CORDS:
Automatic Discovery of Correlations and Soft Functional Dependencies

Ihab Ilyas* Volker Markl* Peter Haas*
Paul Brown* Ashraf Aboulnaga*

*IBM Almaden Research Center
+University of Waterloo (work performed while at IBM)
Motivating Example

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, c.ownerid = o.id 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;

2 hours and 20 minutes

50 seconds
**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)
    
    \[
    \text{Make} = \text{‘Mazda’} \rightarrow \text{Model} = \text{‘Accord’}
    \]

• Applications
  - Data mining
  - Query optimization (our main focus)
Outline

• CORDS details
• Application to query optimization
• Experimental Results
• Related work
• Conclusion
Outline

• CORDS details
  - Overview
  - Enumeration
  - Correlation detection
  - Sampling
  - Dependency graphs
• Application to query optimization
• Experimental results
• Related work
• Conclusion
CORDS: Overview

• **Phase 1: Enumeration**
  - Enumerate all possible candidate column pairs
  - Apply pruning rules to limit # for Phase 2

• **Phase 2: 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 $A \rightarrow B$ from $S$
   Apply Chi-squared test
   If correlated, RETURN (“Correlated with degree of correlation = $x$”)
   else RETURN (“not correlated”)

SIGMOD 2004
CORDS: Sampling

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

- **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

\[
n \approx \frac{-16d^2 \log(p\sqrt{2\pi})^{1/2} - 8\log(p\sqrt{2\pi})}{1.69\delta^{0.858}}
\]
A Fixed Sample Size is OK

Required Sample Size

Maximum Allowed Error Probability (p)

δ = 0.005

10 x 10

50 x 50

SIGMOD 2004
CORDS: Dependency Graph

SYNTHETIC DATABASE
SAMPLE SIZE: 8000

SIGMOD 2004
CORDS: Dependency Graph
(across tables)

SYNTHETIC DATABASE
SAMPLE SIZE: 10000

SIGMOD 2004
Outline

• CORDS details
• Application to query optimization
• Experimental Results
• Related work
• Conclusion
Column Group Stats (**CGS**)

- Statistics about a group of columns

- Relatively easy to compute
  - Concatenate columns
  - Then obtain “usual” statistics

- Ex.: For two columns A and B
  - |A,B| = # of distinct (a,b) values
Using 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 $= |\text{Make}| \times |\text{Model}| / |\text{Make,Model}|$
  - Selectivity($P_A \text{ AND } P_B$) = $1 / |\text{Make,Model}|$
- 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 \textit{distribution} statistics
CORDS for Query Optimization

Column Group Statistics

Catalog Info

Data

Sample

Dependency Discovery

Dependency Graph

(Optional)

Optimizer

Recommend CGS

Stats Collection

SIGMOD 2004
CORDS: Recommending CGS

- Rank Soft FD’s by their Strength
- Rank correlation by their degree of correlation
  - Mean-square contingency or p-value
- Break ties using the adjustment factor:
  \[ \text{Adjustment factor} = \frac{|A| \times |B|}{|A,B|} \]
- Can rank by adjustment factor
CORDS: Recommending CGS

Census Data (2000 samples)
2065 Discovered Correlations

Number of CGS Recommended

Adj. Factor Threshold

P-value Threshold
**CORDS: Recommending CGS**

**Census Data (2000 samples)**
114 Soft Fds Discovered

<table>
<thead>
<tr>
<th>Adj. Factor Threshold</th>
<th>Number of Recommended FDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>6</td>
<td>40</td>
</tr>
<tr>
<td>0.99</td>
<td>60</td>
</tr>
<tr>
<td>0.98</td>
<td>80</td>
</tr>
<tr>
<td>0.96</td>
<td>100</td>
</tr>
<tr>
<td>0.94</td>
<td>120</td>
</tr>
<tr>
<td>0.92</td>
<td>120</td>
</tr>
<tr>
<td>0.9</td>
<td>120</td>
</tr>
</tbody>
</table>

*SIGMOD 2004*
Outline

- CORDS details
- Application to query optimization
- Experimental Results
- Related work
- Conclusion
Experiments (Performance)

[Graphs showing time in seconds for different scenarios: IND and CGS, with and without CGS. The graph on the right shows a regression line indicating improvement with CGS.]
Experiments (Accuracy vs. Time)
Experiments (Diminishing Return)

![Bar chart showing worst-case error factor for different orders of CG Statistics]
Related Work (Ours)

• **Query Driven (LEO)**
  - Compare the actual selectivity to the estimated (adjustment factor)
  - Identify groups with large adjustments
  - Limited to columns in workload
  - "Learning" can take time (lack of robustness)

• **Data-driven (B-HUNT)**
  - Look at the data
  - Identify columns with algebraic constraints
  - Rewrite query to exploit the algebraic constraints
Related Work (Others)

• 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
Outline

• CORDS details
• Application to query optimization
• Experimental Results
• Related work
• Conclusion
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?
Backup Slides
Mean-Square Contingency

- Measures statistical dependence between columns A and B:

\[
\phi^2 = \frac{1}{\min(d_A, d_B) - 1} \sum_{i=1}^{d_A} \sum_{j=1}^{d_B} \frac{(\pi_{ij} - \pi_{ig} \pi_{gj})^2}{\pi_{ig} \pi_{gj}}
\]

- Where

- \(d_X\) = (bucketized) domain size for column \(X\) \((X = A, B)\)

- \(\pi_{ij}\) = fraction of \((a, b)\) pairs with \(a = i\) and \(b = j\)

- \(\pi_{ig}\) = \(\sum_j \pi_{ij}\) and \(\pi_{gj}\) = \(\sum_i \pi_{ij}\)

- Properties
- \(0 \leq \phi^2 \leq 1\)
- \(\phi^2 = 0\): independence
- \(\phi^2 = 1\): hard FD
Chi-Squared Test

- Consider special case: \( d_A = d_B = d \)
- Idea: declare correlation if estimated value of \( n (d-1) \varphi^2 \) is "large"
- Estimate by
  \[
  \chi^2 = \sum_{i=1}^{d_A} \sum_{j=1}^{d_B} \frac{(n_{ij} - n_{ig} n_{gj})^2}{n_{ig} n_{gj}}
  \]
- If true independence (\( \varphi^2 \leq \delta \))
  - \( \chi^2 \) has \( \approx \) chi-squared distribution with \( v = (d-1)^2 \) "degrees of freedom"
- \( p \)-value for observed value \( \chi^2 = u \)
  - \( p \)-value = \( \Pr(\chi^2 \geq u | \text{independence}) \)
- Reject independence if \( p \)-value < \( p_{\text{min}} \) (or \( \chi^2 > u_{\text{max}} \))
  - I.e., reject if independence is too unlikely
- Requirement: not too many small or zero \( n_{ij} \) values