Density- and Correlation-based Table Extension

Benedikt Kleppmann[1], Christian Bizer[1], Edwin Yaqub[2], Fabian Temme[2], Philipp Schlunder[2], David Arnu[2], Ralf Klinkenberg[2]

[1] University of Mannheim, 68131 Mannheim, Germany
benedikt,chris@informatik.uni-mannheim.de
[2] RapidMiner GmbH, Westfalendamm 87, 44141 Dortmund, Germany
research@rapidminer.com

Abstract. With thousands of data sources available on the Web as well as within organizations, data scientists increasingly spend more time searching for data than analyzing it. In order to ease the task of finding relevant data for data mining projects, this paper presents two data discovery and data integration methods that have been developed in a joint research project by RapidMiner Research and the University of Mannheim. Given a corpus of relational tables, the methods extend a query table with additional attributes and automatically fill these new attributes with data values from the corpus. The first method, *density-based table extension*, extends the query table with all attributes that can be filled with data values so that a user-specified density threshold is reached. The second method, *correlation-based table extension*, extends the query table with all attributes that correlate with a specific attribute of the query table. Both methods are integrated as operators into RapidMiner Studio, a popular data mining environment. This enables data scientists to search for data and apply a wide range of different mining methods to the discovered data within the same environment.

Keywords: Data discovery, table extension, holistic matching, web tables

1 Introduction

Table extension is the task of extending a query table with additional attributes and to populate these attributes with data values from a large corpus of relational data tables. Table extension involves table search, schema- and entity matching, and data fusion. Existing table extension systems, such as Octopus [1] or Infogather [4], assume that the user knows which attributes she wants to have added to her query table. This means that the user needs to have a theory which attributes could be relevant for her task. In contrast, in many data mining settings the relevancy of attributes is unknown in advance and is determined during the project by applying automated feature selection methods. This means that data scientists do not need to know in advance which attributes are relevant. Instead, it would be beneficial for them to be able to extend datasets with as many attributes as possible and afterwards have a feature selection method decide which attributes are relevant for the task at hand.

This paper proposes and evaluates two new table extension methods, which try to fulfill these requirements: *Density-based table extension*, which extends a query table with all attributes that can be filled above a user-specified density threshold given a data corpus. For instance, given a table describing cities, the method would add various attributes providing statistics about these cities. The second method performs *correlation-based table extension*: Given a query table describing cities, the method would add all attributes that correlate with a specific attribute of the query table. For instance, the user could specify that she wants the new attributes to correlate with the attribute *unemployment*, which would result in attributes to be added that are connected to the unemployment in the cities. Figure 1 illustrates how a query table describing Roman emperors is extended within RapidMiner Studio with additional attributes covering the emperors birth- and death dates, as well as the cause of their death. The table extension operators are published as part of the *Data Search for Data Mining (DS4DM)* extension on the RapidMiner Marketplace[3]. Beside the actual search operators, the extension also includes functionality for indexing relational tables and for managing table repositories. The extension supports extracting tabular data from various sources including web pages, google tables, tables within pdf documents, online spreadsheets from Microsoft and Google, as well accessing Sharepoint. Detailed information about the extensions is found on the DS4DM website[4].

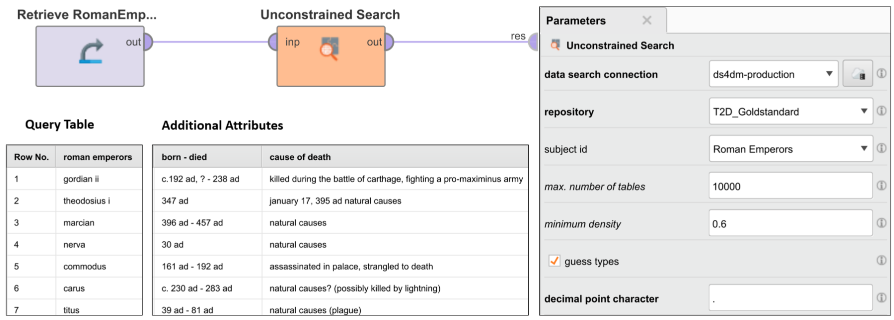

Fig. 1. Extending a table with additional attributes in RapidMiner Studio

2 Density-based Table Extension

In the following, we give an overview of the different processing steps of the density-based table extension method. A detailed description of each step is found on the website of the DS4DM backend components[5]. The method expects

[3] `https://marketplace.rapidminer.com`
[4] `http://ds4dm.de`
[5] `http://web.informatik.uni-mannheim.de/ds4dm/`

a query table, a density threshold, and a reference to a data repository as input from the user. It returns the query table extended with all attributes that could be filled above the density threshold using the data repository. The method performs the following steps in order to create the extended table:

1. *Subject Column Detection:* The method determines the column of the query table that most likely contains the names of the described entities. For this, different regex-patterns are matched against the column headers (such as .*name). If no column header is identified as a subject column header, then the string column with the highest amount of distinct values is chosen as the subject column.
2. *Table Search:* Using a Lucene index, the top-k tables having the highest overlap in subject column values with the query table are retrieved from the repository.
3. *Entity Matching:* The rows of the retrieved tables are matched against the rows of the query table in order to determine entity correspondences. The similarity of two rows is calculated by combining the similarity of the subject-column values (weight 50%) and the maximal similarity of non-subject-column values (weight 50%). The individual similarities are calculated using datatype-specific similarity metrics (string, number, and date).
4. *Schema Matching:* Correspondences between the columns of the query table and the retrieved tables are determined using a combination of label-based and instance-based schema matching techniques.
5. *Data Fusion:* Using the correspondences, the data from the retrieved tables is grouped by entity and attribute. If the retrieved tables contain conflicting values for an attribute of a specific entity, these conflicts are resolved by choosing the value that is most similar to all other values within the group.
6. *Table Extension:* All newly created attributes are added to the query table.

3 Correlation-based Table Extension

In many data analysis settings, the attributes that correlate with a specific target attribute are highly relevant, for instance for learning classification and regression models. The correlation-based table extension method expects a query table, an attribute of this query table to which the new attributes should correlate, a minimum correlation threshold, a density threshold, and a reference to a data repository as input from the user. It returns the query table extended with all attributes that could be filled above the density threshold and correlate with the specified correlation attribute. Only correlations between numeric attributes are considered. Correlations are calculated using the Pearson correlation coefficient. The correlation-based table extension method is implemented as a post-processing step for the density-based table extension. First, the density-based table extension method is used to add as many attributes as possible to the query table. Afterwards, attributes with a correlation below the minimum-correlation threshold are removed.

4 Evaluation

We evaluated both methods on the task of extending various query tables with data from a corpus of relational web tables [2] [3]. We used the T2D Gold Standard V2 for the evaluation. This table corpus consists of 779 tables and covers topics such as populated places, organizations, people, music, etc. The gold standard was originally created for evaluating web table to knowledge base matching systems. For our evaluation, we rearranged the tables into query tables and expected result tables using the schema- and instance-correspondences form the gold standard. We used 13 query tables (airports, currencies, lakes, etc.) to evaluate the density-based table extension method. Comparing the tables that were produced by the method to the expected result tables leads to a precision of 80% and a recall of 98%. This means that the method was able to discover and populate most attributes that could be added to the query tables. The precision of 80% results from errors in the data fusion step, but on the other hand also from the system filling too many cells of the result tables due to matching errors. For evaluating the correlation-based table extension, we used the four query tables that result in the largest number of numeric attributes to be added. The experiment showed a precision of 63% and a recall 77%. These results are due to the rather low density of many of the created attributes, which makes calculating correlations tricky. The results of each individual query as well as the evaluation data can be found on the website about the DS4DM backend. The run times for both of types of table extension are between 5 and 10 seconds when searching a repository of 500 000 web tables[6].

Acknowledgment

The methods and prototypes presented in this paper were developed within the research project DS4DM (Data Search for Data Mining) funded by the German Federal Ministry of Education and Research (BMBF) under grant number 01IS15027A-B.

References

1. M. J. Cafarella, A. Halevy, and N. Khoussainova. Data Integration for the Relational Web. *Proc. of the VLDB Endow.*, 2:1090–1101, 2009.
2. M. J. Cafarella, A. Halevy, D. Z. Wang, E. Wu, and Y. Zhang. WebTables: Exploring the Power of Tables on the Web. *Proc. of the VLDB Endow.*, 1:538–549, 2008.
3. O. Lehmberg, D. Ritze, R. Meusel, and C. Bizer. A large public corpus of web tables containing time and context metadata. *In Proceedings of the 25th International Conference Companion on World Wide Web*, pages 75–76, 2016.
4. M. Yakout, K. Ganjam, K. Chakrabarti, and S. Chaudhuri. InfoGather: Entity Augmentation and Attribute Discovery by Holistic Matching with Web Tables. In *Proc. of the 2012 ACM SIGMOD Int. Conference on Management of Data*, pages 97–108, 2012.

[6] `http://web.informatik.uni-mannheim.de/ds4dm/#evaluation`