Data Quality and Reconciliation

Munir Cochinwala
munir@research.telcordia.com
(973)-829-4864
Credits

• Telcordia
  – Francesco Caruso
  – Uma Ganapathy
  – Gail Lalk
  – Paolo Missier

• Academic
  – Ahmed Elmagarmid
  – Vassilios Verykios
Data Reconciliation

- Matching records
  - Detect duplication
  - Correct data inconsistencies

- Multi-disciplinary problem
  - Record linkage
  - Matching algorithms
  - Clustering algorithms
  - Joins in databases
Problem:
- Mergers and acquisitions of wireless companies resulted in the RBOC’s inability to determine common customers among wireline and wireless businesses.
- Customer databases for each business unit use different schema and contain many quality problems
- RBOC’s experience with commercial vendor’s data reconciliation tool was unsatisfactory

Solution:
- Use small manually-verified data samples (~100 records) to determine appropriate matching rules
- Use machine learning to prune the rules for efficient analysis of the large dataset
- Resulted in 30% more correct matches than the commercial tool
Large Media Conglomorate

Problem:
- Company provides magazines to wholesalers who in turn provide magazines to resalers for distribution. Company loses money because of inconsistencies among wholesaler and retailer databases regarding number of sales.
- Reconciliation of wholesaler and retailer databases would make it easier to track where gaps in reporting are occurring.
- Identify ‘bad’ retailers.

Solution:
- Group by primary keys
- Match by secondary keys
- e.g. 3000 C.V.S. Pharmacies are grouped and compared by zip code and street address – identify ‘bad’ pharmacies
International Government

Problem:

– Reconcile vital taxpayer data from several different sources.
– Known problems include record duplication, address mismatches, address obsolescence, distributed responsibility for database accuracy and updates.
– Identify causes for mistakes

Solution:

– Improve the process flows and architecture to allow for rapid modification of pre-processing rules and matching rules.
– Detection and classification of likely causes of duplication
– Analysis and improvements reduced number of records that required manual verification.
ILEC-CLEC Billing Reconciliation

- **Problem**
  - ILECs charge CLECs for use of network resources
  - Verification of actual usage vs. charging
    - E.g. customer changing providers
  - Identify actual usage and send verification to ILEC
  - Resources identification in ILEC and CLEC are different

- **Solution**
  - Check charges in bill against actual usage
  - Common identification of resources (matching table)
  - Solution has only been implemented for access line charge
Framework

- Rules and solutions differ from domain to domain.
- Flexibility in resolution rules is essential
- Preprocessing and cleaning varies with the nature of the data.

- However, in most cases a common pattern emerges:
  - data is handled in stages,
  - each stage is responsible for one step of transformation (matching, merging, cleaning, etc.) and
  - Stages can be chained together to produce an overall data flow.

- We define a framework that allows
  - any number of custom-made processing blocks
  - Combination of those blocks in various ways
Framework

- data-flow architecture.
  - the framework enforces a simple producer-consumer model between pairs of functional blocks

- flexibility and extensibility:
  - late binding of functional blocks to the framework

- blocks independence
  - component-like: they meet on common interfaces
  - introspection for run-time loading of new blocks.

- XML-based representation of the data flow.
  - Persistence of the flow state

- Data flow compilation:
  - the front-end can be thought of as a way for experts to configure the tool on a specific application.
  - Once configured, the data flow can be compiled into a "black-box" application that can be deployed to end users.
Demo

- Demo of Tool
Framework Issues

- Other Architectures
- Enhancement of the simple producer-consumer model
  - pipelining where possible,
  - parallelize.

- Resource sharing vs blocks independence
- Separating data from algorithms
  - should the matching/cleaning algs be pushed close to the data e.g. DBMS

- Generating code from XML specifications

- Automation of rule generation, pruning
Conclusion

• Framework allows user-defined algorithms and rules
  – multiple clustering algorithms can be incorporated/chosen
  – type specific matching can be designed (e.g. address)

• Extend tool to other domains
  – auto-discovery of networks and populating network databases

• Frequency of data reconciliation
• Metrics for data quality
• Performance of reconciliation process (real-time requirements)