CORDS: Automatic Discovery of Correlations and Soft Functional Dependencies

I. F. Ilyas, V. Markl, P. Haas, P. Brown and A. Aboulnaga

SIGMOD 2004

UNIVERSITY OF WATERLOO

uwaterloo.ca

Presenter: Nabiha Asghar

.

Outline

- Introduction & Motivation
- Main contributions of the paper
- Description of algorithm and techniques
- Experimental results

Intro & Motivation

- Why is it important to discover correlations and functional dependencies in the data?

UNIVERSITY OF WATERLOO

Intro & Motivation

- Why is it important to discover correlations and functional dependencies in the data?

QUERY OPTIMIZATION

UNIVERSITY OF WATERLOO

Intro & Motivation

Query optimizers:

- need to estimate the cost of different access methods (e.g. the optimal join order)
- need to compute 'selectivity' of predicates
- 'selectivity' of a predicate *p* = fraction of rows in a table that are chosen by *p*

UNIVERSITY OF WATERLOO

Example: Selectivity of a Predicate

FirstName	LastName
Emma	Stewart
Chris	Martin
Jennifer	Jackson
Emma	Johnson
Liam	Watson

sel("FirstName = Emma") = 2/5

sel("LastName = Watson") = 1/5

Intro & Motivation

Query optimizers:

- need to estimate the cost of different access methods (e.g. the optimal join order)
- need to compute ‘selectivity’ of predicates

Intro & Motivation

Query optimizers:

- need to estimate the cost of different access methods (e.g. the optimal join order)
- need to compute 'selectivity' of predicates
- example: joining the columns $R.C_1$ and $S.C_2$ over the predicate $R.C_1 = \text{'a'}$ AND $S.C_2 = \text{'b'}$
- typically estimate this selectivity as

$$sel(R.C_1 = \text{'a'}) \times sel(S.C_2 = \text{'b'})$$

UNIVERSITY OF WATERLOO

Intro & Motivation

Query optimizers:

- Assume *sel*(p_1 AND p_2) = *sel*(p_1) * *sel*(p_2)
- This assumption does not hold if the two columns are dependent/correlated

Gist: we need to figure out such dependencies to enable more efficient query execution plans

UNIVERSITY OF WATERLOO

Main Contributions

- CORDS Algorithm
  - detects functional dependencies (FDs) of the form $X \rightarrow Y$
  - detects soft FDs of the form $X \Longrightarrow Y$ (the value of X determines the value of Y with a high probability)
  - scalable, efficient
- Experimental Evaluation

UNIVERSITY OF WATERLOO

Examples of FDs & Soft FDs

Name	SIN	City
Emma Stewart	123456	Toronto
Jack Hugh	456789	Boston
Jennifer Li	567890	LA
Liam Yang	364566	Miami
Sandra B.	871235	Austin
Megan Ray	639123	Seattle
Jo Watson	789012	NYC
Jo Watson	901234	NYC
Jo Watson	765776	Ottawa

UNIVERSITY OF WATERLOO

Examples of FDs & Soft FDs

Name	SIN	City
Emma Stewart	123456	Toronto
Jack Hugh	456789	Boston
Jennifer Li	567890	LA
Liam Yang	364566	Miami
Sandra B.	871235	Austin
Megan Ray	639123	Seattle
Jo Watson	789012	NYC
Jo Watson	901234	NYC
Jo Watson	765776	Ottawa

FDs:

SIN → Name

SIN → City

Examples of FDs & Soft FDs

Name	SIN	City
Emma Stewart	123456	Toronto
Jack Hugh	456789	Boston
Jennifer Li	567890	LA
Liam Yang	364566	Miami
Sandra B.	871235	Austin
Megan Ray	639123	Seattle
Jo Watson	789012	NYC
Jo Watson	901234	NYC
Jo Watson	765776	Ottawa

FDs:

$\text{SIN} \rightarrow \text{Name}$

$\text{SIN} \rightarrow \text{City}$

Soft FDs:

$\text{Name} \Longrightarrow \text{City}$

UNIVERSITY OF WATERLOO

CORDS

Main Idea:

- Generate candidate column pairs (a_1, a_2) that potentially have interesting/useful dependencies
- For each candidate, the dependency is detected by computing some statistics on sampled values from the columns

UNIVERSITY OF WATERLOO

CORDS: Generating Candidates

A candidate is a triple (a_1, a_2, P), where

- a_1 and a_2 are attributes
- P is a pairing rule, which specifies which particular a_1 values get paired with which particular a_2 values to form the set of pairs of potentially correlated values.

UNIVERSITY OF WATERLOO

Example 1: Candidates

orderID	shipDate
2A5	2001-01-03
3C2	2001-04-15
3B8	2002-11-25
2E1	2002-10-31
3D6	2002-07-25
...	...

orders

orderID	deliveryDate	deliveryTime
2A5	2001-01-06	09:50
3C2	2001-04-27	13:00
3B8	2002-12-10	11:20
2E1	2002-12-02	16:10
3D6	2002-07-29	08:50
...	...	...

deliveries

UNIVERSITY OF WATERLOO

Example 1: Candidates

a_1

orderID	shipDate
2A5	2001-01-03
3C2	2001-04-15
3B8	2002-11-25
2E1	2002-10-31
3D6	2002-07-25
...	...

orders

a_2

orderID	deliveryDate	deliveryTime
2A5	2001-01-06	09:50
3C2	2001-04-27	13:00
3B8	2002-12-10	11:20
2E1	2002-12-02	16:10
3D6	2002-07-29	08:50
...	...	...

deliveries

P

A possible candidate: (orders.shipDate,
deliveries.deliveryDate,
orders.orderID = deliveries.orderID)

Example 2: Candidates

orderID	shipDate	deliveryDate
2A5	2001-01-03	2001-01-06
3C2	2001-04-15	2001-04-27
3B8	2002-11-25	2002-12-10
2E1	2002-10-31	2002-12-02
3D6	2002-07-25	2002-07-29
...	...	...

orders

Example 2: Candidates

orderID	shipDate (a_1)	deliveryDate (a_2)
2A5	2001-01-03	2001-01-06
3C2	2001-04-15	2001-04-27
3B8	2002-11-25	2002-12-10
2E1	2002-10-31	2002-12-02
3D6	2002-07-25	2002-07-29
...	...	...

orders

A possible candidate: (orders.shipDate,
orders.deliveryDate,
$\emptyset_{\text{orders}}$)

UNIVERSITY OF WATERLOO

Example 2: Candidates

orderID	shipDate (a_1)	deliveryDate (a_2)
2A5	2001-01-03	2001-01-06
3C2	2001-04-15	2001-04-27
3B8	2002-11-25	2002-12-10
2E1	2002-10-31	2002-12-02
3D6	2002-07-25	2002-07-29
...	...	...

orders

Trivial pairing rule: when the columns are in the same table and each a_1 value is paired with the a_2 value in the same row

A possible candidate: (orders.shipDate,
orders.deliveryDate,
$\emptyset_{\text{orders}}$)

UNIVERSITY OF WATERLOO

CORDS: Generating Candidates

- Generate all possible candidates having a trivial pairing rule
- Generate all possible candidates with nontrivial pairing rules which look like key-to-foreign-key join predicates
  i. First find all the declared primary/unique keys, and all the soft (*almost-unique)* keys
  ii. For each such key, examine every other column in the schema to find potential matches

UNIVERSITY OF WATERLOO

CORDS: Generating Candidates (cont'd)

To prune this huge set of candidates, use:

- Type constraints e.g. prune columns who's data type is not integer
- Statistical constraints e.g. prune columns with too few distinct values
- Pairing constraints e.g. only allow key-to-foreign-key pairing rules

etc.

UNIVERSITY OF WATERLOO

CORDS

Main Idea:

- Generate candidate column pairs (a_1, a_2) that potentially have interesting/useful dependencies ✓
- For each candidate, the dependency is detected by computing some statistics on sampled values from the candidates' columns

CORDS: Testing for Correlation

For each candidate triple C = (a_1, a_2, P):

- Draw a random sample of *n* pairs from the columns (a_1, a_2)
- Estimate $\phi^2 = \frac{1}{d-1}\sum_{i=1}^{d_1}\sum_{j=1}^{d_2}\frac{(\pi_{ij} - \pi_{i\cdot}\pi_{\cdot j})^2}{\pi_{i\cdot}\pi_{\cdot j}}$ where

d_i =no. of distinct values in column a_i

d = min (d_1, d_2)

π_{ij} = fraction of pairs where a_1 = i and a_2 = j

$\pi_{i\cdot} = \sum_j \pi_{ij}$

$\pi_{\cdot j} = \sum_i \pi_{ij}$

- $\phi^2 = 1$ corresponds to an FD
- $\phi^2 \leq \epsilon$ corresponds to independence for a small $\epsilon > 0$

UNIVERSITY OF WATERLOO

CORDS

Main Idea:

- Generate candidate column pairs (a_1, a_2) that potentially have interesting/useful dependencies ✓
- For each candidate, the dependency is detected by computing some statistics on sampled values from the candidates' columns ✓

UNIVERSITY OF WATERLOO

ALGORITHM *DetectCorrelation*
INPUT : A column pair C_1, C_2 with $|C_1|_R \geq |C_2|_R$

Discover Trivial Cases
1. a. IF $|C_i|_R \geq (1 - \epsilon_1)|R|$ for $i = 1$ or $i = 2$
   THEN C_i is a soft key; RETURN.
   b. IF $|C_i|_R = 1$ for $i = 1$ or $i = 2$
   THEN C_i is a trivial column; RETURN.

Sampling
2. Sample R to produce a reduced table S.

Detect Soft Functional Dependencies in the Sample
3. a. Query S to get $|C_1|_S$, $|C_2|_S$ and $|C_1, C_2|_S$.
   b. IF $|C_1, C_2|_S \leq \epsilon_2 |S|$
      AND $|C_1|_S \geq (1 - \epsilon_3)|C_1, C_2|_S$
   THEN $C_1 \Rightarrow C_2$; RETURN.

Skew Handling for Chi-Squared Test
4. FOR $i = 1, 2$
   a. IF $\sum_{j=1}^{N_i} F_{ij} \geq (1 - \epsilon_4)|R|$
   THEN
      SKEW$_i$ = TRUE;
      $d_i = N_i$;
      FILTER = "C_i IN $\{V_{i1}, \ldots, V_{iN_i}\}$"
   ELSE
      SKEW$_i$ = FALSE;
      $d_i = \min(|C_i|_R, d_{\max})$;
      FILTER = NULL.
   b. Apply FILTER.

Sampling-Based Chi-Squared Test
5. a. Initialize each n_{ij}, $n_{i\cdot}$, and $n_{\cdot j}$ to 0.
   b. FOR EACH column-value pair (x_1, x_2)
      $i = \mathit{Category}(1, x_1, d_1, \text{SKEW}_1)$;
      $j = \mathit{Category}(2, x_2, d_2, \text{SKEW}_2)$;
      Increment n_{ij}, $n_{i\cdot}$, and $n_{\cdot j}$ by 1;
   c. IF $\sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \mathit{IsZero}(n_{ij}) > \epsilon_5 d_1 d_2$
   THEN C_1 and C_2 are correlated; RETURN.
   d. Compute χ^2 as in (1); set $\nu = (d_1 - 1)(d_2 - 1)$
   and $t = G_\nu^{-1}(1 - p)$.
   e. IF $\chi^2 > t$
   THEN C_1 and C_2 are correlated; RETURN.
   ELSE C_1 and C_2 are independent; RETURN.

UNIVERSITY OF WATERLOO

Discover Trivial Cases

1\. a. IF $|C_i|_R \geq (1-\epsilon_1)|R|$ for $i=1$ or $i=2$
THEN C_i is a soft key; RETURN.
b. IF $|C_i|_R = 1$ for $i=1$ or $i=2$
THEN C_i is a trivial column; RETURN.

ALGORITHM *DetectCorrelation*
INPUT : A column pair C_1, C_2 with $|C_1|_R \geq |C_2|_R$

Discover Trivial Cases

1\. a. IF $|C_i|_R \geq (1-\epsilon_1)|R|$ for $i=1$ or $i=2$
THEN C_i is a soft key; RETURN.
b. IF $|C_i|_R = 1$ for $i=1$ or $i=2$
THEN C_i is a trivial column; RETURN.

Sampling

2\. Sample R to produce a reduced table S.

Detect Soft Functional Dependencies in the Sample

3\. a. Query S to get $|C_1|_S$, $|C_2|_S$ and $|C_1, C_2|_S$.
b. IF $|C_1, C_2|_S \leq \epsilon_2|S|$
AND $|C_1|_S \geq (1-\epsilon_3)|C_1, C_2|_S$
THEN $C_1 \Rightarrow C_2$; RETURN.

Skew Handling for Chi-Squared Test

4\. FOR $i = 1, 2$
a. IF $\sum_{j=1}^{N_i} F_{ij} \geq (1-\epsilon_4)|R|$
THEN
SKEW$_i$ = TRUE;
$d_i = N_i$;
FILTER = "C_i IN $\{V_{i1}, \ldots, V_{iN_i}\}$"
ELSE
SKEW$_i$ = FALSE;
$d_i = \min(|C_i|_R, d_{\max})$;
FILTER = NULL.
b. Apply FILTER.

Sampling-Based Chi-Squared Test

5\. a. Initialize each n_{ij}, $n_{i.}$, and $n_{.j}$ to 0.
b. FOR EACH column-value pair (x_1, x_2)
$i = \mathit{Category}(1, x_1, d_1, \text{SKEW}_1)$;
$j = \mathit{Category}(2, x_2, d_2, \text{SKEW}_2)$;
Increment n_{ij}, $n_{i.}$, and $n_{.j}$ by 1;
c. IF $\sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \mathit{IsZero}(n_{ij}) > \epsilon_5 d_1 d_2$
THEN C_1 and C_2 are correlated; RETURN.
d. Compute χ^2 as in (1); set $\nu = (d_1 - 1)(d_2 - 1)$
and $t = G_\nu^{-1}(1-p)$.
e. IF $\chi^2 > t$
THEN C_1 and C_2 are correlated; RETURN.
ELSE C_1 and C_2 are independent; RETURN.

Sampling

2. Sample R to produce a reduced table S.

ALGORITHM *DetectCorrelation*
INPUT : A column pair C_1, C_2 **with** $|C_1|_R \geq |C_2|_R$

Discover Trivial Cases

1. a. IF $|C_i|_R \geq (1 - \epsilon_1)|R|$ for $i = 1$ or $i = 2$
   THEN C_i is a soft key; RETURN.
   b. IF $|C_i|_R = 1$ for $i = 1$ or $i = 2$
   THEN C_i is a trivial column; RETURN.

Sampling

2. Sample R to produce a reduced table S.

Detect Soft Functional Dependencies in the Sample

3. a. Query S to get $|C_1|_S$, $|C_2|_S$ and $|C_1, C_2|_S$.
   b. IF $|C_1, C_2|_S \leq \epsilon_2|S|$
   AND $|C_1|_S \geq (1 - \epsilon_3)|C_1, C_2|_S$
   THEN $C_1 \Rightarrow C_2$; RETURN.

Skew Handling for Chi-Squared Test

4. FOR $i = 1, 2$
   a. IF $\sum_{j=1}^{N_i} F_{ij} \geq (1 - \epsilon_4)|R|$
   THEN
   SKEW$_i$ = TRUE;
   $d_i = N_i$;
   FILTER = "C_i IN $\{V_{i1}, \ldots, V_{iN_i}\}$"
   ELSE
   SKEW$_i$ = FALSE;
   $d_i = \min(|C_i|_R, d_{\max})$;
   FILTER = NULL.
   b. Apply FILTER.

Sampling-Based Chi-Squared Test

5. a. Initialize each n_{ij}, $n_{i.}$, and $n_{.j}$ to 0.
   b. FOR EACH column-value pair (x_1, x_2)
   $i = Category(1, x_1, d_1, \text{SKEW}_1)$;
   $j = Category(2, x_2, d_2, \text{SKEW}_2)$;
   Increment n_{ij}, $n_{i.}$, and $n_{.j}$ by 1;
   c. IF $\sum_{i=1}^{d_1} \sum_{j=1}^{d_2} IsZero(n_{ij}) > \epsilon_5 d_1 d_2$
   THEN C_1 and C_2 are correlated; RETURN.
   d. Compute χ^2 as in (1); set $\nu = (d_1 - 1)(d_2 - 1)$
   and $t = G_\nu^{-1}(1 - p)$.
   e. IF $\chi^2 > t$
   THEN C_1 and C_2 are correlated; RETURN.
   ELSE C_1 and C_2 are independent; RETURN.

Detect Soft Functional Dependencies in the Sample

3. a. Query S to get $|C_1|_S$, $|C_2|_S$ and $|C_1, C_2|_S$.
   b. IF $|C_1, C_2|_S \le \epsilon_2 |S|$
      AND $|C_1|_S \ge (1 - \epsilon_3)|C_1, C_2|_S$
      THEN $C_1 \Rightarrow C_2$; RETURN.

ALGORITHM ***DetectCorrelation***
INPUT : A column pair C_1, C_2 with $|C_1|_R \ge |C_2|_R$

Discover Trivial Cases

1. a. IF $|C_i|_R \ge (1 - \epsilon_1)|R|$ for $i = 1$ or $i = 2$
      THEN C_i is a soft key; RETURN.
   b. IF $|C_i|_R = 1$ for $i = 1$ or $i = 2$
      THEN C_i is a trivial column; RETURN.

Sampling

2. Sample R to produce a reduced table S.

Detect Soft Functional Dependencies in the Sample

3. a. Query S to get $|C_1|_S$, $|C_2|_S$ and $|C_1, C_2|_S$.
   b. IF $|C_1, C_2|_S \le \epsilon_2 |S|$
      AND $|C_1|_S \ge (1 - \epsilon_3)|C_1, C_2|_S$
      THEN $C_1 \Rightarrow C_2$; RETURN.

Skew Handling for Chi-Squared Test

4. FOR $i = 1, 2$
   a. IF $\sum_{j=1}^{N_i} F_{ij} \ge (1 - \epsilon_4)|R|$
      THEN
        SKEW$_i$ = TRUE;
        $d_i = N_i$;
        FILTER = "C_i IN $\{V_{i1}, \ldots, V_{iN_i}\}$"
      ELSE
        SKEW$_i$ = FALSE;
        $d_i = \min(|C_i|_R, d_{\max})$;
        FILTER = NULL.
   b. Apply FILTER.

Sampling-Based Chi-Squared Test

5. a. Initialize each n_{ij}, $n_{i\cdot}$, and $n_{\cdot j}$ to 0.
   b. FOR EACH column-value pair (x_1, x_2)
        $i = \textit{Category}(1, x_1, d_1, \text{SKEW}_1)$;
        $j = \textit{Category}(2, x_2, d_2, \text{SKEW}_2)$;
        Increment n_{ij}, $n_{i\cdot}$, and $n_{\cdot j}$ by 1;
   c. IF $\sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \textit{IsZero}(n_{ij}) > \epsilon_5 d_1 d_2$
      THEN C_1 and C_2 are correlated; RETURN.
   d. Compute χ^2 as in (1); set $\nu = (d_1 - 1)(d_2 - 1)$
      and $t = G_\nu^{-1}(1 - p)$.
   e. IF $\chi^2 > t$
      THEN C_1 and C_2 are correlated; RETURN.
      ELSE C_1 and C_2 are independent; RETURN.

UNIVERSITY OF WATERLOO

Helping the Optimizer

ALGORITHM *RecommendCGS*
INPUT: Discovered correlations and soft FDs

1. Sort correlated pairs, (C_i, C_j) in ascending order of p-value
2. Sort soft FDs in descending order of estimated strength
3. Break ties by sorting in descending order of the adjustment factor $|C_i||C_j|/|C_i, C_j|$.
4. Recommend the top k_1 correlated column pairs and the top k_2 soft FDs to the optimizer

Figure 5: Ranking Correlations and Soft FDs.

UNIVERSITY OF WATERLOO

Experimental Evaluation

- Created a schema with 4 relations: CAR, OWNER, DEMOGRAPHICS and ACCIDENTS (size 1GB)

- Noted FDs, soft FDs, correlations

- CORDS, using a sample size of 4000 rows, detected all correlations and soft FDs, and did not incorrectly detect any spurious relationships

UNIVERSITY OF WATERLOO

Experimental Evaluation

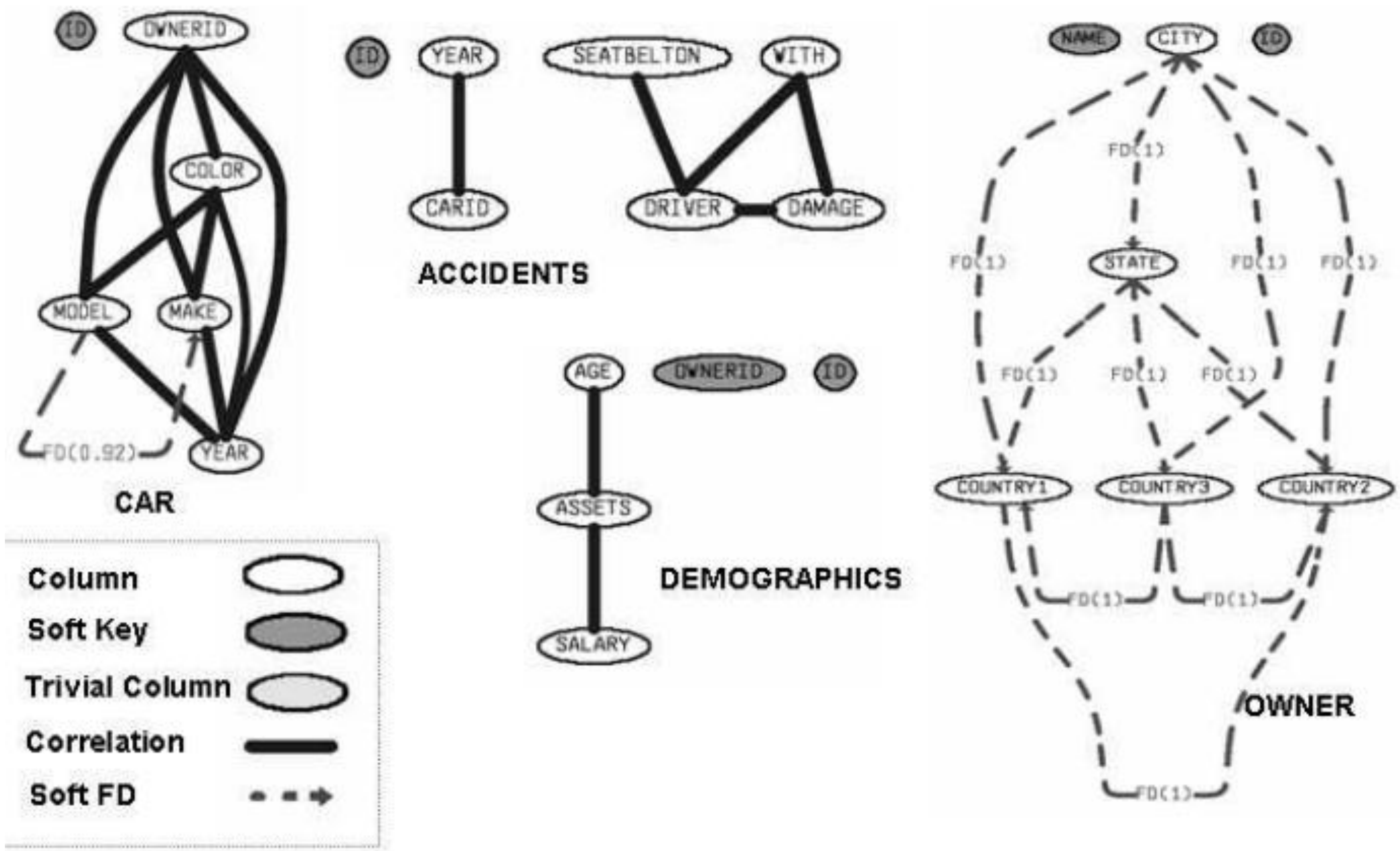


Figure 6: Dependency graph for the *Accidents* database.

UNIVERSITY OF WATERLOO

Experimental Evaluation

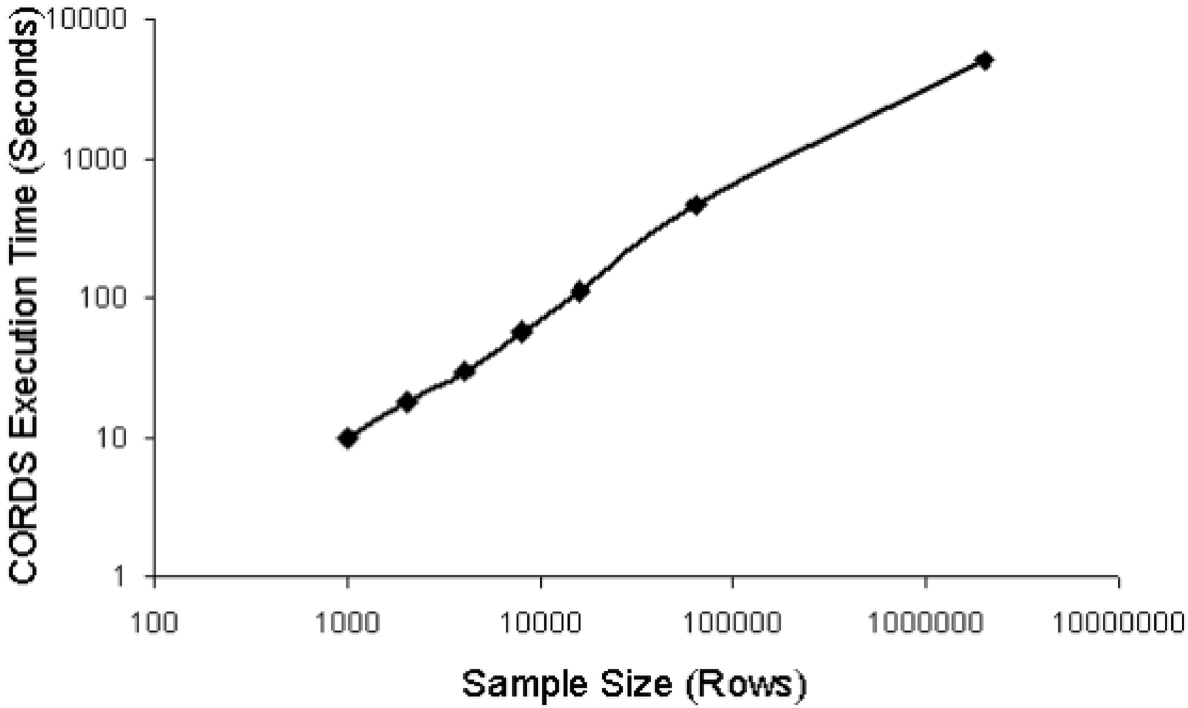


Figure 7: Effect of sample size on execution time.

UNIVERSITY OF WATERLOO

Experimental Evaluation

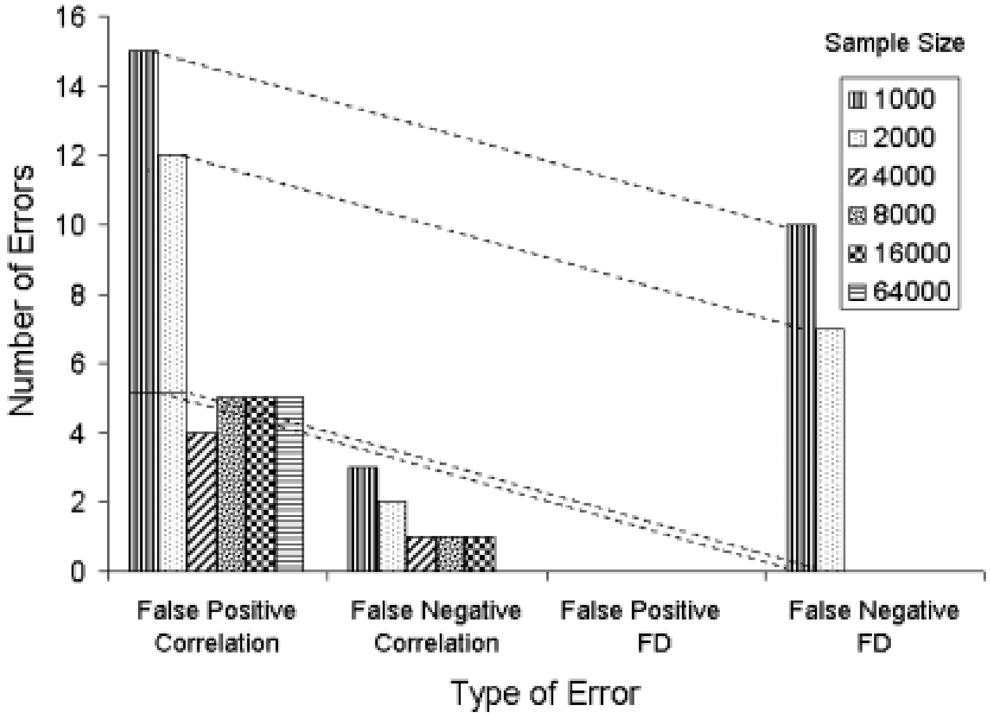


Figure 8: Accuracy versus sample size.

UNIVERSITY OF WATERLOO

Experimental Evaluation

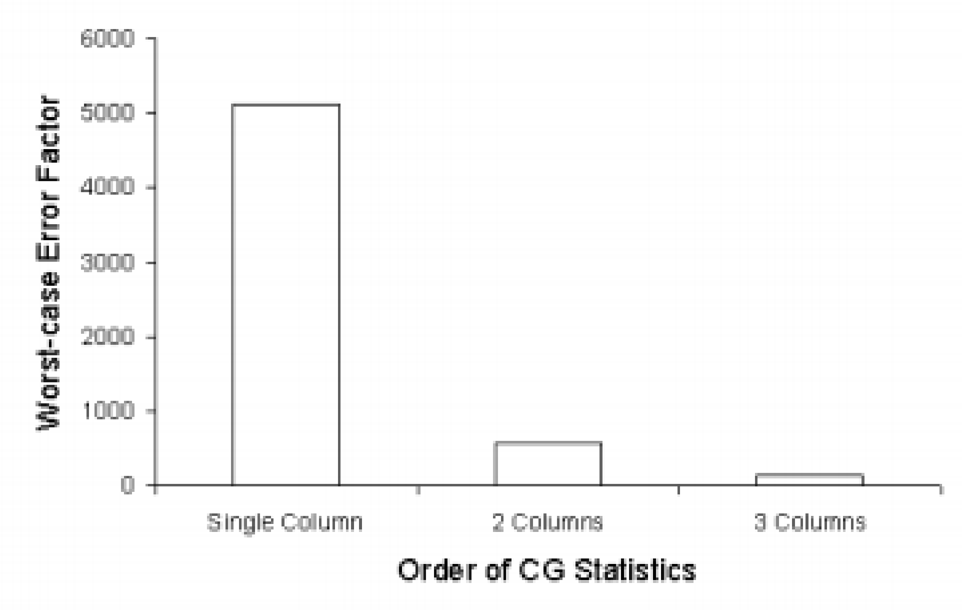
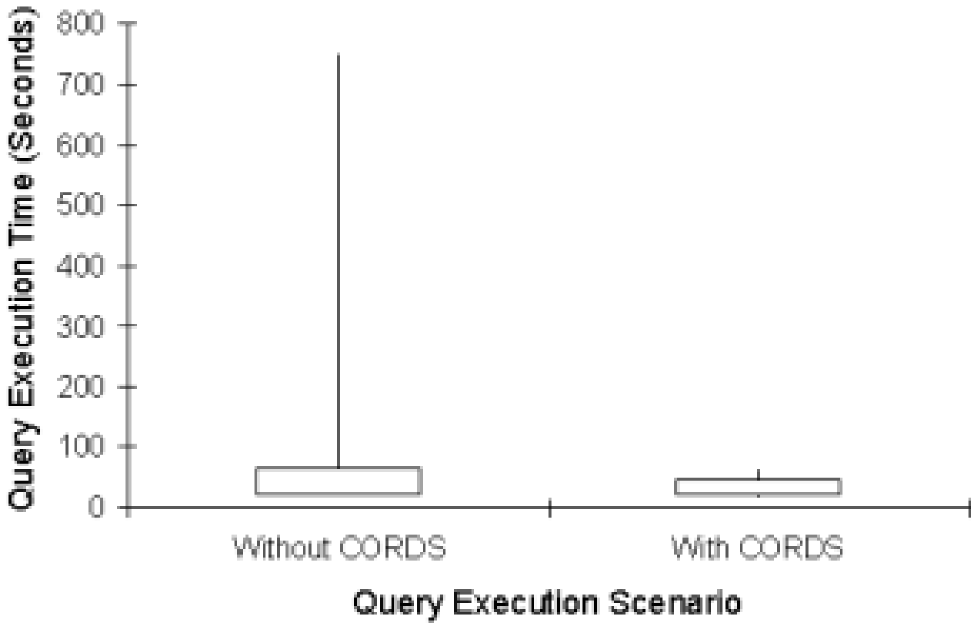


Figure 9: Effect of column-group order on accuracy of selectivity estimates.

Experimental Evaluation



(a) Box Plot

Figure 10: Effect of CORDS on query execution time.

UNIVERSITY OF WATERLOO

Summary

- CORDS automatically discovers correlations and soft FDs in data
- Works on samples instead of the entire data, so very scalable
- Improves selectivity estimation for query optimization, therefore better query plans are generated
- Efficient, because it works on pairs of columns only