Detecting Attribute Dependencies from Query Feedback

Peter J. Haas[1], Fabian Hueske[2], Volker Markl[1]

[1]IBM Almaden Research Center

[2]Universität Ulm

The Problem: Detecting (Pairwise) Dependent Attributes

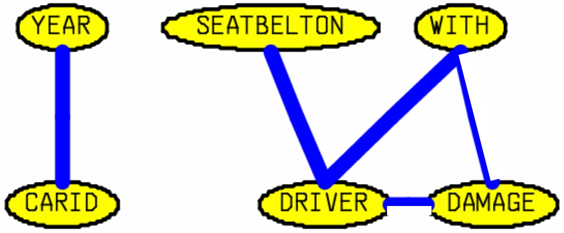
- Example: Color and Year are **independent** if

  F(Color = ‘red’ AND Year = ‘2005 ’) = *F* (Color = ‘red ’) x *F* (Year = ‘2005’)
  F(Color = ‘blue’ AND Year = ‘2007’) = *F* (Color = ‘blue ’) x *F* (Year = ‘2007’)
  etc.

  - *F* (*P*) = fraction of rows in table that satisfy predicate *P*
  - **Dependence** = “significant” departure from independence

- Detection needed for **automatic statistics configuration** in query optimizers
  - Which multivariate statistics should we keep?
  - Need to **rank** the dependencies (limited space budget)

- **Other uses** include
  - Schema discovery for data integration
  - Data mining (dependency diagrams)
  - Root-cause analysis and system monitoring



- Approaches to detection and ranking: **proactive** and **reactive**

Outline

- Previous approaches
  - Proactive approach: CORDS
  - Reactive approaches: SASH, Correlation analyzer
- Our new reactive approach
  - Dependency detection
  - Handling incomplete feedback, inconsistencies
  - Ranking
- Experimental Results

A Proactive Approach: CORDS [IMH+, SIGMOD '04]

- Sample the relation (or view) and compute a **contingency table:**

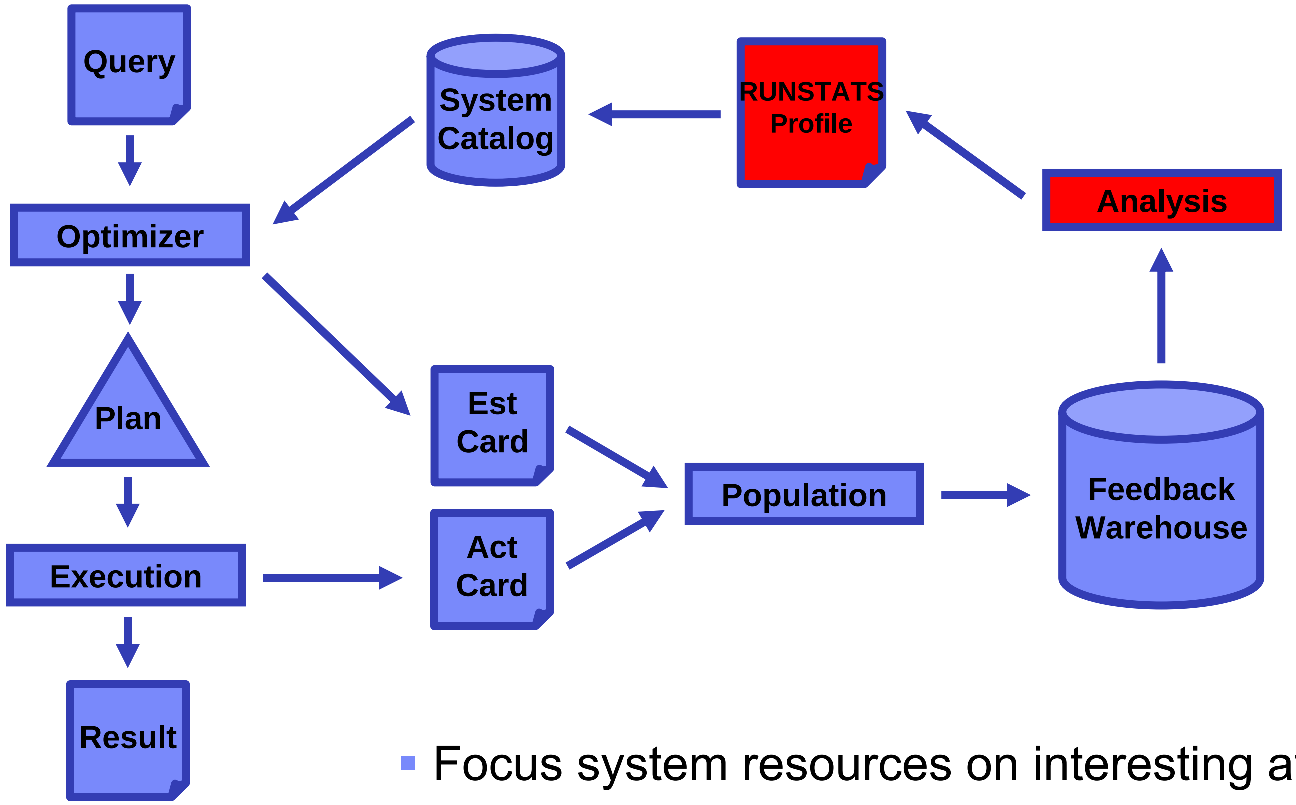
	Blue	Green	Red	
2005	200	400	300	900
2006	150	400	320	870
2007	100	600	200	900
	450	1400	820	2670

- Compute (robust) chi-squared statistic

$$\chi^2 = \sum_i \frac{(O_i - E_i)^2}{E_i} = \frac{\left(200 - \left(\frac{900}{2670}\right)\left(\frac{450}{2670}\right)2670\right)^2}{\left(\frac{900}{2670}\right)\left(\frac{450}{2670}\right)2670} + \cdots$$

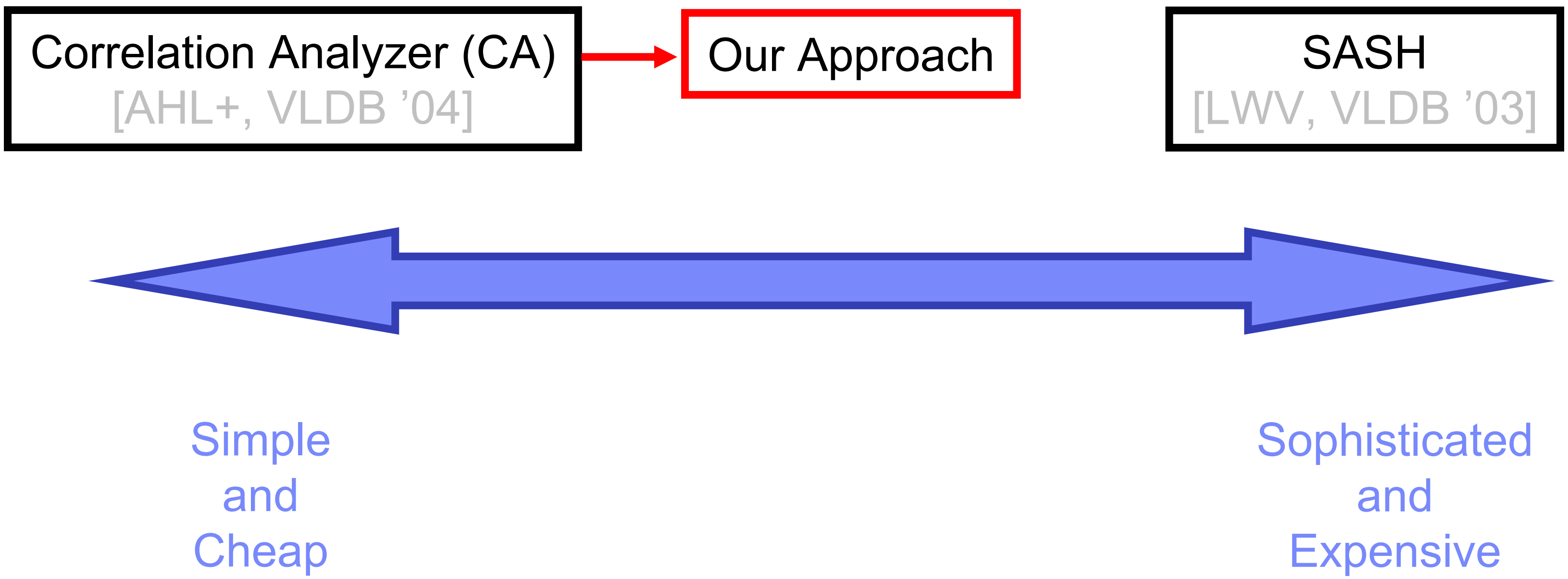
- Declare dependency if $\chi^2 > t$
- Both *t* and sample size chosen using chi-squared theory
- Can rank attribute pairs by **mean-square contingency distance (MSCD)**
  - Normalized chi-squared statistic

Reactive Approaches



- Focus system resources on interesting attributes
- Complement proactive approaches
- Can exploit DB2 feedback warehouse

A Spectrum of Reactive Approaches



Correlation Analyzer

- Uses multiple **observations** (actuals) for each attribute pair
  - **O_1 =** {(blue,2005): 0.02, (blue): 0.2, (2005): 0.103}
  - **O_2 =** {(red,2006): 0.07, (red): 0.82, (2006): 0.11}
  - etc.
- Computes ratio for each pair and compares to $1 \pm \Theta$, e.g. [0.9,1.1]
  - **O_1**: 0.02 / (0.2 x 0.103) = 0.97 independent
  - **O_2**: 0.07 / (0.82 x 0.11) = 0.77 dependent
- Attribute dependency if two or more observations look dependent
- Ranks attributes by weighted sum of violations
- Problems
  - Ad hoc procedures, wasted information
  - Unstable: depends on amount, ordering of feedback

Outline

- Previous approaches
  - Proactive approach: CORDS
  - Reactive approaches: SASH, Correlation analyzer
- Our new reactive approach
  - Dependency detection
  - Handling incomplete feedback, inconsistencies
  - Ranking
- Experimental Results

A New Approach to Dependency Discovery

- Like CORDS, but uses *incomplete* contingency table with *exact* entries

	Blue	Green	Red	
2005	200	?	?	900
2006	?	?	320	870
2007	?	?	?	?
	450	?	820	2670

- Declare dependency if $H_M > u$ (where H_M is our new test statistic)
- Critical value u from extension of classical chi-squared theory
- Normalize H_M to get ranking metric

The H_M Statistic

$f_{\alpha_i\beta_i}$ = fraction of rows with $t.A = \alpha_i$ and $t.B = \beta_i$

- Set $H_M = M\ x^t Q\ x$
  - M = number of rows in table
  - $x_i = (O_i - E_i\ ) / E_i$
  - Q is "pseudo-inverse" of Σ
  - Note: $1 \le i,j \le$ # **observations**

$$\Sigma_{ij} = \begin{cases} \dfrac{(1-f_{\alpha_i\bullet})(1-f_{\bullet\beta_i})}{f_{\alpha_i\bullet}\, f_{\bullet\beta_i}} & \text{if } i = j \\ -\dfrac{1-f_{\alpha_i\bullet}}{f_{\alpha_i\bullet}} & \text{if } i \ne j,\ \alpha_i = \alpha_j, \text{ and } \beta_i \ne \beta_j \\ -\dfrac{1-f_{\bullet\beta_i}}{f_{\bullet\beta_i}} & \text{if } i \ne j,\ \alpha_i \ne \alpha_j, \text{ and } \beta_i = \beta_j \\ 1 & \text{if } i \ne j,\ \alpha_i \ne \alpha_j, \text{ and } \beta_i \ne \beta_j \end{cases}$$

- r = rank of Q

- Properties: similar to χ^2
  - $H_M \ge 0$
  - $H_M = 0$ iff observations consistent with independence
  - Larger $H_M \Rightarrow$ less consistent with independence

Choosing the Threshhold *u*

- Superpopulation approach
  - Assume *A* and *B* generated by truly independent mechanism

Theorem: Under this model, for large # of rows, H_M has approximately a χ_r^2 distribution

- Choose *u* as (1 – *p*) quantile of χ_r^2 for small p. Then

$$\mathrm{Prob}\left\{H_M > u\right\} \approx \mathrm{Prob}\left\{\chi_r^2 > u\right\} = p$$

Missing Feedback

- Most important case: O_i = { (blue,2005): 0.02, (blue): 0.2, (2005): ? }
- Assume optimizer estimate of (2005) frequency available
- Assume (rough) upper bound on abs(relative error of estimate)
  - Can obtain from feedback-warehouse records
- Fill in missing frequency for (2005)
  - Derive rough bounds on true value: $l \le F\,(2005) \le u$
  - Make frequency “as independent as possible” (conservative)
    - E.g., $F\,(2005) = 0.1$ and $E_i = r_i - 1 = 0$
  - Consider ALL observations with missing (2005) frequency
  - Minimize $\Sigma_i\,(E_i)^2$ (closed-form solution available)

Handling Inconsistency

- Problem: No full multivariate frequency distribution consistent with feedback
  - Records collected at different time points
  - Inserts/deletes/updates in between feedback observations
- Solution method 1: use **timestamps** to resolve conflicts
- Solution method 2: **linear programming**
  - Obtain minimal adjustment of frequencies needed for consistency

$$\min \sum_i w_i (s_i^+ + s_i^-)$$

s.t.

$$F(\text{blue}, 2005) + s_3^+ - s_3^- = 0.2$$

$$F(2005) + s_{17}^+ - s_{17}^- = 0.3$$

$$\vdots$$

$$\sum_{\text{color}} F(2005, \text{color}) = F(2005)$$

$$\vdots$$

$$s_i^+, s_i^- \geq 0 \text{ for all } i$$

$$\boxed{F'(\text{blue}, 2005) = F(\text{blue}, 2005) - s_3^+ + s_3^-}$$

Ranking Attribute Pairs

- Problem: normalize H_M (= $M\, x^t\, Q\, x$) to lie in [0,1]
- Guaranteed (conservative) normalization η
  - Based on Courant-Fischer Minimax Theorem

$$H_M \leq \eta = M\, d^* \|x\|^2 \text{, where } d^* = \text{ largest eigenvalue of } Q$$

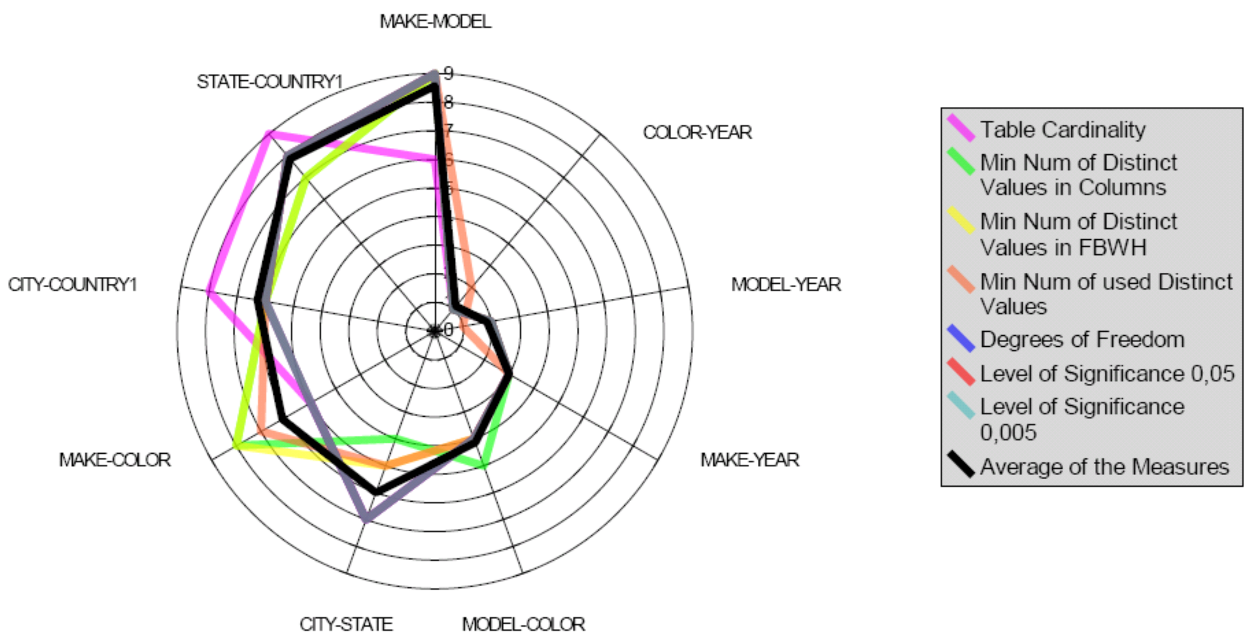
  - Can be numerically unstable (huge values of η)
- Heuristic normalizations $H_M\, /\, z$
  - Table Cardinality
  - Minimal number of distinct values
  - Degrees of freedom of chi-squared distribution
  - 0.99 Quantile of χ_r^2 (“effective” upper bound)

Outline

- Previous approaches
  - Proactive approach: CORDS
  - Reactive approaches: SASH, Correlation analyzer
- Our new reactive approach
  - Dependency detection
  - Handling incomplete feedback, inconsistencies
  - Ranking
- Experimental Results

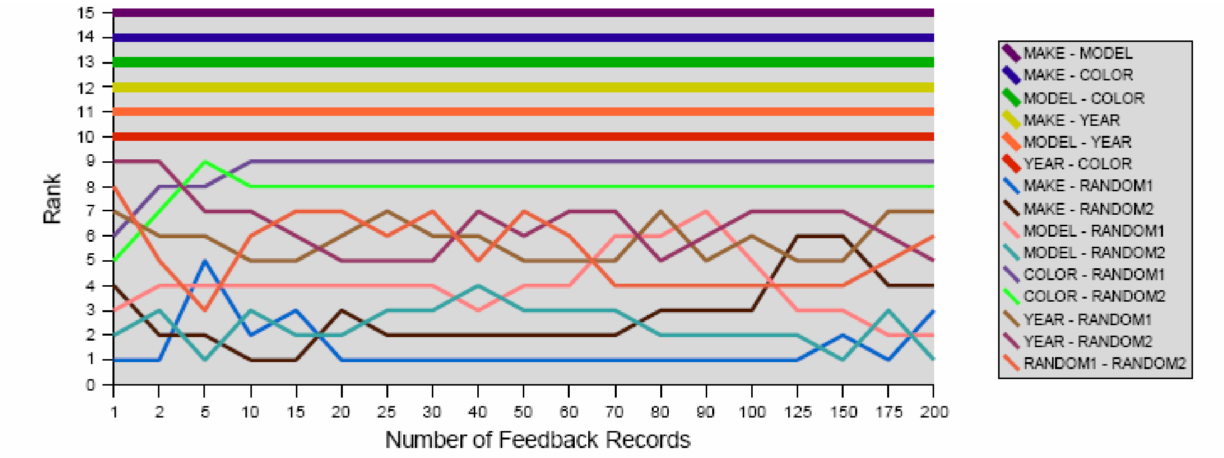
Normalization Constants

- Rankings relatively consistent for different *z* (choice is not too critical)
- Best results: degrees of freedom, quantiles ("high probability" upper bound)

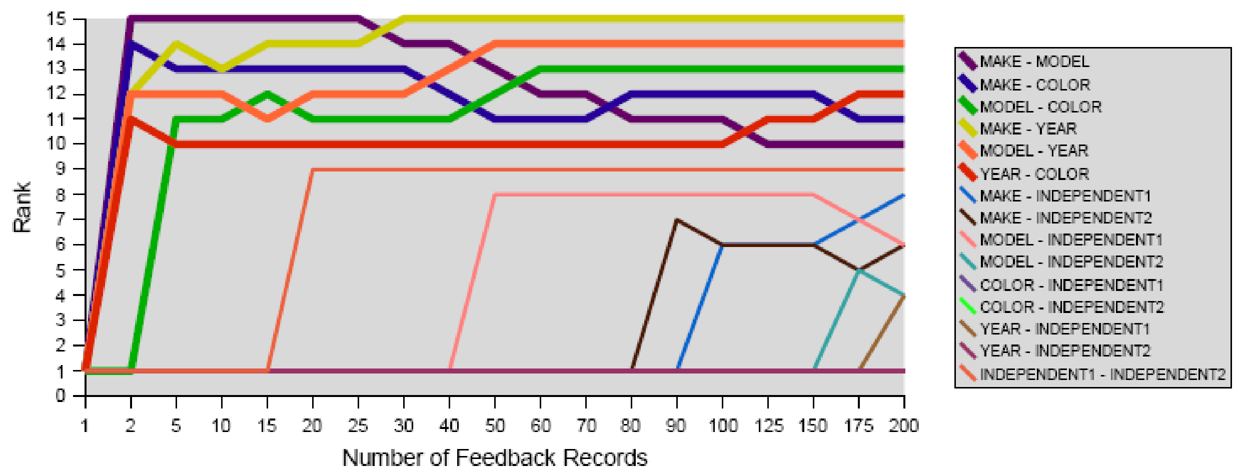


Ranking vs Amount of Feedback

New method:

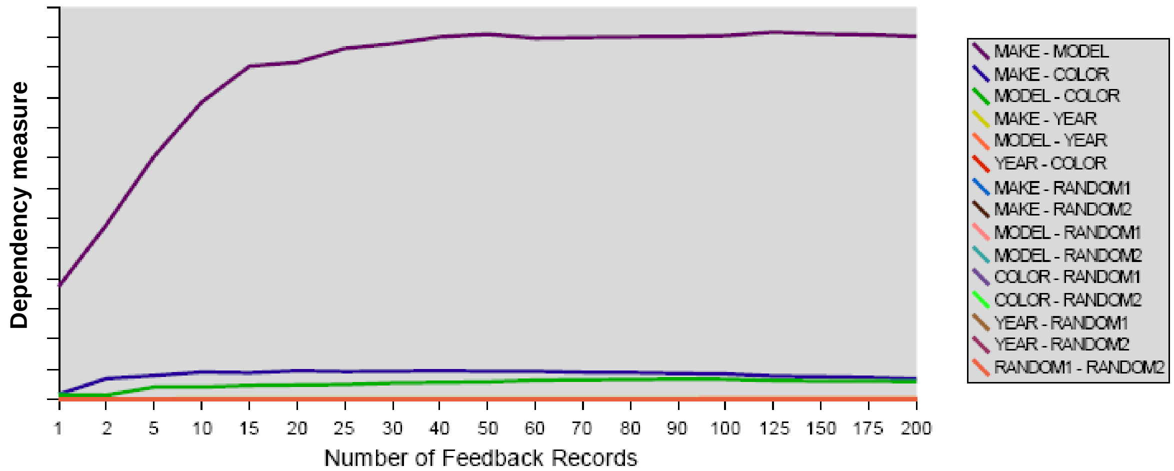


Correlation analyzer:

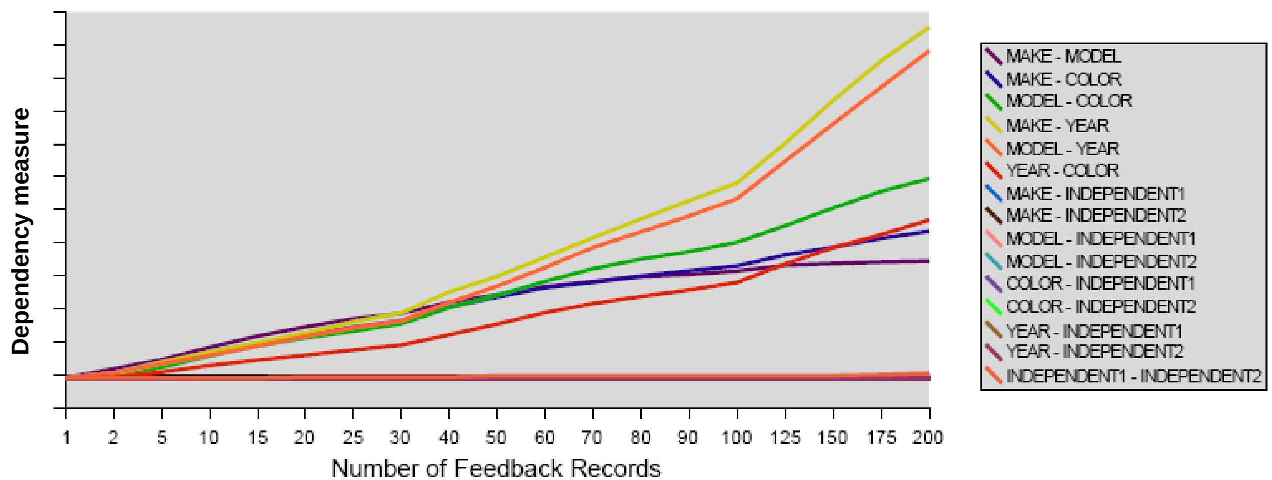


Dependency Measure vs Amount of Feedback

New method:

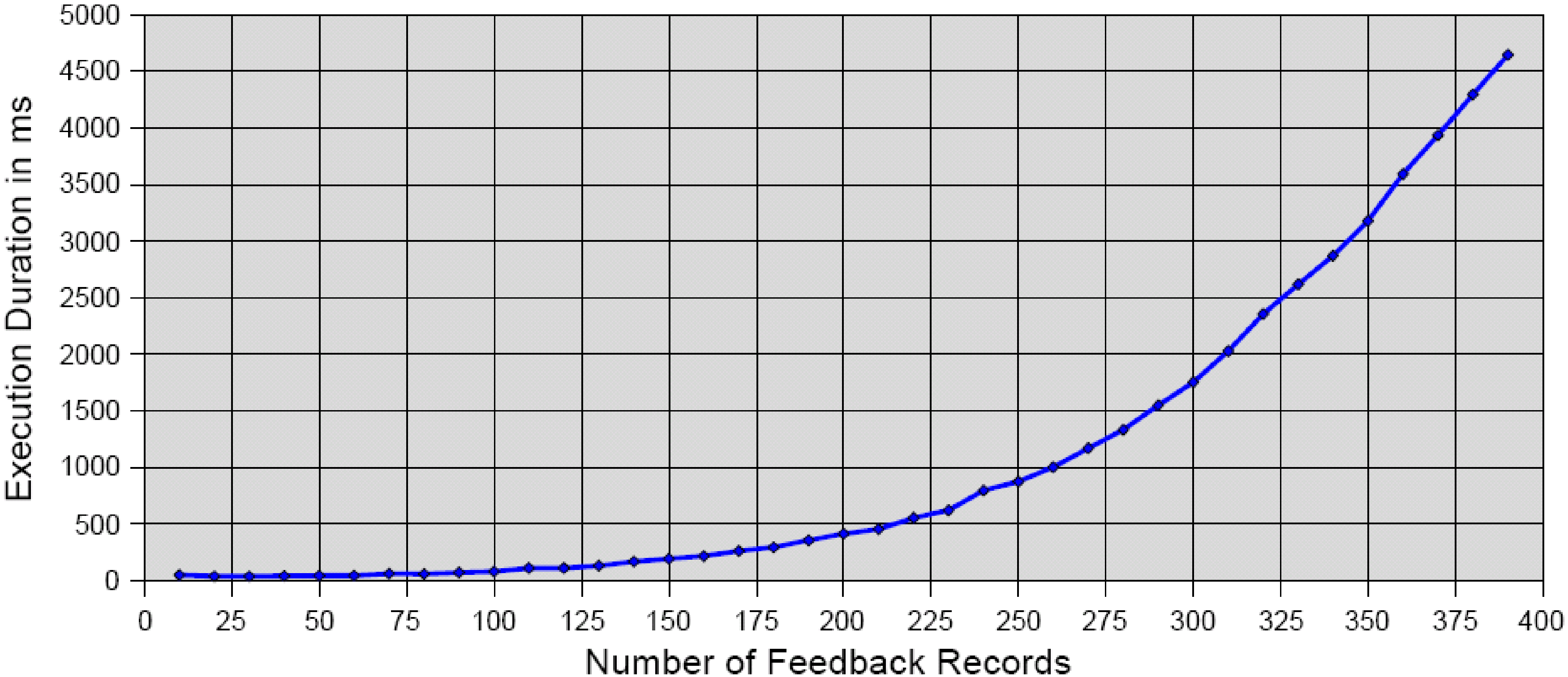


Correlation analyzer:



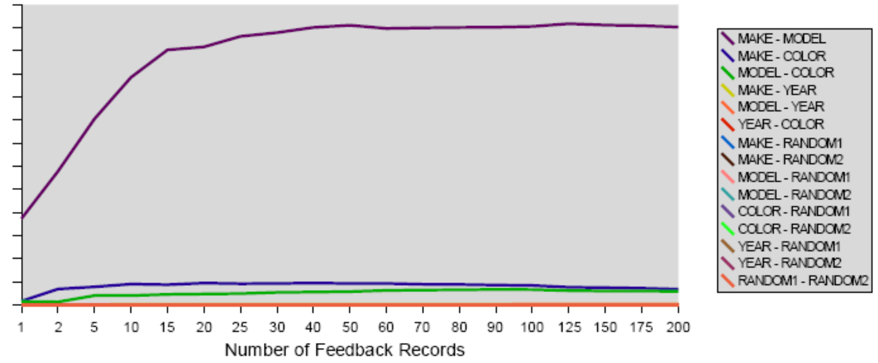
Execution Time

- $O(n^3)$ theoretical complexity
- Subsecond execution time for up to 250 feedback records
- Times based on preliminary Java implementation

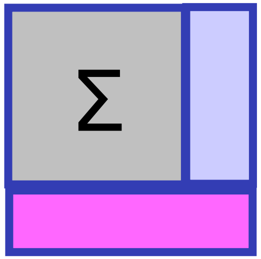


Obtaining Practical Execution Times

- **Sampling**
  - Stable results with small # of obs.
  - Sub-second response times



- **Incremental maintenance** of $H_M = M x^t Q x$
  - New observation = add new row + new column to Σ
  - Want to update Q directly
    - Q = pseudo-inverse of Σ
  - Apply SVD updating methods
    - As in latent semantic indexing
    - E.g., "folding-in" method $O(k^2)$

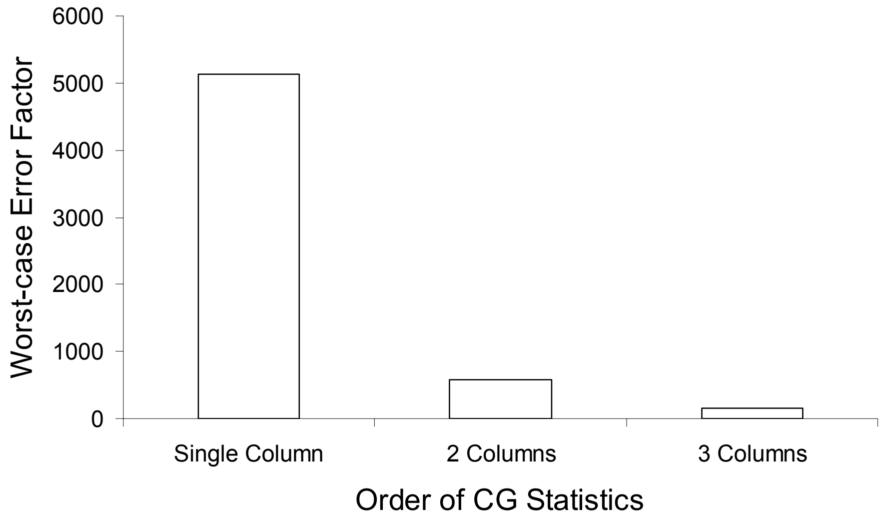


Conclusions

- Dependence is everywhere!
- Query feedback is an effective way to detect dependence
- Chi-squared extension to implement detection
  - Attributes can be in multiple tables
- Effective ranking methods
- Practical solutions for handling inconsistent or missing feedback
- Acceptable performance using sampling and incremental maintenance

Future Work

- Higher-level dependencies



- Full integration of proactive and reactive methods
  - Cf. Aboulnaga et al. [VLDB 2004]

The End

My web page:

www.almaden.ibm.com/cs/people/peterh

LEO (LEarning Optimizer) project:

http://domino.watson.ibm.com/comm/research.nsf/pages/r.datamgmt.innovation.html

The End

Backup Slides

The H_M Statistic (Based on n Observations)

- Set $x_i = \dfrac{f_{\alpha_i\beta_i} - f_{\alpha_i\bullet} f_{\bullet\beta_i}}{f_{\alpha_i\bullet} f_{\bullet\beta_i}}$ for $i = 1, 2, \ldots, n$

$f_{\alpha_i\beta_i}$ = fraction of rows with $t.A = \alpha_i$ and $t.B = \beta_i$

- Set $\Sigma = \left\| \Sigma_{ij} \right\|$, where

$$\Sigma_{ij} = \begin{cases} \dfrac{(1 - f_{\alpha_i\bullet})(1 - f_{\bullet\beta_i})}{f_{\alpha_i\bullet} f_{\bullet\beta_i}} & \text{if } i = j \\ -\dfrac{1 - f_{\alpha_i\bullet}}{f_{\alpha_i\bullet}} & \text{if } i \neq j,\ \alpha_i = \alpha_j, \text{ and } \beta_i \neq \beta_j \\ -\dfrac{1 - f_{\bullet\beta_i}}{f_{\bullet\beta_i}} & \text{if } i \neq j,\ \alpha_i \neq \alpha_j, \text{ and } \beta_i = \beta_j \\ 1 & \text{if } i \neq j,\ \alpha_i \neq \alpha_j, \text{ and } \beta_i \neq \beta_j \end{cases}$$

The H_M Statistic, Continued

- Symmetric Shur decomposition: $\Sigma = G^t D G$

  where $D = \mathrm{diag}(d_1, d_2, \ldots, d_n)$
- Set $\tilde{D} = \mathrm{diag}(\tilde{d}_1, \tilde{d}_2, \ldots, \tilde{d}_n)$, where

$$\tilde{d}_i = \begin{cases} 1/d_i & \text{if } d_i > 0 \\ 0 & \text{if } d_i = 0 \end{cases}$$

- Set $Q = G^t \tilde{D} G$
- Q is pseudo-inverse of Σ: $Q\Sigma = \Sigma Q = I_r$
- Set $M =$ # rows in table
- Then $H_M = M x^t Q x$
- Set $r = r(Q) =$ # positive diagonal entries in D