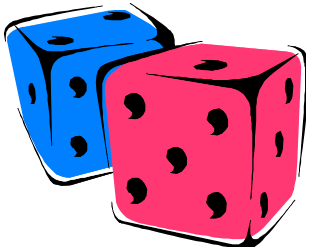
A Bi-Level Bernoulli Scheme for Database Sampling



Peter J. Haas & Christian König
IBM Almaden Research Center
San Jose, CA

not

Motivation: ISO Sampling Queries

- Users are demanding database sampling
  - Quick approximate answers to ad hoc (aggregation) queries
  - Traditionally, only inefficient "simulated" sampling available:

```
SELECT * FROM T WHERE RAND() < 0.01
```

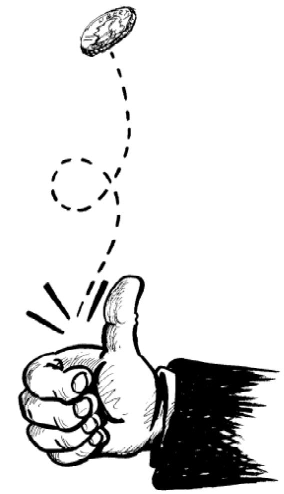
- Proposed ISO SQL sampling standard

TABLESAMPLE *samplingMethod* (*samplingPercent*)

- Currently supported sampling methods:
  - BERNOULLI: row-level "coin flip" sampling
  - SYSTEM: vendor-defined sampling method
    - Page-level Bernoulli sampling

Row-Level Bernoulli Sampling

- Include ith row independently with probability = q
- Example:

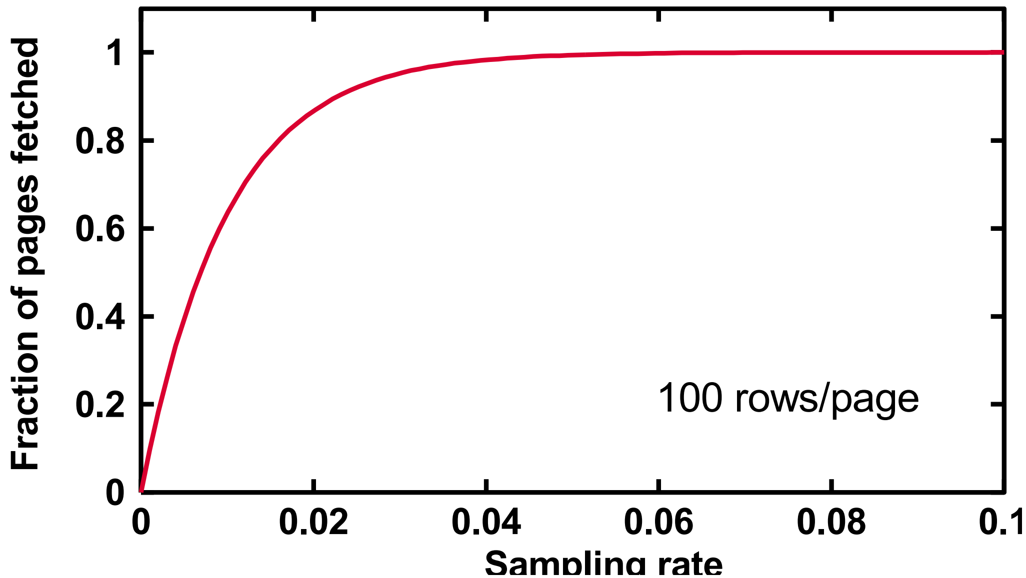


```
SELECT SUM(trans.amount)/0.05 FROM trans
TABLESAMPLE BERNOULLI(100 * 0.05)
WHERE predicate
```

- Sample size is <u>random</u> (100q% of rows on average)
- Easy to parallelize
- Implementation tricks (see paper)
  - Can exploit indexes to save I/Os
  - Can "pre-simulate" coin tosses to save I/Os

Problem: High I/O Costs Persist

- Naïve implementation: fetch every page
- Best possible implementation:



Page-Level Bernoulli Sampling

- Include ith <u>page</u> independently with probability q
  - Include all rows on page in sample
- Much lower I/O costs than row-level Bernoulli
  - Cost ≈ q x (cost of full tablescan)
- Implementation tricks (see paper)
  - Pre-generate geometric page skips
  - Exploit prefetch

Problem: Low Precision

- Higher standard errors than with row-level sampling
  - Sometimes by an order of magnitude

- Two causes (e.g., when estimating SUM by sampling)
  - Random-sample-size effect:

20 values each = 5, q = 0.5   Sample = 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5   Estimate = 2 x 60 = 120

Too large   Too high

  - Clustering effect:

$10^6, 10^6, 10^6, 10^6, 10^6, 10^6$ vs 0, 0, 0, 0, 0, 0, 0, 0, 0 in sample

Overall Problem: Lack of Control

- Can't trade off speed and accuracy in a good way
  - Row-level sampling often too slow
  - Page-level sampling fast, but answer may well be too imprecise
    - Painful trial-and-error search for sample size
  - Currently, user receives no guidance

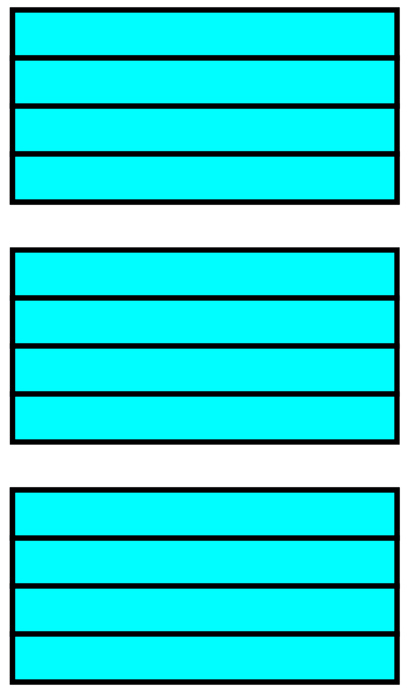
- Our proposed solution:
  - Bi-level Bernoulli sampling: permits <u>spectrum</u> of trade-offs
  - Techniques for automatically finding <u>best</u> (or good) trade-off
  - ➔ Better implementation of ISO sampling clause

TABLESAMPLE SYSTEM(*q*)

Bi-Level Bernoulli Sampling

- For overall sampling rate q
  - Select pages using Bernoulli sampling with rate p ($\geq q$)
  - For each sampled page, take Bernoulli sample with rate $r = q/p$

$$q = p\, r$$



Bi-Level Sampling, Continued

- Previous schemes obtained as special cases
  - $p = q$: pure page-level sampling ($r = 1$ since $p\,r = q$)
  - $p = 1$: pure row-level sampling ($r = q$)
  - … with a spectrum of sampling schemes in between
- Why not use all rows on page?
  - CPU cost issues (cleansing, transformation, expensive operations)
  - Upper bound on sample size (e.g., as in ISO queries)
- Implementation
  - Combine row-level and page-level techniques

Overview of Remainder of Talk

- Focus on aggregation queries

SELECT *op* (*expression*) FROM T
WHERE *predicate*

  - Where op $\in$ {SUM, COUNT, AVG}

- Query optimization: we will derive optimal *p* and *r* values
  - Some new and unexpected results
  - Pilot sampling required

- Heuristic optimization method
  - Avoids pilot sampling
  - Uses catalog statistics
  - Experimental comparison with optimal solutions

Estimates and Their Precision

- To estimate an aggregate
  - SUM: scale sample sum by $1/q$
  - COUNT: special case of SUM
  - AVERAGE: SUM / COUNT

Unbiased estimates

- Precision of approximate aggregates
  - General form of variance = (standard error)^2:

$$v(p,r) \approx \left(\frac{1}{p}-1\right)a + \frac{1}{p}\left(\frac{1}{r}-1\right)b$$

Computational formulas depend on specific aggregate

- a = between-page variability
- b = within-page variability
- Page heterogeneity index: PHI = b / a

Optimal Bi-Level Sampling

- Cost models
  - <u>I/O cost model</u>: $C = C(p)$ increasing
    - Ex: $C(p)$ = expected sampling cost

$$C(p) = c\,|U|\,p$$

    - Ex: $C(p)$ = Probability that sampling cost exceeds theshhold t

$$C(p) = \Pr\{c\,|U^{(S)}| > t\} = \sum_{k=\lceil x/c \rceil}^{|U|} \binom{|U|}{k} p^k (1-p)^{|U|-k}$$

  - <u>Additive cost model</u>: $C = C(p,q) = C_1(p) + C_2(q)$

$|U|$ = # pages in table
$|T|$ = # rows in table
$|U^{(S)}|$ = # pages in sample

- Optimization problem:
  - Minimize standard error, given q and maximum allowable cost c_{max}
  - Other problems considered in paper

Optimal Solutions

	PHI > 1	PHI ≤ 1
I/O Cost Model	$p^* = q$ $r^* = 1$	$p^* = p_{max}$ $r^* = q / p_{max}$
Additive Cost Model	$p^* = q$ $r^* = 1$	$p^* = \min\left(C_1^{-1}\left(c_{max} - C_2(q)\right), 1\right)$ $r^* = q / p^*$
	(page-level sampling)	(row-level sampling)

- Features of optimal solution
  - Supports (and quantifies) intuition
  - “Bang-bang” solution that depends on value of PHI
  - When PHI ≤ 1, need full generality of bi-level sampling

Other Optimality Results (in Paper)

- Estimating the PHI
  - Pilot sampling
- Other aggregates besides SUM, COUNT, AVG
- Multiple aggregates in SELECT statement

```
SELECT SUM(C1*C2), SUM(C3), AVG(C4/C2)
```

A Heuristic Sampling Method

- Goal: avoid need for a pilot sample to estimate PHI
- Initially assume one column appears in SELECT list:

```
SELECT SUM(col1) / q
FROM t TABLESAMPLE SYSTEM(100*q)
WHERE predicate
```

- Based on four simple catalog statistics
  - $\delta =$ average # of distinct values per page
  - $\rho =$ avg # rows per page
  - $\gamma^{(1)} = \mathrm{var}(a_1, \ldots, a_M)$ — $a_i =$ average of col1 values on page i
  - $\gamma^{(2)} = \mathrm{avg}(v_1, \ldots, v_M)$ — $v_i =$ variance of col1 values on page i

Heuristic Scheme, Continued

- Intuition:
  - Many distinct values => page-level sampling;
    Few distinct values => row-level sampling
  - Unless DVs are close to each other (then use row-level sampling)
  - Measure closeness by $\gamma = \gamma^{(2)} / \gamma^{(1)}$
- Target fraction of DVs to sample from a page:

$$f(\gamma,\delta) = 1 + \left(\frac{1}{1+\gamma}\right)\left(\frac{1}{\delta} - 1\right)$$

- Expected # DVs per page @ row-level rate *r* (Cardenas):

$$E[D] = \delta\left(1 - (1-r)^{\rho/\delta}\right)$$

Heuristic Scheme: Final Results

- Set $E[D]/\delta = f(\gamma,\delta)$ and solve for r to get

$$r_0^* = 1-(1-f(\gamma,\delta))^{\delta/\rho}$$

- Final solution: $r^* = \max(r_0^*, q)$ and $p^* = q/r^*$

- Constraint on processing cost: set $r^* \leftarrow \max(r^*, q/p_{\max})$

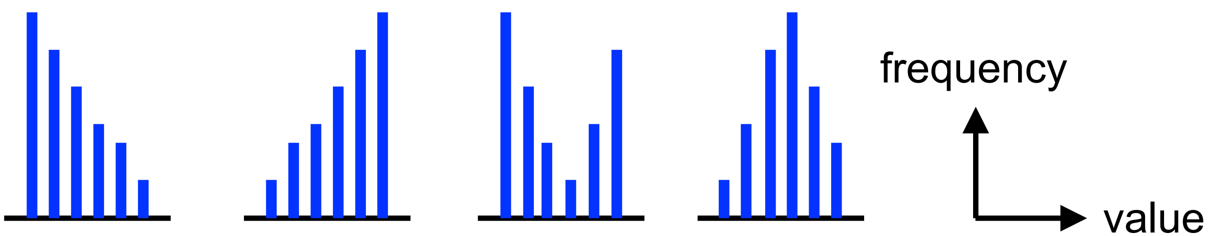
- K columns in SELECT list:

```
SELECT COUNT(col3), SUM(col1/col2) / AVG(col1*col3)
```

Set $p_{\text{comb}} = (p_1 p_2 \cdots p_K)^{1/K}$ and $r_{\text{comb}} = (r_1 r_2 \cdots r_K)^{1/K}$

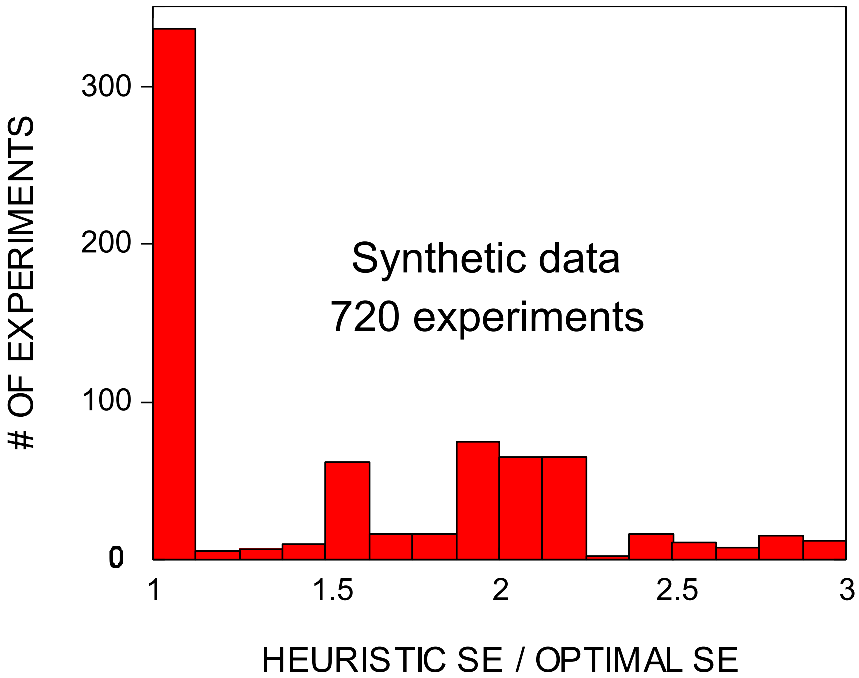
Experimental Evaluation of Heuristic

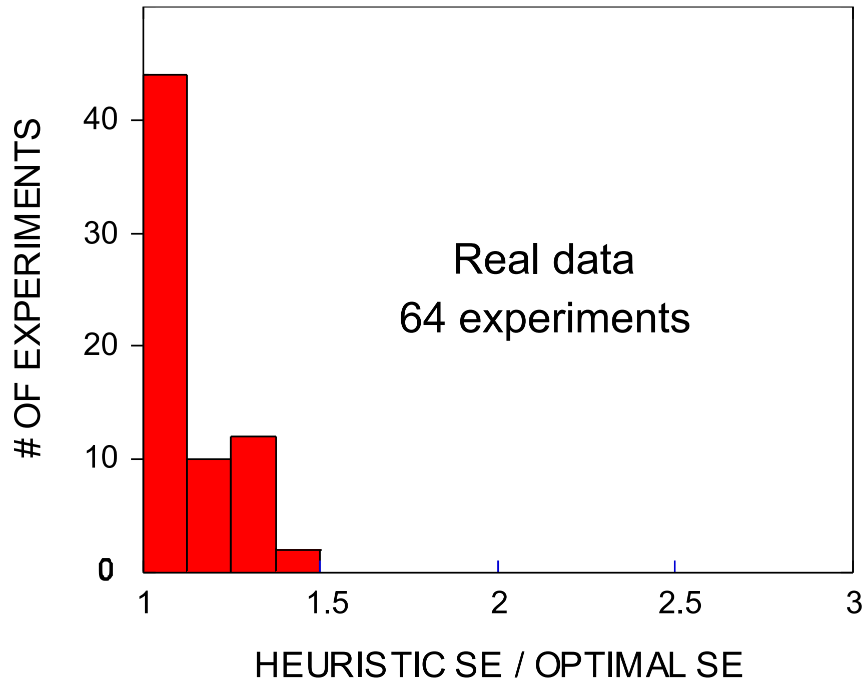
- P1: minimize SE such that *p r* = *q* and cost ≤ C_{max}
- SUM and AVG queries
- One column or two columns in SELECT list
- 324 synthetic tables
  - 10^5 rows, 150 rows per page
  - Varied: # DVs, clustering, data range, Zipfian skew, mode



- Two real-world data sets
  - 8 Gb of automotive data
  - 100 years of baseball stats

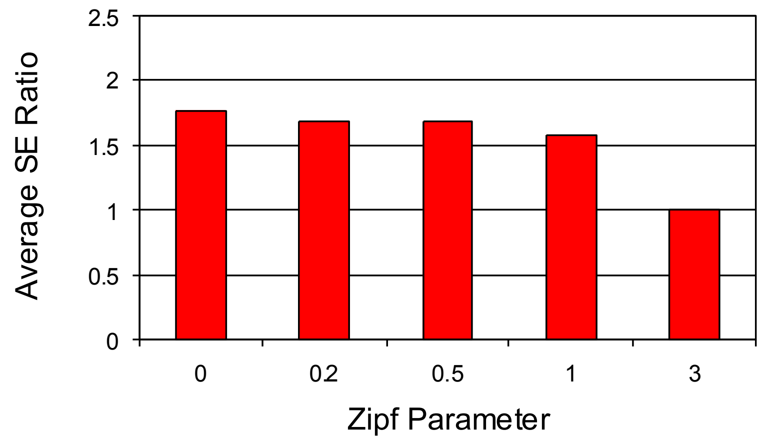
Single-Column Queries

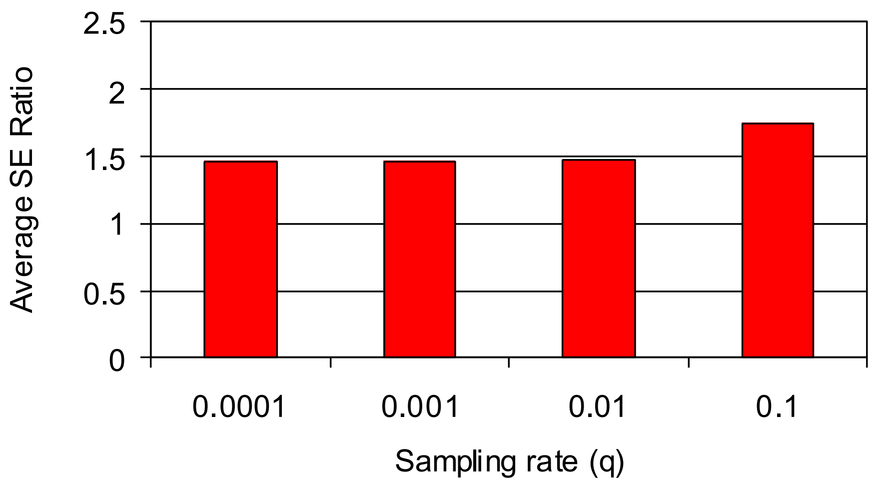




Optimal results in 47% of cases for synthetic data (median = 1.54)
Optimal results in 56% of cases for real-world data

Effect of Clustering, Skew, and *q*

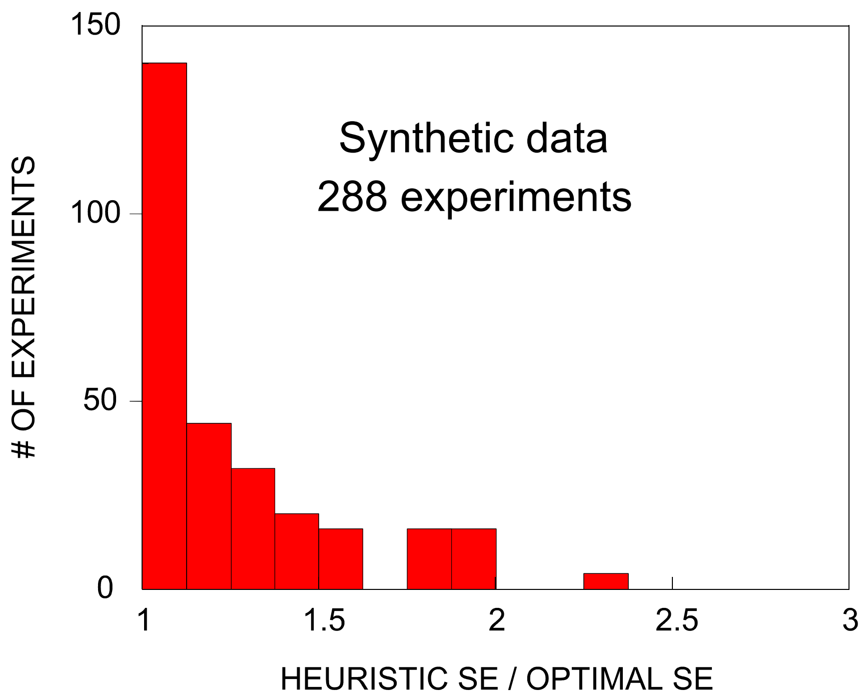




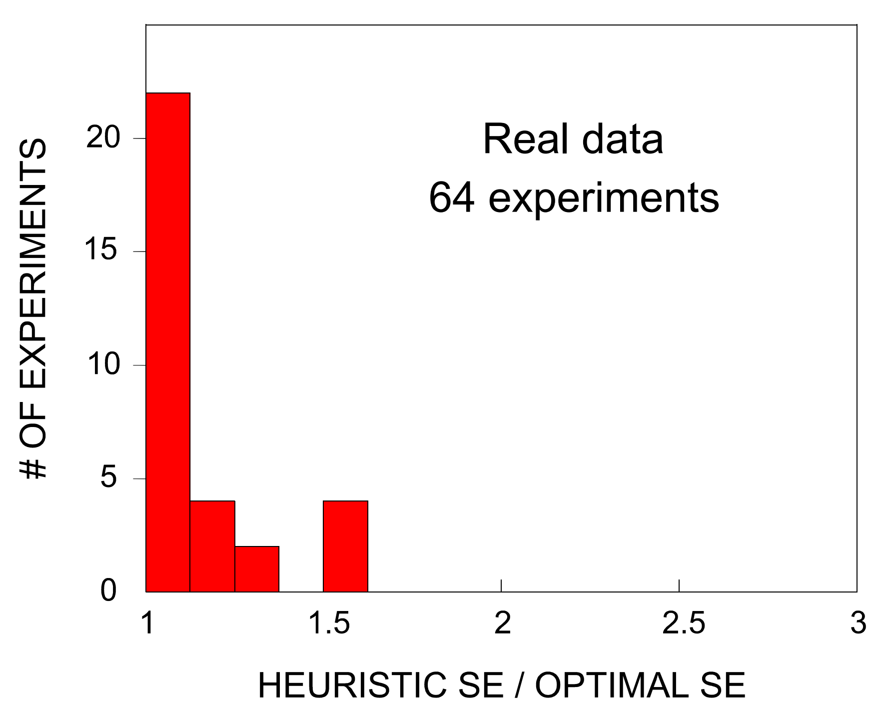
Heuristic works best
In hardest cases!

Two-Column Queries

SELECT SUM(col1*col2)

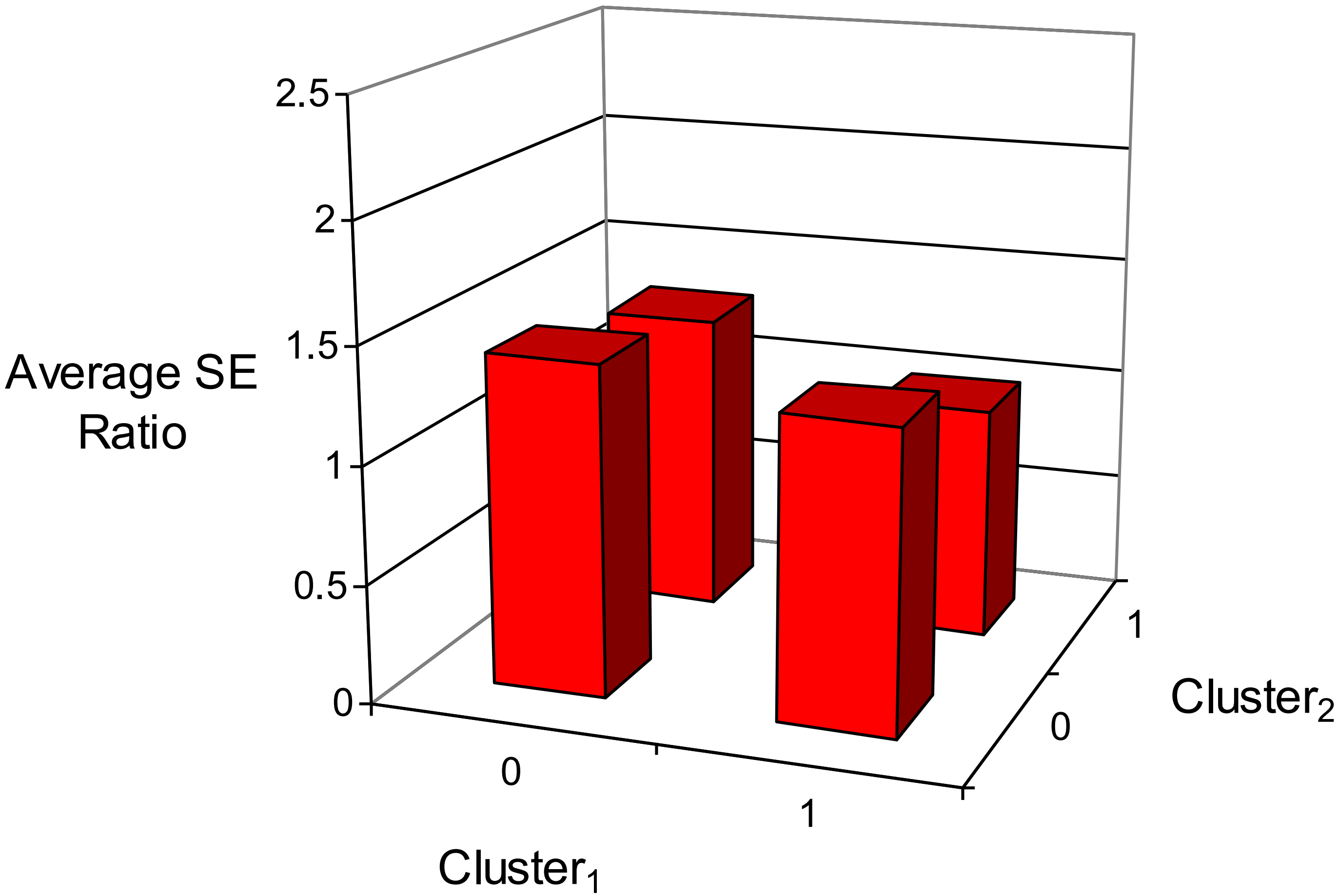


SELECT AVG(col1*col2)



Optimal results in 49% of cases for synthetic data (median = 1.13)
Optimal results in 50% of cases for real-world data

Effect of Clustering: Two Columns



Conclusions

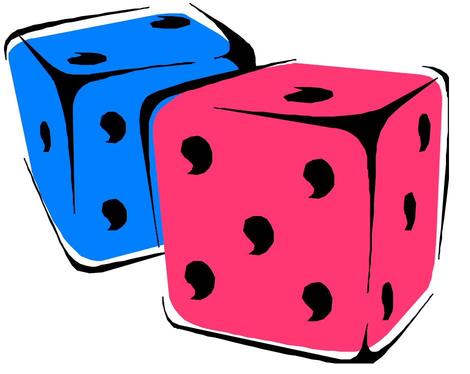
- Bi-level Bernoulli sampling
  - Improved processing of ISO queries
  - Control over speed vs precision
- Have provided optimal parameter settings
  - For important class of aggregation queries
  - Bang-bang solution
- Practical heuristic for setting *p* and *r*
  - Avoids pilot sampling
  - Empirical demonstration of effectiveness

Future Work

- Theory
  - Extend optimality results and heuristics to multi-table queries
  - Extend results to other sampling schemes
    - Workload-aware (Chaudhuri et al. 2001)
    - Synopsis-aware (Acharya et al. 1999; Ganguly et al. 1996)
- Systems
  - SAMPLE UNIT
  - Communication of User ➔ DBMS: time/accuracy constraints
  - Communication of DBMS ➔ User: choice of p and r
  - Built-in computation of standard error?
    - For special cases?
    - DBMS and SQL language issues

Acknowledgements

- P. Brown
- M. Cilimdzic
- DB2 Sampling Folks:
  - K. Kulkarni
  - G. Lapis
  - G. Lohman
  - C. Nakagawa
  - H. Pirahesh
  - R. Sidle
  - D. Simmen
  - A. Singh
  - T. Truong
  - M. Winer
  - M. Zaharioudakis
  - C. Zuzarte
  - …



Further info and references:
www.almaden.ibm.com/cs/people/peterh

Backup Slides

Sampling Queries in SQL

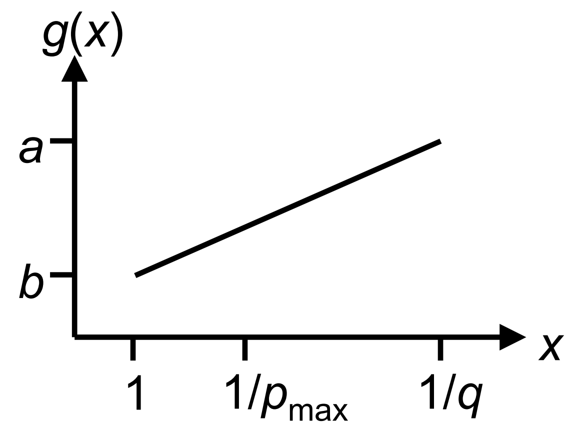
- SAMPLE UNIT = page ID
- Needed for pure page-level sampling also
- A gap in the ISO standard
- SQL extensions?
  - Challenging language issues
  - SELECT STDERR(SUM(T.c))?

```
WITH
dt1 AS (SELECT sales,
            SAMPLE UNIT FOR trans AS s_u
    FROM trans TABLESAMPLE
      BI-LEVEL-BERNOULLI(100*:q,100*:p)),
dt2 AS (SELECT SUM(sales) as s_sales,
     SUM(sales)/:r AS alpha_hat,
     SUM(sales*sales) AS s_v2
     FROM dt1 GROUP BY s_u),
dt3 AS (SELECT
    SUM(alpha_hat*alpha_hat) AS s_alpha_hat2,
    SUM(s_sales) AS tot_s_sales,
    SUM(s_v2) AS tot_s_v2 FROM dt2)
SELECT
   tot_s_sales/:q AS estimated_total_sales,
   SQRT((1e0/:p)*((1e0/:p)-1e0)*s_alpha_hat2
   + (1e0/:q)*((1e0/:r)-1e0)*tot_s_v2) AS std_error
FROM dt3;
```

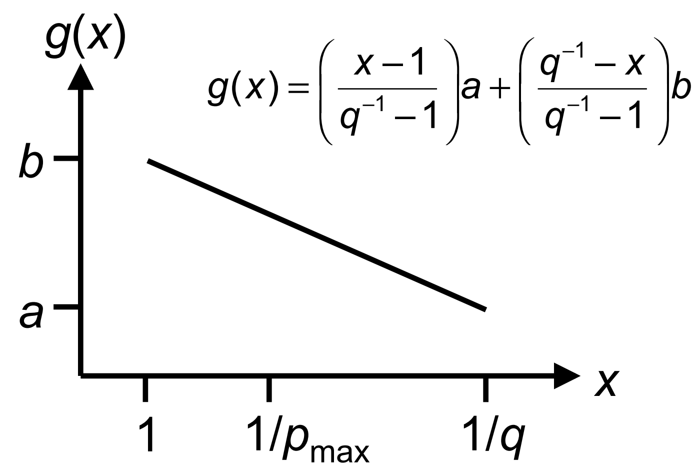
P1 Optimal Solution (I/O Cost Model)

- Original problem:

> Minimize $v(p,r)$
> s.t. $p\ r = q$ and $C(p) \le C_{max}$



- Transform problem:
  - $C(p) \le C_{max} \Leftrightarrow p \le p_{max}$
  - Divide $v(p,r)$ by $q^{-1} - 1$
  - Set $x = 1/p$



- New problem:

> Minimize $g(x)$
> s.t. $1/p_{max} \le x \le 1/q$

Extension of Heuristic

- Multiple aggregates in SELECT statement
  - Look at square root of average variance or
  - Minimize the maximum standard error