Record Linkage: Similarity Measures and Algorithms

Nick Koudas
University of Toronto
koudas@cs.toronto.edu

Sunita Sarawagi
IIT Bombay
sunita@it.iitb.ac.in

Divesh Srivastava
AT&T Labs–Research
divesh@research.att.com

1. MOTIVATION

The quality of the data residing in databases gets degraded due to a multitude of reasons. Such reasons include typing mistakes (e.g., lexicographical errors, character transpositions) during insertion, lack of standards for recording database fields (e.g., person names, addresses), and various errors introduced by poor database design (e.g., update anomalies, missing key constraints). Data of poor quality can result in significant impediments to popular business practices: sending products or bills to incorrect addresses, inability to locate customer records during service calls, or inability to correlate customers across multiple services, etc.

In the presence of data quality errors, a central problem is the ability to identify whether two entities (e.g., relational tuples) are *approximately* the same. Depending on the type of data under consideration, various "similarity metrics" (approximate match predicates) have been defined to quantify the closeness of a pair of data entities in a way that common mistakes are captured. Given any specific similarity metric, a key algorithmic problem in this context is the approximate join problem: given two large multi-attribute data sets, identify all pairs of entities (tuples) in the two sets that are approximately the same. This problem is by no means new. Over the years various communities, including the statistics, machine learning, information retrieval, and database communities, have addressed many aspects of the problem, referring to it by a variety of names, including record linkage, entity identification, entity reconciliation and approximate join. We shall make use of the names *record linkage* and *approximate join* in the sequel. The approximate join operation is often followed by a post-processing phase where the tuple pairs produced as join results are used to cluster together tuples that refer to the same entity while minimizing the number of join pairs that get violated. These clusters define the desired entity boundaries. Given the significance and the inherent difficulty of the record linkage problem, a plethora of techniques have been developed in various communities, deploying diverse approximate match predicates.

The objectives of this tutorial are to: (i) formally define the various flavors of the record linkage problem, (ii) identify and compare the various approximate match predicates for attribute value pairs that have been introduced over the years, (iii) identify and contrast the various methodologies to combine approximate match predicates for approximate match between tuple pairs, (iv) provide a comprehensive and cohesive overview of the key research results, techniques, and tools used for record linkage, (v) present clustering paradigms for consistent partitioning of tuples to identify entities, and (vi) identify key research areas where further work is required.

Recent tutorials in the area of data quality [2, 1] present broad overviews of various aspects of data quality and do not delve into the details of record linkage technology. Our tutorial is a significant extension of a shorter tutorial on this topic [3] and aims to provide a comprehensive overview of this fundamental area of data quality.

SIGMOD 2006, June 27–29, 2006, Chicago, Illinois, USA.

2. TUTORIAL OUTLINE

Our tutorial is example driven, and begins by introducing the various issues and problems that one has to cope with when designing record linkage technology. We then present and categorize typical "errors" encountered in real operational databases using concrete examples; such examples motivate the need for the methodologies we shall introduce in the sequel. We shall also define formally the various flavors of the record linkage problem as optimization problems. The bulk of the tutorial is organized as follows.

2.1 Approximate Match Predicates

We shall review a variety of approximate match predicates that have been proposed to quantify the degree of similarity or closeness of two data entities. We shall compare and contrast them based on their applicability to various data types, algorithmic properties, computational overhead and their adaptability. Most approximate match predicates return a score between 0 and 1 (with 1 being assigned to identical entities) that effectively quantifies the degree of similarity between data entities. Our presentation of such approximate match predicates will consist of three parts.

- **Atomic Similarity Measures**: This part will review measures to assess atomic (attribute value) similarity between a pair of data entities. We shall cover several approximate match predicates in detail including edit distance, phonetic distance (soundex), the Jaro and Winkler measures, tf.idf and many variants thereof. Several approaches to fine tune parameters of such measures will be presented.

- **Functions to combine similarity measures**: In this part, we shall first review techniques dealing with the following basic decision problem: Given a set of pairs of attributes belonging to two entities (tuples), in which each pair is tagged with it's own approximate match score (possibly applying distinct approximate match predicates for each attribute pair), how does one combine such scores to decide whether the entire entities (tuples) are approximately the same. For this basic decision

problem, we shall review an array of proposed methodologies. These include statistical and probabilistic, predictive, cost based, rule based, user assisted as well as learning based methodologies. Moreover, we shall cover several specific functions including Naive Bayes, the Fellegi-Sunter model, linear support vector machines (SVM) and approaches based on voting theory.

- **Similarity between linked entities**: Often the entities over which we need to resolve duplicates are linked together via foreign keys in a multi-relational database. Links might be of various types, including associative links arising out of co-occurrence in a larger context (e.g., co-authors of a paper), or structural links denoting containment. An inter-linked set of entities calls for richer similarity measures that capture the similarity of the context in which the entity pair appears. We shall present various graph-based similarity measures that capture transitive contextual similarity in combination with the intrinsic similarity between two entities.

We shall compare and contrast these similarity measures based on their features, their complexity, scalability and applicability to various domains. We shall present some optimality results and comment on their applicability in the context of record linkage.

2.2 Record Linkage Algorithms

Once the basic techniques for quantifying the degree of approximate match for a pair (or subsets) of attributes have been identified, the next challenging operation is to embed them into an approximate join framework between two data sets. This is a non-trivial task due to the large (quadratic in the size of the input) number of pairs involved. We shall present a set of algorithmic techniques for this task. A common feature of all such algorithms is the ability to keep the total number of pairs (and subsequent decisions) low utilizing various pruning mechanisms. These algorithms can be classified into two main categories.

- Algorithms inspired by relational duplicate elimination and join techniques including sort-merge, band join and indexed nested loops. In this context, we shall review techniques like Merge/Purge (based on the concept of sorted neighborhoods), BigMatch (based on indexed nested loops joins) and Dimension Hierarchies (based on the concept of hierarchically clustered neighborhoods).

- Algorithms inspired by information retrieval that treat each tuple as a *set* of tokens, and return those set pairs whose (weighted) overlap exceeds a specified threshold. In this context, we shall review a variety of set join algorithms.

We shall also discuss two alternative ways to realize (i.e., implement) approximate join techniques. The first is concerned with procedural algorithms operating on data, applying approximate match predicates, without a particular storage or query model in mind. The second is concerned with declarative specifications of data cleaning operations. Both approaches have their relative strengths and weaknesses. A non-declarative specification offers greater algorithmic flexibility and possibly improved performance (e.g., implemented on top of a file system without incurring RDBMS overheads). A declarative specification offers unbeatable ease of deployment (as a set of SQL queries), direct processing of data in their native store (RDBMS) and flexible integration with existing applications utilizing an RDBMS. The choice between the two realizations clearly depends on application constraints and requirements. We shall review both approaches describing a concrete set of methodologies that can assist in each approach. We shall review various data quality tools that deploy record linkage technology and discuss their key functionalities in this area. Where publicly known, we shall discuss specific algorithms they deploy.

2.3 Creation of Data Partitions

The output of the approximate join needs to be post processed to cluster together all tuples that refer to the same entity. The approximate join operation above might produce seemingly inconsistent results like tuple A joins with tuple B, tuple A joins with tuple C, but tuple B does not join with tuple C. A straightforward way to resolve such inconsistencies is to cluster together all tuples via a transitive closure of the join pairs. In practice, this can lead to extremely poor results since unrelated tuples might get connected through noisy links. A number of approaches have been proposed for tackling this problem. In particular, we shall present a newly introduced clustering paradigm called Correlation Clustering for minimizing the number of join pairs violated in creating clusters.

Finally, we shall identify key research questions pertinent to each part of our tutorial. Our tutorial will also contain an extensive survey of related bibliography from various communities.

3. PROFESSIONAL BIOGRAPHIES

Nick Koudas is a faculty member at the University of Toronto, Department of Computer Science. He holds a Ph.D. from the University of Toronto, an M.Sc. from the University of Maryland at College Park, and a B.Tech. from the University of Patras in Greece. He serves as an associate editor for the Information Systems journal, the IEEE TKDE journal and the ACM Transactions on the WEB. He is the recipient of the 1998 ICDE Best Paper award. His research interests include core database management, data quality, metadata management and its applications to networking.

Sunita Sarawagi researches in the fields of databases, data mining, machine learning and statistics. She is an associate professor at the Indian Institute of Technology, Bombay. Prior to that she was a research staff member at IBM Almaden Research Center. She got her Ph.D. in databases from the University of California at Berkeley and a B.Tech. from the Indian Institute of Technology, Kharagpur. She has several publications in databases and data mining, including a best paper award at the 1998 ACM SIGMOD conference, and several patents. She is on the editorial board of the ACM TODS and ACM KDD journals and editor-in-chief of the ACM SIGKDD newsletter. She has served as program committee member for ACM SIGMOD, VLDB, ACM SIGKDD and IEEE ICDE, ICML conferences.

Divesh Srivastava is the head of the Database Research Department at AT&T Labs-Research. He received his Ph.D. from the University of Wisconsin, Madison, and his B.Tech. from the Indian Institute of Technology, Bombay. His current research interests include data quality, data streams and XML databases.

4. REFERENCES

[1] C. Batini, T. Catarci, and M. Scannapieco. A survey of data quality issues in cooperative information systems. *Pre-conference ER tutorial*, 2004.

[2] T. Johnson and P. Dasu. Data quality and data cleaning: An overview. *SIGMOD tutorial*, 2003.

[3] N. Koudas and D. Srivastava. Approximate joins: concepts and techniques. *VLDB tutorial*, 2005.