CIDR 2013 Paper  
Data Curation at Scale: The Data Tamer System, CIDR 2013  
Authors: Stonebraker et al

Summary

The authors introduced a novel framework called Data Tamer for data curation. The primary focus of the paper is on integrating different data sources and remove duplicate results. Even though there exist a significantly large amount of previous work in this area, this is the first fully integrated end-to-end system, as claimed by the authors, which is also scalable. The three data sets experimented on were Web URLs/documents, notebooks of lab scientists (spreadsheets), and client record of insurance companies.

Key ideas in the paper

- Consider that after data conversion, transformation we have various sources of structured and/or semi structured data.
- Assume some partial information is always available; that is there are specific attributes that are always present together for some classes of entities (in a domain).
- Assume there are some known standard dictionaries serving as authoritative tables containing
- The Domain Experts are the classifiers/entities with expertise. The idea is boosting performance using multiple experts/classifiers. For instance, in schema integration, experts are fuzzy string comparisons, TF-IDF cosine similarity, Minimum description length, Welsch's t-test (A human administrator must assign a threshold value for those classifiers).
- We use training data that contain “known duplicates” and “known entities that are not duplicates” to train the classifiers. Clustering and naïve Baysian are the main methods in detecting known duplicates.
- Crowd sourcing: In addition, there are human experts whose purpose is to classify entities in ambiguous cases (the value is close to the threshold or classifiers do not have a consensus.) The authors also suggested a reward-payment system for correct response from the human experts.

Positive/Strong Points

- S1: The framework utilizes many feature extraction techniques to compare records. Boosting definitely has a great effect on the accuracy.
- S2: The attribute mapping in schema integration is a fairly fast method using heuristics. The two-pass algorithm reduces the number of records we have to look at, since the first pass works as a good filter. The time complexity is at most quadratic in term of the number of values in the table columns.
- S3: The framework has been adopted by three companies, which proves that it is practical and fairly accurate.
- S4: The authors introduced several micro-optimization techniques for different cases of rebuilding entity consolidation/schemas when new data comes in. We do not have to process everything from scratch.
Negative/Weak Points
- W1: The authors made some very strong assumptions about the input to the system. For instance, for many real work problems (such as web crawling/ text mining), “wrapper” or data extraction is given, even though in fact it is not an easy task and usually, extraction and transformation are performed in combination.
- W2: How can the DTA (administrator) set up/choose the threshold for the experts/classifiers? There are a lot of parameters to handle in this case (and in addition, each classifier has totally different sets of parameters).
- W3: The validation part is not very persuasive. The authors do not know the ground truth in data deduplication part, but that does not justify the precision of 100% since they are defining their own experimental result as the ground truth.

Research Questions and Points for Discussion
- D1: What are main parameters for adjusting precision vs. recall in duplicate detection?
- D2: As indicated by the authors, how can data cleaning (missing values, outliers) be handled before the integration/mapping step?
- D3: How can we parallelize the integration/duplicate detection process? Since we are talking about scalability, what is the bottleneck in runtime? Will different classifiers be run on different nodes on a cluster? How can we speed up slower classifiers?
- D4: How can we artificially generate large datasets of different schemas to test the integration technique mentioned in the paper?
- D5: In the record clustering and consolidation, we might be performing clustering on a huge number of tuple pairs (millions of records could indicate more than billions of possible tuple duplicates, so is clustering using a similarity graph practical in this case? Or is the number of duplicates reduced first by setting the threshold high?)