Time-completeness trade-offs in record linkage using Adaptive query Processing

Paolo Missier, Alvaro A. A. Fernandes
School of Computer Science, University of Manchester

Roald Lengu, Giovanna Guerrini
DISI, Universita' di Genova, Italy

Marco Mesiti
DiCo, Universita' di Milano, Italy

EDBT 2009, St. Petersburg, Russia
March 26th, 2009
Context

• “Situational Applications”
  • designed to address a transient need and short-lasting
  • constructed “on the fly”
• Data mashups and personal dataspaces

  • sources 1..n: collection of car insurance DBs
  • source n+1: reference street atlas
  • target app: mapping accidents hotspots

  – no prior knowledge of data sets (streams) to be joined
  – reasonable to join on some of the attributes
  – but, no guarantee of matching values
On-the-fly integration of relational data

• slight mismatches in records lead to incomplete integration
  – due to different encodings, conventions
  – due to errors in data

\[ R \bowtie_{A=B} S \]

A case of record linkage
Offline vs online linkage

- **Offline record linkage:**
  - performed once before queries involving joins
  - 1. reconcile R and S on joining attributes R.A, S.B using your favourite record linkage technique
    \[
    \langle R, S \rangle \rightarrow \langle R', S' \rangle
    \]
  - 2. perform regular equijoin on the transformed tables:
    \[
    R' \bowtie S'
    \]
    ➡ ok for tables that can be analysed ahead of the join
    ➡ ok when multiple queries issued on integrated tables
Offline vs online linkage

- Offline record linkage:
  - performed once before queries involving joins
  - 1. reconcile $R$ and $S$ on joining attributes $R.A$, $S.B$ using your favourite record linkage technique
    \[
    \langle R, S \rangle \rightarrow \langle R', S' \rangle
    \]
  - 2. perform regular equijoin on the transformed tables:
    \[
    R' \bowtie S'
    \]
    - ok for tables that can be analysed ahead of the join
    - ok when multiple queries issued on integrated tables

- Online linkage:
  - performed while answering a query
  - exact join $\Rightarrow$ similarity (or approximate) join
Record linkage and similarity joins

Historical timeline:

from:
N. Koudas, S. Sarawagi, and D. Srivastava. Record linkage: similarity measures and algorithms. Tutorial in SIGMOD ’06.
Historical timeline:

from:
N. Koudas, S. Sarawagi, and D. Srivastava. Record linkage: similarity measures and algorithms. Tutorial in SIGMOD ’06.
Measuring string similarity using q-grams

- **q-grams** map string $s$ to a set $q(s)$ of substrings of length $q$:

Ex.: 3-grams:


\[
sim(s_1, s_2) = \frac{|q(s_1) \cap q(s_2)|}{|q(s_1) \cup q(s_2)|}
\]

(Jaccard coefficient)

\[
sim(\text{“Manchester”, “Madchester”}) = 5/8
\]

\[
sim(r_1, r_2) < \theta_1 \rightarrow \text{not match}
\]

\[
\theta_2 < \sim(r_1, r_2) \rightarrow \text{match}
\]
• primitive join operator [CGK06]
Similarity symmetric hash join

Efficient relational representation:
Similarity symmetric hash join

Efficient relational representation:
Similarity symmetric hash join

- pipelined: suitable for stream processing

- Main sources of complexity:
  - overhead for storing and indexing q-grams
  - cost of computing set intersection
Is full similarity join always necessary?

- Pessimistic: always pays full complexity cost
- Typical mismatch rate in real datasets around 5%

**Research Goal:** explore optimistic approach
- detect mismatches and react as you go
- requires estimates of incremental join result size
- statistical + reactive ⇒
  - expect to sacrifice join result completeness for faster execution

**Approach:**
- combine exact and similarity join operators using Adaptive Query Processing techniques
Autonomic computing framework

- Monitor
- Respond
- Assess
Autonomic computing framework

monitor

respond

assess

monitor incremental join result size
monitor incremental join result size

estimate join result size

compute divergence

monitor

respond

assess

Autonomic computing framework
• when using exact join:
  • if observed/estimated sizes diverge “too much”, then switch to approximate join
• when using approximate join:
  • if observed/estimated converge, then switch to exact join
Technical approach and challenges

- **Assess:**
  - need additional assumption for result size estimation
  - estimating result size at specific points during join execution

- **Respond:**
  - when and how can physical join operators switched?
  - Can we avoid loss of work?
    - operator replacement in pipelined query plans [EFP06]

---

Assessment

- Expectation of matching records
  - implicit referential integrity between tables
- Used to derive simple result size estimation model

When there are no mismatches:

after scanning $n < |S|$ tuples on S:

$P(a=x \text{ in } |S| \text{ has been matched}) = P(\text{tuple } c=x \text{ is in top } n \text{ of } R) = n/|R|$

Thus, join result size $O_n$ after $n$ tuples is a binomial random variable:

$$O_n \sim \text{bin}(n, \frac{n}{|R|})$$
Detecting divergent observed result size

Observation $\bar{O}_n$ is outlier wrt expected result size $O_n$
=> divergence $P_{n,p(n)}(\bar{O}_n \leq O) \leq \theta_{out}$

where $P_{n,p(n)}(.)$ is the binomial cdf with parameters $n, p(n)$
Detecting divergent observed result size

Observation $\bar{O}_n$ is outlier wrt expected result size $O_n$  
$=>$ divergence $P_{n,p(n)}(\bar{O}_n \leq O) \leq \theta_{out}$

where $P_{n,p(n)}(\cdot)$ is the binomial cdf with parameters $n, p(n)$

outlier detection on experimental datasets with various mismatch patterns
Detecting divergent observed result size

Observation $\bar{O}_n$ is outlier wrt expected result size $O_n$  
=> divergence $P_{n,p(n)}(\bar{O}_n \leq O) \leq \theta_{out}$

where $P_{n,p(n)}(.)$ is the binomial cdf with parameters $n, p(n)$

outlier detection on experimental datasets with various mismatch patterns

Note: when the data does not follow our referential constraint hypothesis, the model leads to a pure similarity join
• Goal of AQP:
  – switch operators without loss of intermediate work
• sufficient condition: switch when the operator reaches a *quiescent* state [EPF06]

Responder’s state machine

- Operator switch defined in terms of state transitions
- Owing to symmetry, we can use a different operator on each of the two tables
Instantiating the MAR framework

- Monitor
- Respond
- Assess

- Incremental result size
- Estimate result size
- Compute divergence predicates

Switch join operators

$O_n$
Instantiating the MAR framework

monitor

respond

assess

incremental result size

estimate result size

compute divergence predicates

switch
join
operators

$O_n$
Instantiating the MAR framework

\[ \sigma(n) \equiv P_{n,p(n)}(\bar{O}_n \leq O) \leq \theta_{out} \]

\[ \mu_i(t) \equiv \frac{A_{t,W}}{W} \leq \theta_{curpert} \]

\[ \pi_i(t) \equiv \sum_{t' < t} I(\mu_i(t')) \leq \theta_{pastpert} \]

Discrepancy detected

Current perturbations on left/right?

Past perturbations on left/right?
Rationale for state transitions

Evidence that left and/or right input perturbed

Predicates $\sigma(t), \mu(t), \pi(t)$ provide the evidence needed to drive the transitions

Evidence that left and/or right input no longer perturbed
\[ \sigma(n) \equiv P_{n,p(n)}(\bar{O}_n \leq O) \leq \theta_{out} \]

\[ \mu_i(t) \equiv \frac{A_{t,W}}{W} \leq \theta_{curpert} \]

\[ \pi_i(t) \equiv \sum_{t' < t} I(\mu_i(t')) \leq \theta_{pastpert} \]
\[ \sigma(n) \equiv P_{n,p(n)}(\bar{O}_n \leq O) \leq \theta_{out} \]

\[ \mu_i(t) \equiv \frac{A_{t,W}}{W} \leq \theta_{curpert} \]

\[ \pi_i(t) \equiv \sum_{t' < t} I(\mu_i(t')) \leq \theta_{pastpert} \]

\[ \varphi_0(t) = \neg \sigma(t) \land \mu_{\text{left}}(t) \land \mu_{\text{right}}(t) \]

\[ \varphi_1(t) = \sigma(t) \land \neg \mu_{\text{left}}(t) \land \neg \mu_{\text{right}}(t) \]

\[ \varphi_2(t) = \sigma(t) \land \neg \mu_{\text{left}}(t) \land \mu_{\text{right}}(t) \land \pi_{\text{left}}(t) \]
Completing the loop

\[ \varphi_0(t) = \neg \sigma(t) \land \mu_{\text{left}}(t) \land \mu_{\text{right}}(t) \]
\[ \varphi_1(t) = \sigma(t) \land \neg \mu_{\text{left}}(t) \land \neg \mu_{\text{right}}(t) \]
\[ \varphi_2(t) = \sigma(t) \land \neg \mu_{\text{left}}(t) \land \mu_{\text{right}}(t) \land \pi_{\text{left}}(t) \]

\[ \sigma(n) \equiv P_{n,p(n)}(O_n \leq O) \leq \theta_{\text{out}} \]
\[ \mu_i(t) \equiv \frac{A_{t,W}}{W} \leq \theta_{\text{curpert}} \]
\[ \pi_i(t) \equiv \sum_{t' < t} I(\mu_i(t')) \leq \theta_{\text{pastpert}} \]
Experimental evaluation

- Marginal gain of hybrid algorithm:
  - level of completeness

  - R: result size for approx join only
  - r: result size for exact only
  - $r_{\text{abs}}$: result size actually observed

  $$g_{\text{rel}} = (r_{\text{abs}} - r) / (R - r)$$
Experimental evaluation

• Marginal gain of hybrid algorithm:
  – level of completeness
    • R: result size for approx join only
    • r: result size for exact only
    • $r_{\text{abs}}$: result size actually observed
      $$g_{\text{rel}} = \frac{r_{\text{abs}} - r}{R - r}$$

• Marginal Cost
  – baseline: exact join throughout
    • model marginal cost of hybrid algorithm
      unit cost of executing one step in one state
        – (experimental)
      • number of steps in each state
      • unit state transition cost (experimental)
      • number of state transitions over entire join
Test datasets

Datasets chosen as representative of 4 distinct patterns

- **a)** No distinctly-marked perturbation regions
- **b)** Few, long perturbation regions of low density
- **c)** Few, long perturbation regions of high density
- **d)** Many, narrow perturbation regions

we expect our results to vary:
- uniform perturbation: evidence grows slowly => slow reaction
- bursty perturbation: strong evidence => timely reaction
Results

<table>
<thead>
<tr>
<th>Pattern</th>
<th>One-sided</th>
<th>Two-sided</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>124.4</td>
<td>115.6</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>130.5</td>
<td>103.4</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>147.8</td>
<td>103.5</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>118.2</td>
<td>108.7</td>
<td></td>
</tr>
</tbody>
</table>

Efficiency = \( \frac{\text{gain}}{\text{cost}} \times 100 \)
Results

efficiency = \frac{gain}{cost} \times 100

- Pattern A, one-sided: 124.4
- Pattern A, two-sided: 115.6
- Pattern B, one-sided: 130.5
- Pattern B, two-sided: 103.4
- Pattern C, one-sided: 147.8
- Pattern C, two-sided: 103.5
- Pattern D, one-sided: 118.2
- Pattern D, two-sided: 108.7

The bar graphs illustrate the transition costs for different patterns and configurations.
Conclusions

• Optimistic approach to online record linkage
  – Based on implicit referential integrity assumption
  – When assumption not true, goes back to pessimistic

• Technical approach based on autonomic computing
  – Adaptive query processing
  – Mix of exact / approximate physical join operators

• Applications: on-the-fly integration scenarios (mashups, personal dataspaces, sensor data streams)

• Results: positive cost / completeness trade-off