Data Profiling – A Tutorial

Ziawasch Abedjan
TU Berlin,
Germany
abedjan@tu-berlin.de

Lukasz Golab
University of Waterloo,
Canada
lgolab@uwaterloo.ca

Felix Naumann
Hasso Plattner Institute
Potsdam, Germany
felix.naumann@hpi.de

ABSTRACT

One of the crucial requirements before consuming datasets for any application is to understand the dataset at hand and its metadata. The process of metadata discovery is known as data profiling. Profiling activities range from ad-hoc approaches, such as eye-balling random subsets of the data or formulating aggregation queries, to systematic inference of structural information and statistics of a dataset using dedicated profiling tools. In this tutorial, we highlight the importance of data profiling as part of any data-related use-case, and we discuss the area of data profiling by classifying data profiling tasks and reviewing the state-of-the-art data profiling systems and techniques. In particular, we discuss hard problems in data profiling, such as algorithms for dependency discovery and profiling algorithms for dynamic data and streams. We also pay special attention to visualizing and interpreting the results of data profiling. We conclude with directions for future research in the area of data profiling. This tutorial is based on our survey on profiling relational data [2].

1. INTRODUCTION

We can safely assume that most computer or data scientists have engaged in the activity of data profiling, at least by "eye-balling" spreadsheets, database tables, XML files, etc. More advanced techniques may have been used, such as keyword-searching in datasets, writing structured queries, or even using dedicated data profiling tools. Data profiling is the set of activities and processes to determine the metadata about a given dataset. Among the simpler results are per-column statistics, such as the number of null values and distinct values in a column, its data type, or the most frequent patterns of its data values. Metadata that are more difficult to compute involve multiple columns, such as inclusion, functional and order dependencies.

Traditional use cases of data profiling include data exploration, data cleansing, and data integration. Statistics about data are also useful in query optimization. Additionally, domain-specific use cases have emerged in scientific data management and big data analytics. In particular, "big data", with their high volume, high velocity, and high variety [26], are data that cannot be managed with traditional techniques. Fetching, storing, querying, and integrating big data is expensive, despite many modern technologies. Thus, data profiling gains a new importance. For instance, before exposing an infrastructure to the Twitter firehose, it might be worthwhile to find out the properties of the data one is receiving; before downloading significant parts of the linked data cloud, some prior sense of the integration effort is needed; before augmenting a warehouse with text mining results, an understanding of data quality is required. In general, many big data and related data science scenarios call for data mining and machine learning techniques to explore and mine data. Again, data profiling is an important preparatory task to determine which data to mine, how to import data into various tools, and how to interpret the results [31].

Leading researchers have noted that "*if we just have a bunch of datasets in a repository, it is unlikely anyone will ever be able to find, let alone reuse, any of this data. With adequate metadata, there is some hope, but even so, challenges will remain*[...]." [7] Data profiling addresses precisely this problem and faces three challenges:

1. Managing the input
2. Performing the computation
3. Managing the output

Apart from typical data formatting issues, the first challenge includes the problem of specifying the expected outcome, i.e., determining which profiling tasks to execute on which parts of the data. In fact, many tools require a precise specification of what to inspect. Other approaches are more open and perform a wider range of tasks, discovering all metadata automatically.

Data profiling tools and algorithms have tackled these challenges in different ways. Many rely on the capabilities of the underlying DBMS, as many profiling tasks can be expressed as SQL queries. Others have developed innovative ways to handle the individual challenges, for instance using indexing schemes, parallel processing, and reusing intermediate results. Several methods have been proposed that deliver only approximate results for various profiling tasks, for instance by profiling samples. Finally, users may be asked to narrow down the discovery process to certain columns or tables. For instance, there are tools that verify inclusion dependencies on user-suggested pairs of columns, but cannot

SIGMOD'17, May 14-19, 2017, Chicago, IL, USA

 ISBN 978-1-4503-4197-4/17/05...$15.00

DOI: http://dx.doi.org/10.1145/3035918.3054772

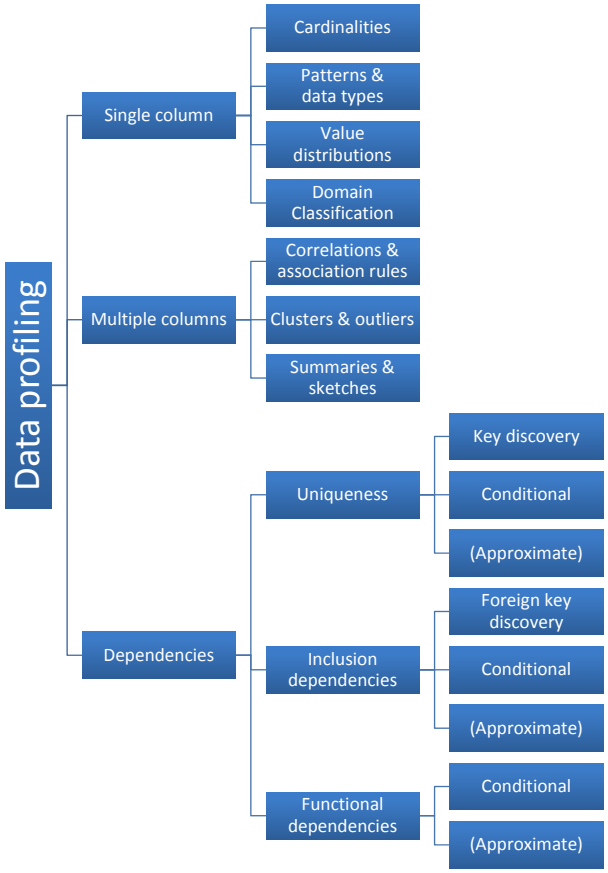

Figure 1: A classification of traditional data profiling tasks [2].

automatically check inclusion between all pairs of columns or column sets.

This tutorial is based on our survey on data profiling [2]. We discuss data profiling use cases, present a classification of data profiling tasks (summarized in Figure 1), and focus on the second and third challenges from the above list (performing the computation, and managing and interpreting the output). The computational complexity of data profiling algorithms depends on the number or rows, with a sort being a typical operation, but also on the number of columns. Many tasks need to inspect all column combinations, i.e., they are exponential in the number of columns. In addition, the scalability of data profiling methods is important, as the ever-growing data volumes demand disk-based and distributed processing. Furthermore, any discovered metadata refer only to the given data instance and cannot be used to derive schematic/semantic properties with certainty, such as value domains, primary keys, or foreign key relationships. Thus, profiling results need interpretation, which is usually performed by database and domain experts.

A typical informal workflow for data profiling techniques is shown in Figure 2. Starting from a set of unfamiliar datasets, often referred to as a data lake, initial data profiling steps can help explore the data and possibly identify relevant datasets. The profiling tasks applied here are usually of linear complexity to cope with the very large volumes in typical data lakes. Further and more complex profiling tasks support users such as data scientists, who want to understand and assess the quality of the smaller selected datasets. After appropriate cleansing and transformation steps, which are mentioned but not covered by our tutorial, specific data profiling techniques can be used to support various use cases, such as improving query optimization, and generating integrity constraints.

2. TUTORIAL OUTLINE

In this section, we present the structure of this 1.5 hour tutorial, outlining the scope and depth of each subtopic. The work presented here is an extension of our previous tutorial [3], which focused on algorithms, especially those for dependency discovery. This tutorial includes algorithmic content, but also focuses on the equally important problem of visualizing and interpreting the results of data profiling. This aspect includes human-in-the-loop techniques, schema visualization, and ranking the discovered dependencies and candidate keys according to some measures of interestingness or informativeness. We also include new results, such as algorithms for discovering order dependencies [27,34] and systems for finding interesting datasets in very large databases via profiling [18].

2.1 Motivation and current profiling systems

We motivate the problem of data profiling with real life scenarios from data integration, data exploration, and data management. In particular, we present a definition that separates data profiling as such from data mining and data exploration. Furthermore, we discuss the capability of state-of-the-art data profiling tools from industry and research [4, 13, 17, 20, 23, 28, 32]. Because data profiling is such an important capability for many data management tasks, there are various commercial data profiling tools. In many cases, they are a part of a data quality / data cleansing tool suite, to support the use case of profiling for frequent patterns or rules and then cleaning those records that violate them. In addition, most Extract-Transform-Load tools have some profiling capabilities.

In the research literature, data profiling tools are often embedded in data cleaning systems. For example, the Bellman [17] data quality browser supports column analysis (counting the number of rows, distinct values, and NULL values, finding the most frequently occurring values, etc.), and key detection (up to four columns). Furthermore, an interesting application of Bellman was to profile the evolution of a database using value distributions and correlations [16]: which tables change over time and in what ways (insertions, deletions, modifications), and which groups of tables tend to change in the same way. The Potters Wheel tool [32] also supports column analysis, in particular, detecting data types and syntactic structures/patterns. We discuss the pros and cons of these and other related systems, and identify easier and more challenging profiling tasks.

2.2 Classification of profiling tasks

We then categorize data profiling tasks as shown in Figure 1. We classified the tasks according to their dimensional complexity. Single column profiling refers to the analysis of values in a single column, and ranges from simple counts and aggregation functions to distribution analysis and discovery of patterns and data types. Multi-column profiling is the set of activities that can be applied to a single col-

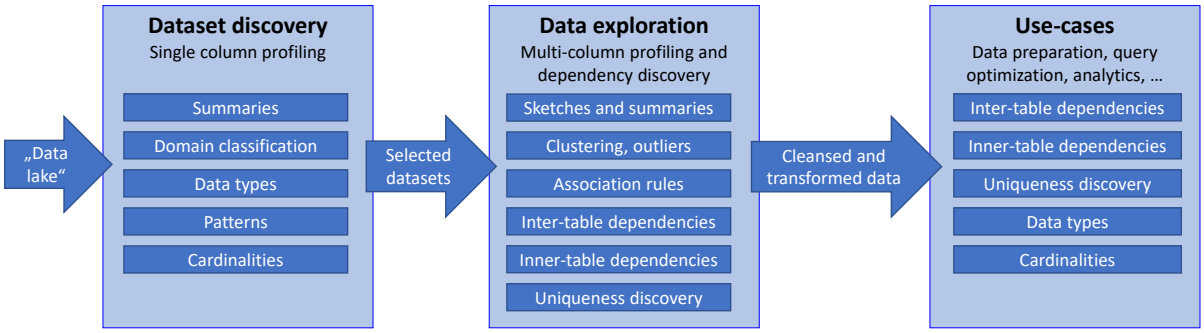

Figure 2: Typical steps during of a data profiling workflow.

umn but allows for the analysis of inter-value dependencies across columns, resulting in association rules, clustering, and outlier detection. Finally, we describe the class of meta-data that constitute dependencies describing relationships among columns of a single table, such as keys and functional dependencies, and relationships across multiple tables, such as foreign keys and inclusion dependencies [2]. We address relevant concepts, such as partial dependencies and approximate solutions, before we discuss the challenging aspects of dependency discovery.

2.3 Single- and multi-column analysis

We begin our discussion of data profiling algorithms with those for single and multi column analysis. We overview distribution and outlier analysis, data summaries, sketches and signatures, pattern discovery, characterizing missing and default values, as well as clustering and association analysis over multiple columns.

2.4 Dependency discovery

We then zoom in on dependency discovery, and give an in-depth technical description of strategies that tackle the exponential complexity of dependency discovery tasks. An important concept to discuss here is the concept of "minimality" that reduces the result set of a dependency discovery task to only non-redundant dependencies. In particular, we discuss traditional apriori-based approaches [11, 24] for pruning the search space of attribute combinations and present new algorithms [5,6,22,29] that significantly outperform traditional approaches through improved pruning techniques. Here we categorize existing algorithms into two major classes: row-based and column-based algorithms. Row-based algorithms process the set of candidate dependencies row by row and check which dependencies still hold. Column-based approaches generate candidate dependencies of a certain size (i.e., a certain number of columns), validate them through scanning the whole database, and then generate a new candidate set by expanding current candidates with more columns.

We further explore the area of dependency discovery by revisiting variations of dependencies and algorithms. In particular, we address approximate algorithms that discover dependencies that may not hold on the entire dataset. Those are discovered using sampling [25] or summarization techniques [15]. The benefit of approximate solutions is that they often are significantly faster to compute than exact approaches, yet they can be used in many scenarios where inaccuracies are tolerated, e.g., in dirty datasets where glitches are expected. Furthermore, we address conditional dependency discovery, whose goal is to identify dependencies as well as conditions which specify subsets of the data where the given dependency holds [10, 12].

Finally, in addition to "classical" dependencies, such as functional and inclusion dependencies, we discuss discovering other types of dependencies that are relevant to data profiling. These include order dependencies [35], denial constraints [14], differential dependencies [33], sequential dependencies [19], and temporal rules [1].

2.5 Exploratory profiling and result interpretation

Previous research on profiling, such as dependency discovery, focused on the design of algorithms that can tame the complexity of the search space. Recent efforts try to organize the generated metadata for specific use cases. In this tutorial, we will discuss new approaches for visualizing and ranking the results of data profiling, such as the discovered functional dependencies and candidate keys. For instance, Andritsos et al. rank FDs used for normalization by the entropy of their attribute sets: The more duplication an FD removes, the "better" it is [8]. Novel FD discovery approaches also benefit from these insights by pruning the search space accordingly [30]. Finally, Figure 3 shows our own rendering of a large FD set.

Furthermore, we will discuss recent systems that use efficient profiling techniques for data discovery. For example, the *Data Civilizer* system leverages profiling signatures to obtain a metadata graph that enables the discovery of datasets that are related, similar, or joinable [18]. Another project that leverages metadata to improve the data discovery process is the GOODs project [21]. This system extracts and aggregates metadata to identify similar or related datasets. GOODs manages and discovers different categories of metadata, from basic metadata that encompass data size, format, and aliases over user-supplied annotations, to temporal and provenance related metadata. A similar open-source project for managing data and metadata is Atlas [9], which provides data visualization and browsing as part of a Hadoop framework. In all three systems, scalable profiling and metadata management are critical to enable an interactive data browsing and discovery experience.

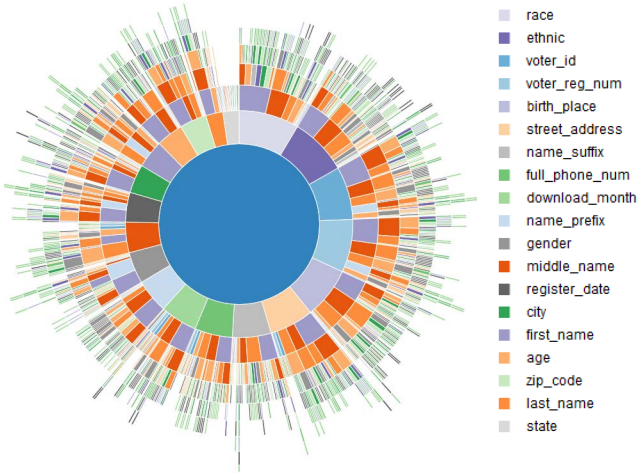

Figure 3: A sunburst diagram of many FDs. The central circle represents a dependent attribute and the rays represent various referencing attribute sets.

2.6 Open problems and directions for future research in data profiling

Recent trends in data management have led to new challenges but also new opportunities for data profiling. Under the *big data* umbrella, industry and research have turned their attention to data they do not own or have not made use of yet. Much of the data that shall be used are of non-traditional type for data profiling, i.e., non-relational, non-structured (textual), and heterogeneous. Many existing profiling methods cannot adequately handle these kinds of data: Either they do not scale well, or there simply are no methods yet. We discuss some of these trends and their implications on data profiling. In particular, we address specific challenges in profiling dynamic data, i.e., data that change and make previously obtained meta-data obsolete or data that are consumed as a continuous stream. Here, some of the challenges include efficient incremental profiling and tracking how metadata change over time (e.g., are some dependencies valid only in certain time intervals?). Furthermore, we address the peculiarities of data profiling when data reside in various non-relational formats, such as XML, RDF, or structure-less text documents.

Finally, a substantial challenge in the area of data profiling is the effective visualization and interpretation of the generated meta-data. Effective visualization of meta-data requires not only effective visualization technologies but also methods and metrics for appropriate selection of the meta-data at hand. While a dataset might contain thousands of syntactically valid dependencies, it is important to identify the top-k most interesting dependencies that can be presented to the user. We hope that this tutorial encourages a stronger cooperation between the database and the visualization community.

3. SPEAKER BIOGRAPHIES

Ziawasch Abedjan studied IT Systems Engineering at the Hasso Plattner Institute in Potsdam, Germany, where he received his bachelor's degree in 2008 and his master's degree in 2010. In 2014, he graduated as a member of the "HPI Research School on Service Oriented Computing" and joined the database group at MIT CSAIL as a postdoctoral associate. Since 2016, he is assistant professor at the TU Berlin, heading the Big Data Management Group. His research interests include data integration, data mining, and data cleansing.

Lukasz Golab is an Associate Professor and Canada Research Chair at the University of Waterloo. Prior to joining Waterloo, he was a Senior Member of Research Staff at AT&T Labs. He holds a B.Sc. from the University of Toronto (2001) and a Ph.D. from the University of Waterloo (2006). Lukasz's current research focuses on data stream management, data quality, data analytics for sustainability, and educational data mining.

Felix Naumann studied mathematics at the Technical University of Berlin and received his diploma in 1997. As a member of the graduate school "Distributed Information Systems" at Humboldt-University of Berlin, he finished his PhD thesis in 2000. In the following two years Felix Naumann worked at the IBM Almaden Research Center. From 2003-2006 he was an assistant professor at Humboldt-University. Since 2006 he is a full professor at the Hasso Plattner Institute heading the information systems group and working in the areas of data integration, data profiling, and text mining.

4. REFERENCES

[1] Z. Abedjan, C. Akcora, M. Ouzzani, P. Papotti, and M. Stonebraker. Temporal rules discovery for web data cleaning. *Proceedings of the VLDB Endowment (PVLDB)*, 9(4):336–347, 2015.

[2] Z. Abedjan, L. Golab, and F. Naumann. Profiling relational data: a survey. *VLDB Journal*, 24(4):557–581, 2015.

[3] Z. Abedjan, L. Golab, and F. Naumann. Data profiling (tutorial). In *Proceedings of the International Conference on Data Engineering (ICDE)*, pages 1432–1435, 2016.

[4] Z. Abedjan, T. Grütze, A. Jentzsch, and F. Naumann. Profiling and mining RDF data with ProLOD++. In *Proceedings of the International Conference on Data Engineering (ICDE)*, pages 1198–1201, 2014. Demo.

[5] Z. Abedjan, J.-A. Quiané-Ruiz, and F. Naumann. Detecting unique column combinations on dynamic data. In *Proceedings of the International Conference on Data Engineering (ICDE)*, pages 1036–1047, 2014.

[6] Z. Abedjan, P. Schulze, and F. Naumann. DFD: Efficient functional dependency discovery. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pages 949–958, 2014.

[7] D. Agrawal, P. Bernstein, E. Bertino, S. Davidson, U. Dayal, M. Franklin, J. Gehrke, L. Haas, A. Halevy, J. Han, H. V. Jagadish, A. Labrinidis, S. Madden, Y. Papakonstantinou, J. M. Patel, R. Ramakrishnan, K. Ross, C. Shahabi, D. Suciu, S. Vaithyanathan, and J. Widom. Challenges and opportunities with Big Data. Technical report, Computing Community Consortium, http://cra.org/ccc/docs/init/bigdatawhitepaper.pdf, 2012.

[8] P. Andritsos, R. J. Miller, and P. Tsaparas. Information-theoretic tools for mining database structure from large data sets. In *Proceedings of the International Conference on Management of Data (SIGMOD)*, pages 731–742, 2004.

[9] Apache. ATLAS: Data governance and metadata framework for Hadoop. http://atlas.incubator.apache.org, 2016. [Online; accessed 25-October-2016].

[10] J. Bauckmann, Z. Abedjan, H. Müller, U. Leser, and F. Naumann. Discovering conditional inclusion dependencies. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pages 2094–2098, 2012.

[11] J. Bauckmann, U. Leser, F. Naumann, and V. Tietz. Efficiently detecting inclusion dependencies. In *Proceedings of the International Conference on Data Engineering (ICDE)*, pages 1448–1450, 2007.

[12] L. Bravo, W. Fan, and S. Ma. Extending dependencies with conditions. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, pages 243–254, 2007.

[13] X. Chu, I. Ilyas, P. Papotti, and Y. Ye. RuleMiner: Data quality rules discovery. In *Proceedings of the International Conference on Data Engineering (ICDE)*, pages 1222–1225, 2014.

[14] X. Chu, I. F. Ilyas, and P. Papotti. Discovering denial constraints. *Proceedings of the VLDB Endowment (PVLDB)*, 6(13):1498–1509, 2013.

[15] G. Cormode, M. Garofalakis, P. J. Haas, and C. Jermaine. Synopses for massive data: Samples, histograms, wavelets, sketches. *Foundations and Trends in Databases*, 4(1âĂŞ3):1–294, 2011.

[16] T. Dasu, T. Johnson, and A. Marathe. Database exploration using database dynamics. *IEEE Data Engineering Bulletin*, 29(2):43–59, 2006.

[17] T. Dasu, T. Johnson, S. Muthukrishnan, and V. Shkapenyuk. Mining database structure; or, how to build a data quality browser. In *Proceedings of the International Conference on Management of Data (SIGMOD)*, pages 240–251, 2002.

[18] D. Deng, R. Castro Fernandez, Z. Abedjan, S. Wang, M. Stonebraker, A. Elmagarmid, I. Ilyas, S. Madden, M. Ouzzani, and N. Tang. The data civilizer system. In *Proceedings of the Conference on Innovative Data Systems Research (CIDR)*, 2017.

[19] L. Golab, H. Karloff, F. Korn, A. Saha, and D. Srivastava. Sequential dependencies. *Proceedings of the VLDB Endowment (PVLDB)*, 2(1):574–585, 2009.

[20] L. Golab, H. Karloff, F. Korn, and D. Srivastava. Data Auditor: Exploring data quality and semantics using pattern tableaux. *Proceedings of the VLDB Endowment (PVLDB)*, 3(1-2):1641–1644, 2010.

[21] A. Halevy, F. Korn, N. F. Noy, C. Olston, N. Polyzotis, S. Roy, and S. E. Whang. Goods: Organizing Google's datasets. In *Proceedings of the 2016 International Conference on Management of Data*, Proceedings of the International Conference on Management of Data (SIGMOD), pages 795–806, 2016.

[22] A. Heise, J.-A. Quiané-Ruiz, Z. Abedjan, A. Jentzsch, and F. Naumann. Scalable Discovery of Unique Column Combinations. *Proceedings of the VLDB Endowment (PVLDB)*, 7(4):301 – 312, 2013.

[23] J. M. Hellerstein, C. Ré, F. Schoppmann, D. Z. Wang, E. Fratkin, A. Gorajek, K. S. Ng, C. Welton, X. Feng, K. Li, and A. Kumar. The MADlib analytics library or MAD skills, the SQL. *Proceedings of the VLDB Endowment (PVLDB)*, 5(12):1700–1711, 2012.

[24] Y. Huhtala, J. Kärkkäinen, P. Porkka, and H. Toivonen. TANE: An efficient algorithm for discovering functional and approximate dependencies. *Computer Journal*, 42(2):100–111, 1999.

[25] I. F. Ilyas, V. Markl, P. J. Haas, P. Brown, and A. Aboulnaga. CORDS: Automatic discovery of correlations and soft functional dependencies. In *Proceedings of the International Conference on Management of Data (SIGMOD)*, pages 647–658, 2004.

[26] D. Laney. 3D data management: Controlling data volume, velocity and variety. Technical report, Gartner, 2001.

[27] P. Langer and F. Naumann. Efficient order dependency detection. *VLDB J.*, 25(2):223–241, 2016.

[28] T. Papenbrock, T. Bergmann, M. Finke, J. Zwiener, and F. Naumann. Data profiling with Metanome. In *Proceedings of the VLDB Endowment (PVLDB)*, volume 8, 2015.

[29] T. Papenbrock, S. Kruse, J.-A. Quiané-Ruiz, and F. Naumann. Divide & conquer-based inclusion dependency discovery. *Proceedings of the VLDB Endowment (PVLDB)*, 8(7), 2015.

[30] T. Papenbrock and F. Naumann. A hybrid approach to functional dependency discovery. In *Proceedings of the 2016 International Conference on Management of Data*, Proceedings of the International Conference on Management of Data (SIGMOD), pages 821–833, 2016.

[31] D. Pyle. *Data Preparation for Data Mining*. Morgan Kaufmann, 1999.

[32] V. Raman and J. M. Hellerstein. Potters Wheel: An interactive data cleaning system. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, pages 381–390, 2001.

[33] S. Song and L. Chen. Differential dependencies: Reasoning and discovery. *ACM Transactions on Database Systems (TODS)*, 36(3):16:1–16:41, 2011.

[34] J. Szlichta, P. Godfrey, L. Golab, M. Kargar, and D. Srivastava. Effective and complete discovery of order dependencies via set-based axiomatization. *CoRR*, abs/1608.06169, 2016.

[35] J. Szlichta, P. Godfrey, and J. Gryz. Fundamentals of order dependencies. *Proceedings of the VLDB Endowment (PVLDB)*, 5(11):1220–1231, 2012.