CORDS
Finding Dependencies For Accurate Selectivity Estimation
By
Ihab Ilyas
SELECT o.name, a.driver
FROM owner o,
     car c,
     demographics d,
     accidents a
WHERE c.ownerid = o.id AND
     o.id = d.ownerid AND
     c.id = a.id AND
     c.make = 'Mazda' AND
     c.model = '323' AND
     o.country3 = 'EG' AND
     o.city = 'Cairo' AND
     d.age < 30;
SELECT o.name, a.driver
FROM owner o,
car c,
demographics d,
accidents a
WHERE
c.ownerid = o.id AND
o.id = d.ownerid AND
c.id = a.id AND
c.make = 'Mazda' AND
c.model = '323' AND
o.country3 = 'EG' AND
o.city = 'Cairo' AND
d.age < 30 ;
Motivation (Cont’d)

• The Independence Assumption
  - Orders of magnitude error in estimating selectivity
  - Optimizer chooses sub-optimal plans

• A simple solution: build statistics on groups of columns

• The Challenge: Huge # of possible groups
  - Get highly “correlated” groups only
Cords

- A system for automatically detecting
  - Soft functional dependencies
  - Correlations (statistical dependencies)

- Applications
  - Data mining
  - Query optimization (our main focus)
Notations

- Relation $R$ with two columns $A, B$
- A predicate $P_A$: “$A = \text{const}$”
- Selectivity of a predicate $S(P_A) = 1/|A|$
- $S(P_A \land P_B) = 1/|A,B|$
- Under independence and Uniformity $S(P_A \land P_B) = 1/|A| \times 1/|B|$
Previous Approaches (IBM)

• **Query Driven (LEO):**
  - Compare the actual selectivity to the estimated (adjustment factor)
  - Identify groups with large adjustments
  - Limited to columns in available workload
  - “Learning” can take time

• **Data-driven (B-Hunt)**
  - Look at the data
  - Identify columns with algebraic constraints
  - Rewrite query to exploit the algebraic constraints
CORDS

- Discover **correlation** between columns within or across tables
- **Not** limited to numerical data
- **Not** limited to algebraic relationships
- The goal is to **collect useful joint statistics** not to rewrite the query
- Useful for many other applications
  - Schema discovery, mining,..
CORDS: Overview

• Phase1: Enumeration
  - Enumerate all possible candidate column pairs
  - Apply pruning rules to limit # for Phase 2

• Phase2: Correlation detection
  - For each candidate column pair:
    • Test for spurious correlation (trivial cases)
    • Test for soft functional dependency
    • Test for correlation
CORDS: Enumeration

- All possible column pairs
  - Within each table (“trivial” pairing rule)
  - Across all joinable tables (PK-FK pairing rule)

- Prune the candidates (flexible rule set)
  - **Type** constraints
    - No CLOBs or BLOBs
    - Compatible types
  - **Pairing** constraints
    - Declared PK with all possible FK
    - Declared PK and FK
  - **Workload** constraints
CORDS: Correlation Detection

[1] Test for trivial cases (assume $|A| \geq |B|$)
   IF $|A| \approx |R|$: RETURN ("A is a soft key")
   IF $|A| \approx 1$ or $|B| \approx 1$: RETURN ("Trivial column")

[2] Sample R to get S

   IF $|S| \gg |A,B|$ AND $|A| \approx |A,B|$:  
      RETURN ("$A \rightarrow B$ with strength $|A|/|A,B|$")

[4] Skew Handling for Chi-squared Test
   IF “skewed”: FILTER S with the frequent values

[5] Sampling-based Chi-squared Test
   Build a (skew-dependent) contingency table for $A \times B$ from S
   Apply Chi-squared test
   If correlated, RETURN ("Correlated with degree of correlation = x")
   else RETURN ("not correlated")
CORTDS: Sampling

• **Choose size of S such that**
  - \( \Pr(\text{"correlated" | correlation} < \delta) < p \)
  - \( \Pr(\text{"not correlated" | correlation} > \delta) < p \)

• **Required sample size independent of**
  - # rows in R
  - Dimensions of contingency table (almost)
  - Error probability p (almost)

• **Novel approximation for sample size**
  - Special case: \( d \times d \) contingency table
    \[
    n \approx \frac{-16d^2 \log(p\sqrt{2\pi})^{1/2} - 8\log(p\sqrt{2\pi})}{1.69\delta d^{0.858}}
    \]

Correlation measured by “mean-square contingency”
A Fixed Sample Size is OK

\[ \delta = 0.005 \]

\[
\begin{array}{c}
\text{Required Sample Size} \\
\text{Maximum Allowed Error Probability (p)}
\end{array}
\]

- \(10 \times 10\) (solid line)
- \(50 \times 50\) (dashed line)
Correlation Discovery: Requirements

• We need a low-overhead technique
• False positives are ok (if not many)
• Pairs give good correction
• Diminishing return for higher order groups
Dependency Graph

SYNTHETIC DATABASE
SAMPLE SIZE: 8000
Dependency Graph (across tables)

SYNTHETIC DATABASE
SAMPLE SIZE: 10000
CATALOG_RETURNS
126233 rows
7.9 % sample rate
CATALOG_SALES
1260000 rows
0.8 % sample rate
CUSTOMER
50000 rows
20 % sample rate
Column Group Stats (CGS)

- Statistics about a group of columns
- Basic CGS is the number of distinct combinations for a group of columns
- For a column group \((A,B,C)\) a basic CGS is \(|A,B,C|\)
- CGS gives a better estimate of joint selectivity by not assuming independence
- CGS still assumes uniform distribution of data
Using Column Group Stats (CGS)

• $P_A$: “Make = Mazda” and $P_B$: “Model = 323”

• Assuming uniformity & independence:
  \[ \text{Selectivity}(P_A \text{ AND } P_B) = \frac{1}{|\text{Make}|} \times \frac{1}{|\text{Model}|} \]

• Exploit CGS $|\text{Make,Model}|$:
  - Apply adjustment factor $= \frac{|\text{Make}| \times |\text{Model}|}{|\text{Make,Model}|}$
  - Selectivity($P_A$ AND $P_B$) = $1 / |\text{Make,Model}|$

• Error due to faulty independence assumption is eliminated!

• Error due to uniformity assumption remains
  - In practice, most error is due to independence assumption
  - Future work: exploit column group distribution statistics
CORDS for Query Optimization

- Column Group Statistics
- STATS COLLECTION
- Recommend CGS
- Dependency Discovery
  (Optional)

- CATALOG INFO
- DATA
- Sample

- Optimizer
Recommending CGS

• Rank Soft FD’s by their Strength

• Rank correlation by their degree of correlation

• Break ties using the adjustment factor:
  For a column pair \((A, B)\)
  
  \[
  \text{Adjustment factor} = \frac{|A| \times |B|}{|A, B|}
  \]
Recommending CGS

Census Data (2000 samples)
2065 Discovered Correlations

Number of CGS Recommended

Adj. Factor Threshold

P-value Threshold

26
Recommending CGS

Census Data (2000 samples)
114 Soft Fds Discovered

Number of Recommended FDS

Adj. Factor Threshold

Strength Threshold

2 4 6 0.99 0.98 0.96 0.94 0.92 0.9

20 40 60 80 100 120

45 90 135
Recommending CGS

REAL Data (10000 samples)
1818 Discovered Correlations
Recommending CGS

REAL data (1000 samples)
187 Discovered Soft FDs
Experiments (Performance)
Experiments (Accuracy vs. Time)
Experiments (Diminishing Return)

![Bar chart showing worst-case error factor for different orders of CG statistics: Single Column has the highest error factor, followed by 2 Columns and then 3 Columns.]
Column Group Distribution Statistics (CGDS)

- Example: In the CARS database:
  - The accidents table has a correlation between “SEATBELTON” and “DRIVER” because of the following Association:

<table>
<thead>
<tr>
<th>SEATBELT</th>
<th>DRIVER</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>UNHARMED</td>
<td>0.7</td>
</tr>
<tr>
<td>YES</td>
<td>INJURED</td>
<td>0.25</td>
</tr>
<tr>
<td>YES</td>
<td>DEAD</td>
<td>0.05</td>
</tr>
<tr>
<td>NO</td>
<td>UNHARMED</td>
<td>0.1</td>
</tr>
<tr>
<td>NO</td>
<td>INJURED</td>
<td>0.8</td>
</tr>
<tr>
<td>NO</td>
<td>DEAD</td>
<td>0.1</td>
</tr>
</tbody>
</table>

- $|\text{SEATBELT}| = 2$, $|\text{DRIVER}| = 3$ and $|\text{SEATBELT, DRIVER}| = 6$
- $\text{ADJUSTMENT} = (2 \times 3) / 6 = 1$!
Column Group Distribution Statistics (CGDS)

- For more details, column group distribution stats
- CGDS give information like the top 10 frequent values and their frequencies and the n quantiles
- Adjustment factor $\frac{|A=a| \times |B=b|}{|A=a, B=b|}$
Other Related Work

• **Data-driven:**
  - **Bayesian/Markov networks**
    • Correlation criteria: conditional independence, x-entropy, mean-square contingency, etc.
    • Scalability issues: Can be expensive to construct, maintain
  
  - **Mining of FDs and semantic integrity constraints**
    • Exact results obtained
    • No sampling, so very expensive
  
  - **Association-rule mining**
    • Relations between specific attribute *values*
    • CORDS considers attributes as a *whole*
Related Work (Others)

• Query-driven:
  - SITs
    • Query workload + optimizer estimates determine stored stats (single column of views)

  - STHoles
    • Detects correlation for specified columns

  - SASH
    • Dynamic Markov network model (scalability?)
Advantages of CORDS

• Simplicity
  - Pairwise correlations only
  - Effective combination of simple algorithms

• Scalability to large DBs
  - Simplicity + use of sampling

• Feasible and effective for commercial systems
  - Relatively easy to implement
  - Low runtime overhead
  - Large speedups in query processing
Conclusion

• **Goal:** Automatically, efficiently discover correlations + soft FDs

• **A simple and effective solution:** CORDS
  - Enumeration + Pruning Rules + Sampling + Chi-square/Counting
  - Dependency graphs for mining
  - CGS ranking and exploitation for optimization

• **Future work**
  - 3-way dependencies?
  - Interactive dependency graphs ("slider bars")
  - Applications to schema discovery
  - Synthesize query + data-driven approaches
  - XML data?