Detecting Attribute Dependencies from Query Feedback

Peter J. Haas¹, Fabian Hueske², Volker Markl¹

¹IBM Almaden Research Center
²Universität Ulm
The Problem: Detecting (Pairwise) Dependent Attributes

- Example: Color and Year are independent if
  \[
  F(\text{Color} = \text{'red'} \text{ AND Year} = \text{'2005'}) = F(\text{Color} = \text{'red'}) \times F(\text{Year} = \text{'2005'})
  
  F(\text{Color} = \text{'blue'} \text{ AND Year} = \text{'2007'}) = F(\text{Color} = \text{'blue'}) \times F(\text{Year} = \text{'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:

<table>
<thead>
<tr>
<th></th>
<th>Blue</th>
<th>Green</th>
<th>Red</th>
</tr>
</thead>
<tbody>
<tr>
<td>2005</td>
<td>200</td>
<td>400</td>
<td>300</td>
</tr>
<tr>
<td>2006</td>
<td>150</td>
<td>400</td>
<td>320</td>
</tr>
<tr>
<td>2007</td>
<td>100</td>
<td>600</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td>450</td>
<td>1400</td>
<td>820</td>
</tr>
</tbody>
</table>

- Compute (robust) chi-squared statistic

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

- 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 (CA) [AHL+, VLDB ’04]
- Our Approach
- SASH [LWV, VLDB ’03]

Simple and Cheap

Sophisticated and Expensive
Correlation Analyzer

- Uses multiple observations (actuals) for each attribute pair
  - \( O_1 = \{(\text{blue}, 2005): 0.02, \text{blue}: 0.2, (2005): 0.103\}\)
  - \( O_2 = \{(\text{red}, 2006): 0.07, \text{red}: 0.82, (2006): 0.11\}\)
  - etc.

- Computes ratio for each pair and compares to \( 1 \pm \Theta \), e.g. \([0.9, 1.1]\)
  - \( O_1: 0.02 / (0.2 \times 0.103) = 0.97 \) independent
  - \( O_2: 0.07 / (0.82 \times 0.11) = 0.77 \) 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

<table>
<thead>
<tr>
<th></th>
<th>Blue</th>
<th>Green</th>
<th>Red</th>
</tr>
</thead>
<tbody>
<tr>
<td>2005</td>
<td>200</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>2006</td>
<td>?</td>
<td>?</td>
<td>320</td>
</tr>
<tr>
<td>450</td>
<td>?</td>
<td>820</td>
<td>2670</td>
</tr>
</tbody>
</table>

- 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

- 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 $\Sigma$
  - Note: $1 \leq i,j \leq \#$ observations

- $r =$ rank of $Q$

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

\[ f_{\alpha_\beta_i} = \text{fraction of rows with } t.A = \alpha_i \text{ and } t.B = \beta_i \]

\[ \Sigma_{ij} = \begin{cases} 
(1 - f_{\alpha_i})(1 - f_{\beta_j}) & \text{if } i = j \\
\frac{1 - f_{\alpha_i}}{f_{\alpha_i}} & \text{if } i \neq j, \alpha_i = \alpha_j, \text{ and } \beta_i \neq \beta_j \\
\frac{1 - f_{\beta_i}}{f_{\beta_i}} & \text{if } i \neq j, \alpha_i \neq \alpha_j, \text{ and } \beta_i = \beta_j \\
1 & \text{if } i \neq j, \alpha_i \neq \alpha_j, \text{ and } \beta_i \neq \beta_j 
\end{cases} \]
Choosing the Threshold $u$

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

**Theorem:** Under this model, for large # of rows, $H_M$ has approximately a $\chi^2$ distribution

- Choose $u$ as $(1 - p)$ quantile of $\chi^2_r$ for small $p$. Then

$$\text{Prob}\{H_M > u\} \approx \text{Prob}\{\chi^2_r > u\} = p$$
Missing Feedback

- Most important case: $O_i = \{ \text{blue,2005): 0.02, (blue): 0.2, (2005): ? } \}$

- 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: $l \leq F(2005) \leq u$
  - 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 $\sum_i (E_i)^2$ (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

\[
\begin{align*}
\min & \sum_i w_i (s_i^+ + s_i^-) \\
\text{s.t.} & \\
F(\text{blue}, 2005) + s_3^+ - s_3^- &= 0.2 \\
F(2005) + s_{17}^+ - s_{17}^- &= 0.3 \\
\vdots & \\
\sum_{\text{color}} F(2005, \text{color}) &= F(2005) \\
\vdots & \\
s_i^+, s_i^- &\geq 0 \text{ for all } i
\end{align*}
\]

\[
F'(\text{blue}, 2005) = F(\text{blue}, 2005) - s_3^+ + s_3^-
\]
Ranking Attribute Pairs

- Problem: normalize $H_M ( = M x^t Q x)$ to lie in [0,1]
- Guaranteed (conservative) normalization $\eta$
  - 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 $\eta$)
- Heuristic normalizations $H_M / z$
  - Table Cardinality
  - Minimal number of distinct values
  - Degrees of freedom of chi-squared distribution
  - 0.99 Quantile of $\chi^2_r$ (“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 $H_M = M \times^t Q \times$
  - New observation =
    - add new row + new column to $\Sigma$
  - Want to update $Q$ directly
    - $Q = \text{pseudo-inverse of } \Sigma$
  - 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:

The End

Backup Slides
The $H_M$ Statistic (Based on $n$ Observations)

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

- Set $\Sigma = \sum_{ij}$, where

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

$f_{\alpha_i \beta_i}$ = fraction of rows with $t.A = \alpha_i$ and $t.B = \beta_i$
The $H_M$ Statistic, Continued

- Symmetric Shur decomposition: $\Sigma = G^t DG$
  where $D = \text{diag}(d_1, d_2, \ldots, d_n)$
- Set $\tilde{D} = \text{diag}(\tilde{d}_1, \tilde{d}_2, \ldots, \tilde{d}_n)$, 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 $\Sigma$: $Q \Sigma = \Sigma Q = I_r$
- Set $M = \# \text{ rows in table}$
- Then $H_M = Mx^t Qx$
- Set $r = r(Q) = \# \text{ positive diagonal entries in } D$