables
Discovering Table Spaces: Motivation

- Complex databases are challenging to explore and query
  - Consisting of hundreds of inter-linked tables
  - Users unfamiliar with the schema
  - Insufficient or unavailable schema information

- Propose a principled approach to discover table spaces
  - Cluster similar tables
  - Label each cluster by its most important table
Discovering Table Spaces: Motivation

TPCE

Broker
Customer
Market
Dimension
Outline

♦ Motivation
  – Extracting schema from tabular data

♦ Discovering good foreign keys from tabular data
  – Schema graph = nodes (tables, attributes) + edges (foreign keys)

♦ Discovering good table spaces
  – Clustering tables by topic, identifying important tables
  – Table importance, weighted k-center clustering
Table Importance

- Depends on
  - Internal information content
  - External connectivity
    - Join behavior
    - Taxrate: 1 join
    - Customer: 5 joins
Table Importance (cont’d)

♦ Entropy of Attribute $A$ in table $R$ is defined as

$$H(R, A) = \sum_{i=1}^{k} p_i \log(1/p_i)$$

- $R.A = \{a_1, ..., a_k\}$
- $p_i$ is the fraction of tuples in $R$ that have value $a_i$ on attribute $A$

♦ The Information Content of a table $R$ is defined as

$$IC(R) = \log|R| + \sum_{R.A} H(R, A)$$

- Create a primary key $R.Key$ to table $R$
- Add a self-loop $R.Key - R.Key$

*R.Key consists of all attributes*
Table Importance (cont’d)

- **Entropy transfer matrix** $\Pi$ associated with schema graph $G$ is defined as:
  - For a join edge $e = R.A - S.B$
    \[
    Pr(R.A \rightarrow S.B) = \frac{H(R.A)}{\log|R| + \sum_{R.A'} q_{A'} \cdot H(R.A')} \cdot \frac{H(R.A)}{IC(R) + \sum_{R.A'} (q_{A'} - 1) \cdot H(R.A')}
    \]
    $q_A$: number of join edges involving $R.A'$ (including self-join)

- For a pair of tables $R$ and $S$, define
  \[
  \Pi[R, S] = \sum_{R.A - S.B} Pr(R.A \rightarrow S.B), \quad \Pi[R, R] = 1 - \sum_{S \neq R} \Pi[R, S]
  \]
The importance of table $R$ is defined as the stable-state value of a random walk on $G$, using probability matrix $\Pi$.

- Vector $\mathcal{I}$, s.t. $\mathcal{I} \times \Pi = \mathcal{I}$
- Importance $\mathcal{I}(R)$, $R \in G$

**Example**
Table Similarity

- Distance = 1 - similarity
- Goal: define metric distance
  - Enables meaningful clustering over relational databases
- Table similarity depends on how *join edges* and *join paths* are instantiated

\[ R.A = S.B \]
Table Similarity (cont’d)

- Consider a join edge \( e = R.A - S.B \)
  - Tuples \( t_1, t_2 \) instantiate \( e \)
  - \( \text{fanout}_e(t_i) \) is the fanout of \( t_i \) along \( e \)
    - \( \text{fanout}_e(t_1) = 3 \)
  - Let \( q \) be the number of tuples in \( R \) s.t. \( \text{fanout}_e(t_i) > 0 \), define the matching fraction of \( R \) w.r.t. \( e \) as \( f_e(R) = q/n, |R| = n \)
    - \( f_e(R) = 2/3 \leq 1; \quad f_e(S) = 5/5 = 1 \leq 1 \)
  - Define the matched average fanout of \( R \) w.r.t. \( e \) as
    \[
    maf_e(R) = \frac{\sum_{i=1}^{n} \text{fanout}_e(t_i)}{q}
    \]
    - \( maf_e(R) = (3+2)/2 = 2.5 \geq 1; maf_e(S) = 5/5 = 1 \geq 1 \)
Table Similarity (cont’d)

- The similarity of tables $R$ and $S$ (w.r.t. $e_{(R,S)}$) must satisfy:
  - Property 1: Proportional to the matching fractions $f_{e(R)}$ and $f_{e(S)}$.
  - Property 2: Inverse proportional to the matched average fanouts $maf_{e}(R)$ and $maf_{e}(S)$.

- Define the strength of tables $R$ and $S$ (w.r.t. $e_{(R,S)}$) as

  - Property 2: Inverse proportional to the matched average fanouts $maf_{e}(R)$ and $maf_{e}(S)$.
Table Similarity (cont’d)

- Let $\pi : R = R_0 - R_1 - \ldots - R_\alpha = S$ be a path in $G$, define
  
  $\text{Strength}_\pi(R, S) = \prod_{i=1}^{\alpha} \text{Strength}_{e_i}(R_{i-1}, R_i)$

- Table similarity $(R, S)$:
  
  $\text{Strength}(R, S) = \max_{\pi} \text{Strength}_\pi(R, S)$

- Distance $(R, S)$
  
  - $\text{dist}_s(R, S) = 1 - \text{strength}(R, S)$
  
  - $(R, \text{dist}_s)$ is a metric space
Clustering: Weighted $k$-Center

- **Clustering Criteria:**
  - Minimize the maximum *distance* between a cluster center and a table in that cluster
  - Take table *importance* into consideration, avoid grouping top important tables into one cluster

- **Weighted $k$-Center clustering**
  - Weights: table importance
  - Given $k$ clusters $C = \{C_1, C_2, ..., C_k\}$, minimize
    
    $$
    \mu(C) = \max_{i=1}^{k} \max_{R \in C_i} \mathcal{I}(R) \text{dist}(R, \text{center}(C_i))
    $$
  
  - NP-Hard
Weighted $k$-Center: Greedy Algorithm

$\text{GREEDYCLUS}(G = (R, E), k)$

$\mathcal{C} = \{C_1\}$: current clustering;
1. $\text{center}(C_1) = R_1$ s.t. $I(R_1) = \max_{R \in R} I(R)$;
2. $\text{cluster}(R) = C_1, \forall R \in R$: assign all tables to $C_1$;
3. for $i = 2$ to $k$
   
   /* $\Delta(R) = I(R) \text{dist}(R, \text{center}($cluster$(R)))$*/
   4. $\text{center}(C_i) = R_i$ s.t. $\Delta(R_i) = \max_{R} \Delta(R)$;
   5. for each $R \in R$
   6. if $\text{dist}(R, \text{center}($cluster$(R))) > \text{dist}(R, R_i))$
   7. $\text{cluster}(R) = C_i$;
   8. endfor
   9. $\mathcal{C} = \mathcal{C} \cup \{C_i\}$
10. endfor
11. return $(\mathcal{C}, \text{cluster}(\cdot))$

- Start with one cluster, whose center is the **top-1 important table**.
- Iteratively chooses the table $R_i$ whose **weighted distance** from its cluster center is largest, and creates a new cluster with $R_i$ as its center.
- All tables that are closer to $R_i$ than to their current cluster center are reassigned to cluster $C_i$. 
Outline

♦ Motivation
  – Extracting schema from tabular data

♦ Discovering good foreign keys from tabular data
  – Schema graph = nodes (tables, attributes) + edges (foreign keys)

♦ Discovering good table spaces
  – Clustering tables by topic, identifying important tables
  – Table importance, weighted k-center clustering
  – Experimental results
Experimental Results

- Validate the proposed three components in our approach
  - Model for table importance $I_E$ (Entropy-based)
  - Distance function $dist_s$ (Strength-based)
  - Clustering: Weighted $k$-Center

- Other methods

<table>
<thead>
<tr>
<th>Table Importance</th>
<th>Distance</th>
<th>Clustering</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_E$</td>
<td>$dist_s$</td>
<td>Weighted $k$-Center</td>
</tr>
<tr>
<td>$I_c^{[1]}$</td>
<td>$dist_c^{[1]}$</td>
<td>Balanced-Summary$^{[1]}$</td>
</tr>
<tr>
<td></td>
<td>$dist_p^{[2]}$</td>
<td></td>
</tr>
</tbody>
</table>

- $l_c$: Cardinality-initialized
- $dist_c = 1 - \text{coverage}$
- $dist_p = 1 - \text{proximity}$

Experimental Results (cont’d)

- **Data Sets: TPCE schema**
  - Benchmark database simulating OLTP workload
  - 33 tables pre-classified into 4 categories
  - Two database instances: TPCE-1 / TPCE-2

<table>
<thead>
<tr>
<th>Parameters</th>
<th>TPCE-1</th>
<th>TPCE-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of customers</td>
<td>1,000</td>
<td>5,000</td>
</tr>
<tr>
<td>Initial Trade Days</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Scale Factor</td>
<td>1,000</td>
<td>36,000</td>
</tr>
</tbody>
</table>

- Affect the size of the majority of tables
- Affect $Pr(R.A \rightarrow S.B)$, $strength(R,S)$ for most pairs and $maf_e$ for 1/3 of edges
# Table Importance

- Comparison of $I_E$ and $I_C$ models

- Top-5 Important Tables in $I_E$ and their ranks in $I_C$

$I_E$ more accurate than $I_C$

- Top-5 Important Tables in $I_C$ and their ranks in $I_E$

<table>
<thead>
<tr>
<th>Rank</th>
<th>Table</th>
<th>Info. Content</th>
<th>$I_E$</th>
<th>$I_C$ Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Trade</td>
<td>39.730</td>
<td>57.798</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Security</td>
<td>37.350</td>
<td>41.405</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>Customer</td>
<td>45.781</td>
<td>36.202</td>
<td>17</td>
</tr>
<tr>
<td>4</td>
<td>Financial</td>
<td>43.575</td>
<td>30.647</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>Holding</td>
<td>26.112</td>
<td>28.866</td>
<td>11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rank</th>
<th>Table</th>
<th>Card.</th>
<th>$I_C$</th>
<th>$I_E$ Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Trade</td>
<td>576000</td>
<td>1805787.6</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Trade_History</td>
<td>1382621</td>
<td>659751.7</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>Status_Type</td>
<td>5</td>
<td>503280.9</td>
<td>32</td>
</tr>
<tr>
<td>4</td>
<td>Security</td>
<td>685</td>
<td>487461.5</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>Holding_History</td>
<td>722143</td>
<td>321415.2</td>
<td>9</td>
</tr>
</tbody>
</table>
### Table Importance (cont’d)

- **Consistency of $I_E$ and $I_C$ models**

#### Top-7 Important Tables in $I_E$ and $I_C$ for TPCE-1 and TPCE-2

<table>
<thead>
<tr>
<th>Rank</th>
<th>$I_E$/TPCE-1</th>
<th>$I_E$/TPCE-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Trade</td>
<td>Trade</td>
</tr>
<tr>
<td>2</td>
<td>Security</td>
<td>Security</td>
</tr>
<tr>
<td>3</td>
<td>Customer</td>
<td>Customer</td>
</tr>
<tr>
<td>4</td>
<td>Financial</td>
<td>Financial</td>
</tr>
<tr>
<td>5</td>
<td>Holding</td>
<td>Company</td>
</tr>
<tr>
<td>6</td>
<td>Company</td>
<td>Customer_Account</td>
</tr>
<tr>
<td>7</td>
<td>Customer_Account</td>
<td>Holding</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rank</th>
<th>$I_C$/TPCE-1</th>
<th>$I_C$/TPCE-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Trade</td>
<td>Security</td>
</tr>
<tr>
<td>2</td>
<td>Trade_History</td>
<td>Daily_Market</td>
</tr>
<tr>
<td>3</td>
<td>Status_Type</td>
<td>Watch_Item</td>
</tr>
<tr>
<td>4</td>
<td>Security</td>
<td>Watch_List</td>
</tr>
<tr>
<td>5</td>
<td>Holding_History</td>
<td>Trade</td>
</tr>
<tr>
<td>6</td>
<td>Daily_Market</td>
<td>Trade_History</td>
</tr>
<tr>
<td>7</td>
<td>Customer_Account</td>
<td>Customer_Account</td>
</tr>
</tbody>
</table>

$I_E$ more consistent than $I_C$
Distance Between Tables

- **Accuracy of distance functions**
  - Observation: for each table $R$, its distances to tables within the same category (*pre-defined*) should be smaller than its distances to tables in different categories
  - $n(R)$: # top-$q$ nbrs ($NN_R$) of $R$ under dist. $d$ ($dist_s, dist_c, dist_p$)
  - $m(R)$: # tables ($\in NN_R$) in the same category as $R$ under dist. $d$
  - Calculate:

\[
acc(d) = \frac{\sum_R m(R)}{n} \frac{n(R)}{n}
\]

<table>
<thead>
<tr>
<th>$q=5$</th>
<th>All tables</th>
<th>No <em>Dimension</em> tables</th>
</tr>
</thead>
<tbody>
<tr>
<td>$dist_s$</td>
<td>0.659</td>
<td>0.649</td>
</tr>
<tr>
<td>$dist_c$</td>
<td>0.589</td>
<td>0.621</td>
</tr>
<tr>
<td>$dist_p$</td>
<td>0.5</td>
<td>0.557</td>
</tr>
</tbody>
</table>

$dist_s$ most accurate
Table Space Discovery Algorithms

- Weighted $k$-Center over three distance functions

<table>
<thead>
<tr>
<th>$k$</th>
<th>$C_i$</th>
<th>$dist_n$</th>
<th>$n(C_i)$</th>
<th>$m(C_i)$</th>
<th>acc($C_i$)</th>
<th>$dist_c$</th>
<th>$n(C_i)$</th>
<th>$m(C_i)$</th>
<th>acc($C_i$)</th>
<th>$dist_p$</th>
<th>$n(C_i)$</th>
<th>$m(C_i)$</th>
<th>acc($C_i$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Trade</td>
<td>9</td>
<td>6</td>
<td>0.67</td>
<td>Trade</td>
<td>19</td>
<td>8</td>
<td>0.42</td>
<td>Trade</td>
<td>21</td>
<td>8</td>
<td>0.38</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Security</td>
<td>13</td>
<td>11</td>
<td>0.85</td>
<td>Financial</td>
<td>7</td>
<td>7</td>
<td>1.0</td>
<td>Security</td>
<td>6</td>
<td>4</td>
<td>0.67</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Customer</td>
<td>10</td>
<td>6</td>
<td>0.6</td>
<td>Customer</td>
<td>6</td>
<td>3</td>
<td>0.5</td>
<td>Customer</td>
<td>5</td>
<td>2</td>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Trade</td>
<td>9</td>
<td>6</td>
<td>0.67</td>
<td>Trade</td>
<td>13</td>
<td>7</td>
<td>0.54</td>
<td>Trade</td>
<td>14</td>
<td>8</td>
<td>0.57</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Security</td>
<td>12</td>
<td>10</td>
<td>0.83</td>
<td>Financial</td>
<td>7</td>
<td>7</td>
<td>1.0</td>
<td>Security</td>
<td>6</td>
<td>4</td>
<td>0.67</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Customer</td>
<td>10</td>
<td>6</td>
<td>0.6</td>
<td>Customer</td>
<td>6</td>
<td>3</td>
<td>0.5</td>
<td>Customer</td>
<td>5</td>
<td>2</td>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Financial</td>
<td>1</td>
<td>1</td>
<td>1.0</td>
<td>Security</td>
<td>4</td>
<td>6</td>
<td>0.67</td>
<td>Financial</td>
<td>7</td>
<td>7</td>
<td>1.0</td>
<td></td>
</tr>
</tbody>
</table>

✓ Summary Accuracy

$acc(C) = \sum_{i=1}^{k} \frac{m(C_i)}{n}$

$dist_5$: most balanced and accurate
Summary

♦ Novel approach for discovering good table spaces
  – A new model for table importance
  – A metric distance over schema tables
  – A summarization algorithm

♦ Ongoing work
  – Summarizing schema graphs for at-a-glance understanding
Parting Thoughts

- Schema extraction is critical for automatically creating databases from collections of tables
  - We focused on discovering good foreign keys, tables spaces

- Other work on discovering good primary keys, good FDs:
  - P. Andritsos, R. Miller, and P. Tsaparas. Information-theoretic tools for mining database structure from large data sets. SIGMOD 2004

- Exciting research area with a lot of practical utility!
Discovering Foreign Keys: Motivation

♦ In practice, foreign keys are often NOT specified in the schema

♦ What if this happens in enterprise databases?
  – Thousands of tables
  – Tens of thousands of columns
  – Insufficient (missing/out-of-date) documentation
Objective

- To efficiently discover FK/PK relationships in relational databases
  - Single-column
  - Multi-column
Randomness

• Randomness measure
  – How close are the (multi-dimensional) distributions of $F$ and $P$?
Our Approach

- Counter-examples exist for randomness rule as well
  - Table $P$ contains all NUS graduate students
  - $SID$ is generated according to the year, e.g. g10xxxxx
  - Table $F$ references only the students who enrolled in NUS in 2010
  - $F.SID$ is not a random sample of $P.SID$
  - Foreign key table is correlated to the way keys are generated
Our Approach

♦ Counter-examples exist for randomness rule as well
  – Table $P$ contains all NUS graduate students
  – $SID$ is generated according to the year, e.g. g10xxxxxx
  – Table $F$ references the students who come from China
  – $F.SID$ is a random sample of $P.SID$

♦ No solution with 100% precision/recall

♦ Experiments on real databases show randomness rule can effectively eliminate false positives and achieve high recall!
Overall Algorithm

♦ Two passes over data

♦ Phase 1
  – Read all columns in table-wise order
  – Build bottom-k sketches for all single columns and all multi-column PKs
  – Build quantile summaries for all single/multi-column PKs
  – Evaluate single-column inclusions
Overall Algorithm

♦ Two passes over data

♦ Phase 2
  – Compute multi-column candidate FKs
  – For each single-column candidate FK, scan it, compute distribution histograms w.r.t all relevant PKs
  – For each multi-column candidate FK, scan it, compute bottom-k sketch and distribution histograms w.r.t all PKs
  – Evaluate randomness
Clustering Algorithms

- **Accuracy of a summary**
  - TPCE is pre-classified into 4 categories: *Broker, Customer, Market* and *Dimension*
  - \( m(C_i) \): # tables in \( C_i \) with the same category as \( center(C_i) \)
  - Given a summary \( C = \{C_1, C_2, \ldots, C_k\} \), calculate \( \text{acc}(C) = \frac{\sum_{i=1}^{k} m(C_i)}{n} \)
  - Balanced-Summary (BS) [1]
  - Weighted \( k \)-Center (WKC)

Based on \( I_C \) and \( \text{dist}_C \) (coverage)

WKC is more accurate
Related Work

♦ C. Yu and H. V. Jagadish. *Schema summarization*. VLDB’06
♦ H. Tong, C. Faloutsos and Y. Koren. *Fast direction-aware proximity for graph mining*. KDD’07
♦ W. Wu, B. Reinwald, Y. Sismanis and R. Manjrekar. *Discovering topical structures of databases*. SIGMOD’08