Data Fusion – Resolving Data Conflicts for Integration

Xin Luna Dong
AT&T Labs-Research
Florham Park, NJ, USA
lunadong@research.att.com

Felix Naumann
Hasso Plattner Institute (HPI)
Potsdam, Germany
naumann@hpi.uni-potsdam.de

1. MOTIVATION

The amount of information produced in the world increases by 30% every year and this rate will only go up. With advanced network technology, more and more sources are available either over the Internet or in enterprise intranets. Modern data management applications, such as setting up Web portals, managing enterprise data, managing community data, and sharing scientific data, often require integrating available data sources and providing a uniform interface for users to access data from different sources; such requirements have been driving fruitful research on data integration over the last two decades [11, 13].

Data integration systems face two folds of challenges. First, data from disparate sources are often heterogeneous. Heterogeneity can exist at the schema level, where different data sources often describe the same domain using different schemas; it can also exist at the instance level, where different sources can represent the same real-world entity in different ways. There has been rich body of work on resolving heterogeneity in data, including, at the schema level, schema mapping and matching [14], model management [1], answering queries using views [12], data exchange [8], and at the instance level, record linkage (entity resolution, object matching, reference linkage, etc.) [7, 15], string similarity comparison [4], etc.

Second, different sources can provide conflicting data. Conflicts can arise because of incomplete data, erroneous data, and out-of-date data. Returning incorrect data in a query result can be misleading and even harmful: one may contact a person by an out-of-date phone number, visit a clinic at a wrong address, and even make poor business decisions. It is thus critical for data integration systems to resolve conflicts from various sources and identify true values. This problem becomes especially prominent with the ease of publishing and spreading false information on the Web.

This tutorial focuses on *data fusion*, which addresses the second challenge by fusing records on the same real-world entity into a single record and resolving possible conflicts from different data sources. Data fusion plays an important role in data integration systems: it detects and removes dirty data and increases correctness of the integrated data.

The main objective of the tutorial is to gather models, techniques, and systems of the wide but yet unconsolidated field of data fusion and present them in a concise and consolidated manner. In the tutorial we provide an overview of the causes and challenges of data fusion. We cover a wide set of both simple and advanced techniques to resolve data conflicts in different types of settings and systems. Finally, we provide a classification of existing information management systems with respect to their ability to perform data fusion.

The tutorial is based on a recent survey on data fusion [3] and various techniques proposed for truth discovery, including, but not limited to, [2, 5, 6, 16, 18].

VLDB '09, August 24-28, 2009, Lyon, France

2. TUTORIAL OUTLINE

Our tutorial begins with an overview of the importance of data fusion in data integration and possible reasons for data conflicts. We then present a classification of existing data fusion techniques and introduce relational operations for conflict resolution. After that, we describe several advanced techniques for finding the best (true) values in presence of data conflicts. We end our tutorial with surveying data fusion techniques in existing data integration systems and suggesting future research directions.

2.1 Overview

Data integration has three broad goals: increasing the *completeness, conciseness,* and *correctness* of data. *Completeness* measures the amount of data, in terms of both the number of tuples and the number of attributes. *Conciseness* measures the uniqueness of object representations in the integrated data, in terms of both the number of unique objects and the number of unique attributes of the objects. Finally, *correctness* measures correctness of data; that is, whether the data conform to the real world. Data fusion, which is the focus of this tutorial, aims at resolving conflicts from data and increasing correctness of data.

We distinguish two kinds of data conflicts: *uncertainty* and *contradiction*. *Uncertainty* is a conflict between a non-null value and one or more null values that are all used to describe the same property of a real-world entity. Uncertainty is caused by missing information, such as null values in a source or a completely missing attribute in a source. *Contradiction* is a conflict between two or more different non-null values that are all used to describe the same property of the same entity. Contradiction is caused by different

sources providing different values for the same attribute of a real-world entity.

There are two key issues in data fusion. First, how to find the best values among conflicting values? Second, how to do so efficiently?

2.2 Conflict resolution and data merging

There are many different data integration and fusion systems, each with their own solution. In the tutorial we classify and describe existing strategies to approach data conflicts. In particular, *Conflict ignoring* strategies are not aware of conflicts, perform no resolution, and thus may produce inconsistent results. *Conflict avoiding* strategies are aware of conflicts, but make simple decisions to avoid conflicts rather than perform individual resolution for each conflict. Finally, *conflict resolving* strategies provide the means for individual fusion decisions for each conflict. Such decisions can be *instance-based* or *metadata-based*. Finally, strategies can be classified by the result they are able to produce: *Deciding strategies* choose a preferred value among the existing values, while *mediating strategies* can produce an entirely new value, such as the average of a set of conflicting numbers.

Relational operators, such as *join* and *union* (and their relatives), already perform data fusion of sorts. Further, *full disjunction* combines two or more input relations by combining all matching tuples into a single result-tuple [9]. A slight enhancement is given by the *minimum union* and *complementation* operations, which remove additional redundant tuples. Further operators such as *match-join* [17] or *prioritized merge* [10], go beyond removing uncertainties by resolving contradictions. Finally, we discuss fusion using SQL-based techniques, such as user-defined-functions, the coalesce function, and aggregation functions.

2.3 Advanced techniques

We next describe several advanced techniques that consider accuracy of sources, freshness of sources, and dependencies between sources to solve the problems.

Accuracy: Data sources are of different accuracy and some are more trustworthy. It is proposed in [5, 16, 18] that we should consider accuracy of sources when deciding the true values. We describe their probabilistic models that iteratively compute source accuracy and decide the true values.

Freshness: The world often changes dynamically and a value, in addition to being true or false, can be in a subtle third case: *out-of-date*. It is proposed in [6] that one should consider *freshness* of sources (staleness of data) and treat incorrect values and out-of-date values differently in truth discovery. We describe their probabilistic model.

Dependency: In many domains, especially on the Web, data sources may copy from each other for some of their data. It is proposed in [2, 5, 6] that we should consider dependence between sources in truth discovery. We describe their algorithms that iteratively detect dependence between sources and discover the true values taking into consideration such dependence.

2.4 Data fusion in existing DI systems

This part of the tutorial examines relevant properties of both commercial and prototypical data fusion systems. Among the analyzed research prototypes with some fusion capabilities are Multibase, Hermes, FusionPlex, HumMer, Ajax, TSIMMIS, SIMS, Ariadne, ConQuer, Infomix, HIPPO, and Rainbow (see [3] for references). Among the analyzed commercial data integration systems are several DBMS and ETL tools, such as IBM's Information Server or Microsoft's SQL Server Integration Services.

2.5 Open problems

We conclude the tutorial with a discussion of open problems and desiderata for data fusion systems, including *complex fusion functions*, techniques for *incremental fusion* and *online fusion*, the better inclusion of *data lineage*, and the *combination of truth discovery and record linkage*.

3. REFERENCES

[1] P. A. Bernstein and S. Melnik. Model management 2.0: manipulating richer mappings. In *Proc. of SIGMOD*, pages 1–12, 2007.

[2] L. Berti-Equille, A. D. Sarma, X. L. Dong, A. Marian, and D. Srivastava. Sailing the information ocean with awareness of currents: Discovery and application of source dependence. In *CIDR*, 2009.

[3] J. Bleiholder and F. Naumann. Data fusion. *ACM Computing Surveys*, 41(1):1–41, 2008.

[4] W. W. Cohen, P. Ravikumar, and S. E. Fienberg. A comparison of string distance metrics for name-matching tasks. In *Proc. of IIWEB*, pages 73–78, 2003.

[5] X. L. Dong, L. Berti-Equille, and D. Srivastava. Integrating conflicting data: the role of source dependence. *PVLDB*, 2(1), 2009.

[6] X. L. Dong, L. Berti-Equille, and D. Srivastava. Truth discovery and copying detection in a dynamic world. *PVLDB*, 2(1), 2009.

[7] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 19(1):1–16, 2007.

[8] R. Fagin, P. G. Kolaitis, and L. Popa. Data exchange: Getting to the core. *ACM Transactions on Database Systems (TODS)*, 30(1):174–201, 2005.

[9] C. A. Galindo-Legaria. Outerjoins as disjunctions. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, pages 348–358, Minneapolis, Minnesota, May 1994.

[10] S. Greco, L. Pontieri, and E. Zumpano. Integrating and managing conflicting data. In *Revised Papers from the 4th International Andrei Ershov Memorial Conference on Perspectives of System Informatics*, pages 349–362, 2001.

[11] L. M. Haas. Beauty and the beast: The theory and practice of information integration. In *Proc. of ICDT*, pages 28–43, 2007.

[12] A. Y. Halevy. Answering queries using views: A survey. *VLDB Journal*, 10(4):270–294, 2001.

[13] A. Y. Halevy, A. Rajaraman, and J. J. Ordille. Data integration: The teenage years. In *Proc. of VLDB*, pages 9–16, 2006.

[14] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334–350, 2001.

[15] W. Winkler. Overview of record linkage and current research directions. Technical report, Statistical Research Division, U. S. Bureau of the Census, 2006.

[16] M. Wu and A. Marian. Corroborating answers from multiple web sources. In *Proc. of WebDB*, 2007.

[17] L. L. Yan and M. T. Özsu. Conflict tolerant queries in AURORA. In *Proceedings of the International Conference on Cooperative Information Systems (CoopIS)*, pages 279–290, 1999.

[18] X. Yin, J. Han, and P. S. Yu. Truth discovery with multiple conflicting information providers on the web. In *Proc. of SIGKDD*, 2007.