Ahmad Ahmadov, Maik Thiele, Julian Eberius, Wolfgang Lehner, Robert Wrembel

Towards a Hybrid Imputation Approach Using Web Tables

Erstveröffentlichung in / First published in:

2015 IEEE/ACM 2nd International Symposium on Big Data Computing. Limassol, 07.-10.12.2015. IEEE, S. 21-30. ISBN 978-0-7695-5696-3.

DOI: http://dx.doi.org/10.1109/BDC.2015.38

Final edited form was published in "2015 IEEE/ACM 2nd International Symposium on Big Data Computing. Limassol 2015", S. 21-30, ISBN 978-0-7695-5696-3
http://dx.doi.org/10.1109/BDC.2015.38

Towards a Hybrid Imputation Approach Using Web Tables

Ahmad Ahmadov, Maik Thiele, Julian Eberius and Wolfgang Lehner
School of Electrical and
Computer Engineering
Dresden University of Technology
Faculty of Computer Science, Database Technology Group
01062 Dresden, Germany
Email: firstname.lastname@tu-dresden.de

Robert Wrembel
Poznan University of Technology
Institute of Computing Science
Poznan, Poland
Email: Robert.Wrembel@cs.put.poznan.pl

***Abstract*—Data completeness is one of the most important data quality dimensions and an essential premise in data analytics. With new emerging Big Data trends such as the data lake concept, which provides a low cost data preparation repository instead of moving curated data into a data warehouse, the problem of data completeness is additionally reinforced. While traditionally the process of filling in missing values is addressed by the data imputation community using statistical techniques, we complement these approaches by using external data sources from the data lake or even the Web to lookup missing values. In this paper we propose a novel hybrid data imputation strategy that, takes into account the characteristics of an incomplete dataset and based on that chooses the best imputation approach, i.e. either a statistical approach such as regression analysis or a Web-based lookup or a combination of both. We formalize and implement both imputation approaches, including a Web table retrieval and matching system and evaluate them extensively using a corpus with 125M Web tables. We show that applying statistical techniques in conjunction with external data sources will lead to a imputation system which is robust, accurate, and has high coverage at the same time.**

***Keywords*-Web mining; Data preprocessing; Machine learning;**

I. Introduction

The importance of data quality for information-decision systems was already assessed by Ballou et al. in 1985 [1] who identified four data quality dimensions: accuracy, completeness, consistency, and timeliness. Thirty years later, the same problems remain and with the rise of Big Data and its new capabilities to easily generate data in a large-scale manner, the issue of data quality is more important than ever before. Beside the continued growth of data volume and its increasing heterogeneity there are also changes from the data consumption side, where we can see a development towards agile data analysis overcoming inflexible and slow data warehouse infrastructures. Instead of that new information management principles such as MAD [4] or data lakes[1] arise, that allow to easily ingest, transform, and analyze data in a flexible and agile manner. Both principles assume that data is stored in its original format, eliminating all upfront integration costs but at the same time also introducing a lot of data quality problems. However, while on the one side the data lake concept introduces error-prone data, on the other side the sheer data volume provides the unique opportunity to ”repair” this data by inferring correlations between different data sources or just looking up missing data in related data source. We therefore propose a data imputation approach that given a table with missing data, is able to find these data in a Big Data source, which in this case is a large-scale corpus of arbitrary tables. Here we aim the already introduced data lake but also tables from the Web, that due to their extensive volume and wide range of information they cover, are an extreme valuable source of information. The usability of Web tables has already been shown in various other areas, e.g. factual search [17], entity augmentation [6], [7], [16], [18], ontology enrichment and so on.

As it cannot be assumed that every missing piece of data can be found in third-party data sources, whether they are stored in the data lake or in the Web, we additionally investigate machine learning (ML) techniques to impute

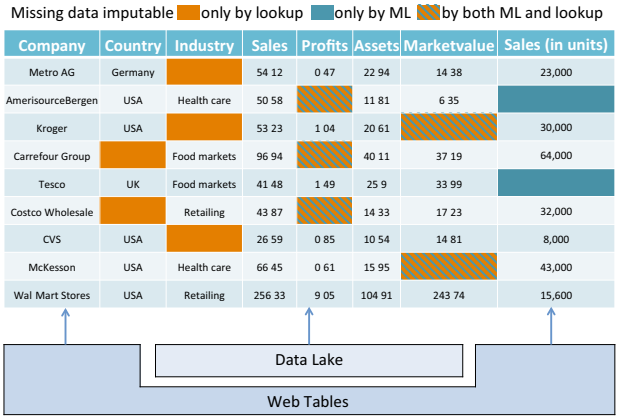


Figure 1: Example Imputation Scenario

[1] http://www.forbes.com/sites/ciocentral/2011/07/21/big-data-requires-a-big-new-architecture/

Final edited form was published in "2015 IEEE/ACM 2nd International Symposium on Big Data Computing. Limassol 2015", S. 21-30, ISBN 978-0-7695-5696-3
http://dx.doi.org/10.1109/BDC.2015.38

missing data. On the other hand, choosing the most suitable ML method for a given dataset can be a difficult task as it depends on various factors such as the type of missing attribute and its sparseness, number of available entities and explanatory variables etc. Thus, there is a need to automate ML imputation step in order to make it feasible for analysts without sophisticated background on statistics and machine learning to efficiently apply desired data analysis. In a nutshell, we propose a hybrid data imputation approach that, depending on the characteristics of a dataset containing missing values, is able to: 1) lookup these values from external data sources, 2) learn these values by applying machine learning techniques, or 3) combine both approaches in order to find the most appropriate value.

Motivating example We will discuss and motivate our hybrid imputation approach using the example in Figure 1. It shows a relational table with retailer data to which a fictional company sells their products. Here, the table contains a few missing values in four of its attributes: country, industry, profits, and sales. Intuitively, applying ML to impute categorical data such as *country* and *industry* is either not feasible within high precision threshold or close to impossible. However, it is very likely that a table in the Web contains information about the country that a retailer originates from or the industry it belongs to. On the other hand, profits is imputable by both applying ML or a lookup. There is a high correlation between the profits and sales, assets and market value of a retailer company, which can be successfully utilized to come up with an ML model to impute the missing values. At the same time, there is the chance that we would find a Web table about retailer companies containing profits related data. Finally, column "sales (in units)", which keeps track of how many units the company sold through the corresponding retailer, can only be imputed using ML, assuming that there is no other data source in the data lake containing this information.

In detail, our contributions in this paper are: (1) We present a novel Web-based imputation that is able to substitute missing values leveraging a large corpus of Web tables. (2) We propose an Automatic Imputation Model Selection approach for model-based imputation using an ensemble approach. (Currently in implementation phase) (3) We combine the best of both worlds in a hybrid imputation approach and present first experiments on a corpus of real Web tables showing its effectiveness.

The remainder of this paper is organized as follows: Section 2 briefly discusses the different types of missingness. In Sections 3 and 4 we give a detailed description of the Web-based and model-based imputation approach. The interplay of both approaches is described in Section 5. An experimental evaluation of both imputation approaches is presented in Section 6 followed by related work and a conclusion in Sections 7 and 8, respectively.

II. Data Imputation

It is very important to understand the reason why data may be missing in order to decide whether a model is able to come up with an imputation or not. In general there are three major mechanisms behind why data is missing which are shortly discussed below:

MCAR: A variable is *Missing Completely At Random* when the probability of missingness is not dependent on either the variable itself or the other variables in the dataset.

MAR: The *Missing At Random* assumption is satisfied when the probability that a variable is missing only depends on the other variables in the dataset. For example, if salary data is missing in a survey dataset, the probability of it missing depends on the other variables such as sex and education.

MNAR If missingness depends only on unobserved data itself, it is said to be *Missing Not At Random*. For example, in a survey, people do not report their salary because their salary is too high or too low.

Out of three major missingness models, MNAR is the most difficult to handle which is why most data imputation algorithms assume that the data to be imputed is *Missing At Random*. Since our novel Web-based data imputation approach does not rely only on the given dataset characteristics containing missing values, we are able to handle all different types of missingness, even MNAR data where values are missing systematically.

III. Web Table-based Data Imputation

The Web comprises large number of unstructured documents together with considerable structured and semi structured content. Furthermore, recently the Open Data trend has encouraged governments and public agencies to publish their data on the platforms such as data.gov or data.un.org. Although knowledge bases are of higher quality, their coverage is very low compared to the huge volume of Web tables that offer a lot more long tail information and does not require missing attributes to be defined in some central repository. In addition, these Web tables contain high quality relational information, which can be harnessed to cope with problem of missing values. To provide a list of candidate data sources for our Web-based imputation approach we use our *Dresden Web Table Corpus*[2] (DWTC). This corpus was extracted from the Common Crawl[3], a publicly available Web crawl and consists of 125M Web tables, which makes it a very rich and attractive source to be utilized for data enrichment and imputation purposes.

In the following we describe the overall Web table-based data imputation process (see Figure 2) consisting of two main phases. The offline phase is responsible for the extracting and indexing Web tables, whereas the online

[2]http://wwwdb.inf.tu-dresden.de/misc/dwtc
[3]http://commoncrawl.org

Final edited form was published in "2015 IEEE/ACM 2nd International Symposium on Big Data Computing. Limassol 2015", S. 21-30, ISBN 978-0-7695-5696-3
http://dx.doi.org/10.1109/BDC.2015.38

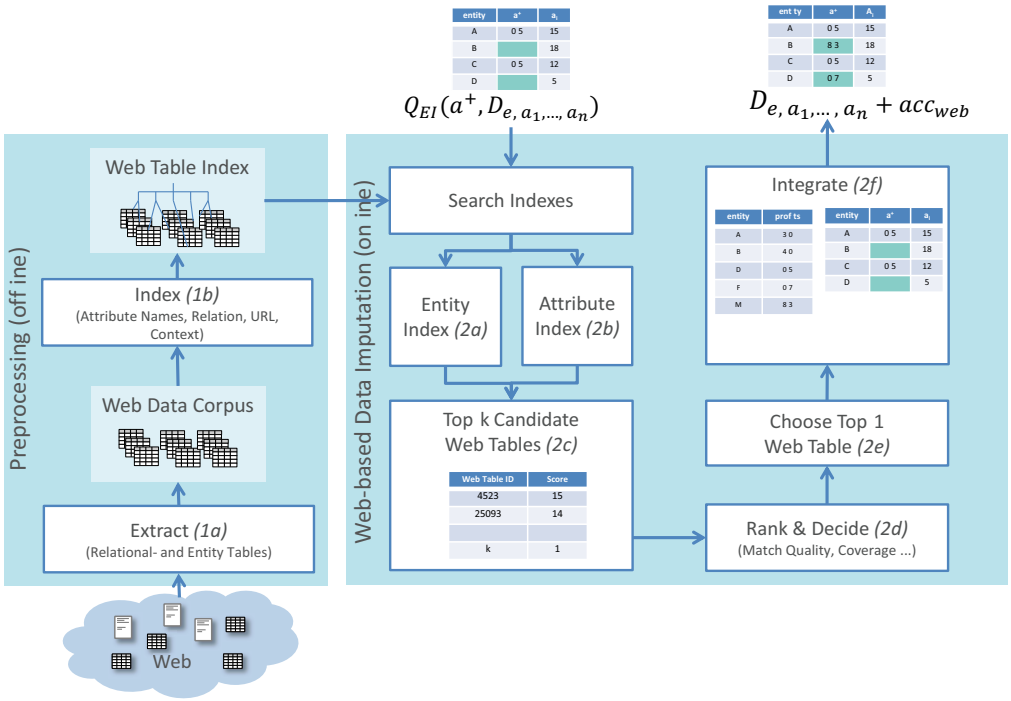


Figure 2: Web-based Data Imputation Process

phase takes a dataset as an input and looks up the missing values in the indexed Web Tables.

A. *Extracting and Indexing Web Tables*

Most of the table-structured data that can be found on the Web but also within an enterprise intranet or data lake is optimized for human consumption, such as spreadsheets, HTML, and PDF. To reuse the wealth of this information contained in these datasets for data augmentation or imputation tasks we first need to distinguish between tables that contain useful data and tables that are used for layout purposes or that contain garbage data. In [5] the authors introduced different table classes which we reduced and restructured to the following: *layout*, *relational*, *entity*, and *matrix* tables as well as type *others*. In order to categorize a given table into one of the five classes we investigated and evaluated a wide range of features and trained different classifiers on them.

Regarding the feature we distinguish between local and global ones: Global features describe the table structure itself, e.g. the maximum, average, and standard deviation of columns and row numbers, the maximum string length within all tables cells, the cumulative length consistency, the ratio of different content types such as anchors, digits, or images, and many more. Local features are determined per row and per column respectively.

To reduce the number of local features we focus on the first two rows and columns of a table as well as their last row and column. For this rows and columns we determine the average cell length, its variance and several local ratios regarding the cell content (headers, anchors, images, fonts, linebreaks, colons, etc.). In total we identified 127 different features which could be reduced to 27 by applying *correlation feature subset selection* [10] without any loss in accuracy.

Using these features we trained different classifiers such as RandomForest, J48, SimpleCart and SMO from which the RandomForest classificator performed best. For more details regarding the extraction and classification step we refer the reader to implementation of the complete extractor[4] that was used to create our Dresden Web Table Corpus. Note again, that the same techniques could be applied to relational-like data in a data lake, but since we cannot provide any enterprise data we focus on Web tables instead.

From the five table classes mentioned above we focus on relational tables and entity tables, that contain one entity only. Examples for the latter type are Wikipedia infoboxes. This allows us to retrieve large and consistent results for domains where relational tables are available and a lot of values need to be imputed, but increases recall in rare domains where only one or a few values are missing. To leverage these Web tables in the entity imputation phase we

[4]https://github.com/JulianEberius/dwtc-extractor

Final edited form was published in "2015 IEEE/ACM 2nd International Symposium on Big Data Computing. Limassol 2015", S. 21-30, ISBN 978-0-7695-5696-3
http://dx.doi.org/10.1109/BDC.2015.38

build a series of indexes on them. In detail, we index attribute names, content cells and metadata, e.g., page title, and most frequent terms in the surrounding context. For attributes, we optimistically assume that, after basic cleaning, attribute names are located in the first row for relational tables, or the first column for entity tables.

Spreadsheets, which form another class of data, that are especially common in enterprise data lakes, are most often not provided in a pure relational form. Instead their relational payload is intermingled with formatting, layout, and textual metadata. To use relational data for our data imputation approach, a transformation into first normal form relations is necessary. Here we refer the reader to [8], a previous work of us, which provides exactly this functionality.

B. Imputing Missing Values Using Web Tables

We build our Web table-based data imputation system on top of the inverted indexes described in Section III-A. Therefore, we assume a generic system that exposes an interface for keyword-based document search. This is provided for example by the well-known Lucene[5] search engine that we used for our implementation.

Given a dataset $D_{e,a_1,...,a_n}$ containing missing values in at least one of attribute $a_+ \in a_1, ..., a_n$ (e.g. country, industry, profits, or sales in Figure 1) and an entity column e, i.e. containing values identifying the individual tuples (e.g. the company column in Figure 1) we construct an entity imputation query $Q_{EI}(a_+, D_{e,a_1,...,a_n})$.

This query is divided into two keyword subqueries. The first subquery is run against the inverted index storing the entities of the Web tables (Step 2a in Figure 3), while the second subquery is send to the index containing the attributes (Step 2b). Both return a set of relational candidate tables which are then intersected with each other. The result, shown in Step 2c, is a ranked list of Top-k Web tables containing at least one of the queried entities as well as attribute a_+. Note that k is a system parameter used to lower the effort for the further post-processing of candidates.s

Subsequently the Top-k Web tables are re-evaluated (Step 2d) by computing their provided coverage, their attribute similarity and metadata match as well as their overall quality:

Entity Coverage Obviously Web table candidates covering more entities with missing values are scored higher. The coverage of a retrieved Web table candidate is determined by first creating a similarity matrix between the entities of the local table and those of the Web table. For each row of the similarity matrix, the corresponding column value with maximum similarity is returned given that this similarity is greater than a pre-defined threshold. By applying this technique we get 1) the coverage of the Web table and 2) the mapping of entities from the local table to the Web table. Each corresponding value of the similarity matrix is calculated by taking weighted sum average of several string distance measures.

Attribute Similarity Attribute similarity is calculated in a similar way. First, a similarity matrix is formed whose rows and columns are the attributes of the local table and the Web table respectively. Web tables with similar attributes as those of the local dataset are scored higher. Obviously, the most important attributes are the key attributes and the attributes with missing values.

Metadata Match Lastly, in order to score the tables which also match in content, we compare metadata of the tables. Attributes with numerical values are represented in various different formats. For example, the same GDP of a country could be represented in one table in millions while in others in billions or by a rank instead of an absolute number. We therefore use an order of magnitude approach to compare the mean of the corresponding attributes with numerical attributes in both the given dataset and a Web table to deduce whether the numbers are similar to each other. There are differences in units also: continuing with our GDP example, there are Web sources which use Euro for this purpose, while others use US Dollar. Furthermore, there are differences in metric and non-metric units. These issues have been extensively studied in the literature [18], [7], [15]. We also take into consideration the set distances between top terms in the query and tables' metadata, such as top terms extracted from page content, from the title and the URL to determine the quality of a matched Web table.

Web Table Quality A perfect match to an untrustworthy Web table may be less desirable than an almost perfect one to an established source. There are many techniques to measure trust in a Web page, such as the well-known PageRank algorithm, but these are out of scope for this paper. We approximate source quality using the popularity scores returned by the Alexa Web information service available through Amazon Web Services[6]. Coming back to the motivation stated in the introduction, namely the use of data source in an enterprise data lake, we could extend the approach into a two-stage procedure where data lake tables are scored higher compared to external data sources.

To complete Step 2d we use the weighted sum of the above mentioned quality metrics which leads us to the best matching Web tables to be used for imputation. For simplicity reasons, in this paper, we only consider a single Web table to be used for the final step of the imputation process (Step 2e). Other methods such as [7] exist for merging multiple Web tables into one result to increase coverage. In the follow up of this work we are also planning to implement more sophisticated candidate ranking and generation techniques.

Finally, the best ranked table is used to impute the missing

[5]https://lucene.apache.org/core/

[6]http://aws.amazon.com/awis/

Final edited form was published in "2015 IEEE/ACM 2nd International Symposium on Big Data Computing. Limassol 2015", S. 21-30, ISBN 978-0-7695-5696-3
http://dx.doi.org/10.1109/BDC.2015.38

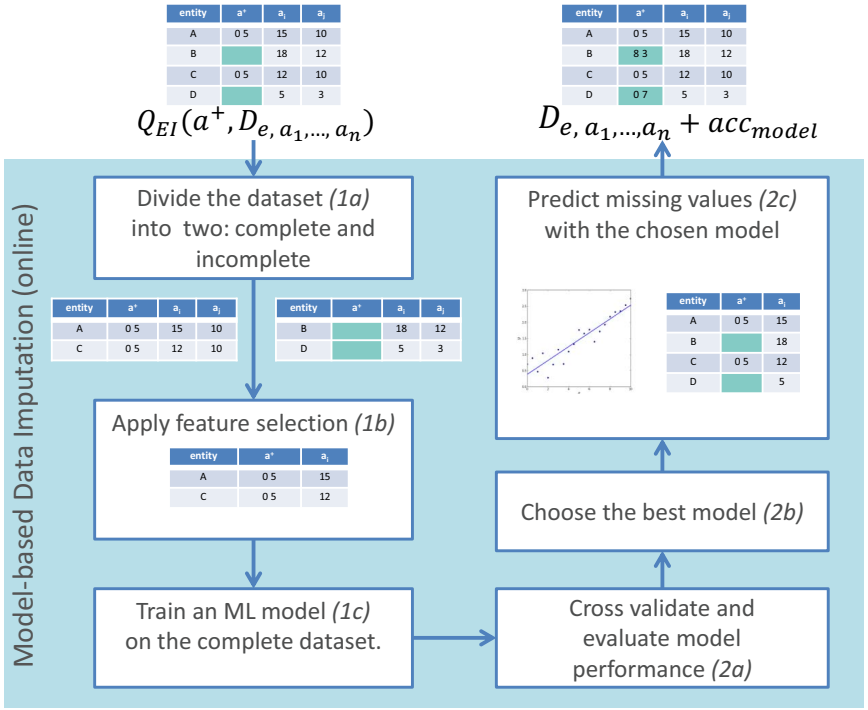


Figure 3: Model-based data imputation process

values. To match the tuples of the given dataset to the tuples of the Top-1 Web we use again the entity similarity matrix and the attribute similarity matrix. The result of the overall process is the original dataset supplemented by the value found in the Top-1 candidate.

IV. Model-based Data Imputation

Traditionally the problem of replacing missing data with substituted values is solved using various machine learning techniques. Here, a significant amount of research has been done to test the performance of machine learning techniques for data imputation [11], [12]. We would like to take this a step further to a black-box approach that automatically decides on the best technique and parameters for a given dataset with an attribute a_+ containing empty cells.

We therefore propose an imputation ensemble approach where different machine learning techniques are applied in parallel in order to determine the missing values. In order to reduce the search space and to decrease the amount of time taken to find the best matching technique, we aim to develop a set of heuristics to filter out potentially unsuitable ones. In general it can be stated that machine learning approaches for data imputation perform well if the missing values are not unique in nature and the dataset is relatively dense. Datasets satisfying those properties can be further categorized to decide on the best ensemble predictive model to be used for imputation. For example, when the attribute with missing values is categorical, building a kNN model would be more suitable than building a linear regression model. Actually, a kNN model can be used to predict both categorical (the most frequent value among the k nearest neighbors) and continuous (the mean of the k nearest neighbors) missing values. On the other hand, continuous missing values can be best imputed with the application of a linear regression model.

Subsequently, for each model we need to optimize their various parameters. To further reduce the search space, automatic feature selection algorithms will be applied to choose the most promising features. Depending on the output metrics specific to each model (such as R^2 for linear regression and accuracy for kNN), the ensemble framework will choose the most suitable imputation technique and its corresponding parameters.

The overall process is depicted in Figure 3. At the beginning we divide the dataset D into two parts: 1) the complete dataset D_{comp} with no missing values in a_+ and 2) the incomplete dataset D_{incomp} where the values for attribute a_+ are missing (Step 1a). Then we proceed to apply automatic model selection for data imputation. We start first by applying feature selection algorithms to select the best possible subset of attributes to increase the models performance, which is shown in Step 1b. We call this subset IND_{set}. Next, in Step 1c, using D_{comp} we train a model where an

Final edited form was published in "2015 IEEE/ACM 2nd International Symposium on Big Data Computing. Limassol 2015", S. 21-30, ISBN 978-0-7695-5696-3
http://dx.doi.org/10.1109/BDC.2015.38

attribute with missing values is the dependent variable and the attributes in IND_{set} are independent variables. In Steps 2a and 2b we evaluate the applied models. We iterate Steps 1c, 2a, and 2b until we obtain a model with satisfactory output. In the final step, we proceed to fill in the missing values to D_{incomp} using the chosen model. A similar approach has been taken by the authors of BayesDB[7] and Google Predictions APIhttps://cloud.google.com/prediction/docs. In BayesDB, authors make use of CrossCat, a new nonparametric Bayesian method for analyzing high-dimensional data tables to automatically infer missing values among other data analysis tasks. However, we haven?t obtained satisfactory results with the datasets we used for the experiments. Currently we manually test a variety of ML techniques with different models to come up with the best fitting model to predict missing values. The performance measures taken into account depend on the specific ML technique. For the linear regression model, we use the R^2 metric for the optimization process.

Although in many cases model-based data imputation outperforms naïve statistical approaches, such as mean value substitution, there are several drawback to it. First of all, in order for the model to work, there needs to be high correlation among the independent variables be it linear or otherwise. If no such correlation exists, generated models will not perform well and introduce further bias to the dataset. Secondly, there is a problem of overfitting. That is, the selected model might perform well on training and test datasets, however it might not perform as well in imputing missing values, i.e. by introducing imprecise and incorrect data. This is especially true if the missingness characteristics is MNAR (missing not at random, see Section II). Therefore, we combine the model-based approach with the Web table-based data imputation discussed in the previous section.

V. Towards a Hybrid Imputation Approach

After introducing both, the Web-based and model-based data imputation approaches we now want to describe how to combine both solutions in order to maximize the quality of the imputed values. In principle we have the three following options: 1) use the Web-based approach only, 2) use the model-based approach only and 3) combine both approaches. The decision between these three options is made based on the sparseness of attribute a_+, the accuracy acc_{web} and acc_{model} of both approaches (see the output of the processes illustrated in Figure 3 and 2) and the coverage of the Web-based approach (see Section III-B). If attribute a_+ does not contain any value, we have to consider the Web-based approach since no model can be trained without training data (D_{comp} is empty, see Section IV). If a_+ is very dense the probability that the model-based approach will deliver good results increases. To estimate the accuracy acc_{web} for the Web-based approach, we apply the following trick: since the Top-1 candidate will, with very high probability, not only contain entities where the values for a_+ are missing but also entities for which the values are already in the dataset, we can use these values to estimate the accuracy. In a similar way we compute the accuracy for the model-based approach by taking out some values from the training data applying 10-fold cross-validation, to test the model and derive the accuracy acc_{model}.

In the particular case of having a 100% coverage by the Web-based approach, i.e. all entities with missing values could be found in Top-1 Web table candidate, we decided between both approaches by comparing the respective accuracy values acc_{web} and acc_{model}. However, in the most cases the coverage will be lower than 100%. This is especially true when the required coverage threshold is decreased in order to find smaller Web tables, that may have a better quality and hence provide a higher accuracy acc_{web}. In the event of a coverage lower than 100% and $acc_{web} > acc_{model}$ we append to the already existing values in a_+ the values determined by the Web-based imputation approach. This will further increase the density of attribute a_+ and consequently increases the amount of training data. This in turn helps to build better performing models to impute attributes that can only be imputed by machine learning approaches.

To summarize, by applying both techniques at the same time, we are able to impute datasets with a much higher accuracy and can address all types of missingness especially the hardest one: Missing Not At Random (MNAR).

VI. Evaluation

We conducted an experimental evaluation of the proposed hybrid approach to test the following question: what is the sweet spot for the hybrid approach? Under which circumstances and which type of datasets the hybrid approach has more coverage and better accuracy?

A. Datasets

To provide a large amount of Web tables, we use an index over our Dresden Web Table Corpus (see Section III) for the Web-based imputation. To demonstrate the benefits of the proposed hybrid approach we have used two real-life datasets with attributes of various characteristics:

- Country macroeconomics dataset: We have manually compiled different macroeconomic metrics of 200 countries using various online sources such as Wikipedia and the World Bank website. The metrics include population, GDP, GDP per capita, GDP real growth rate, external debt, imports in US$, exports in US$, internet penetration rate, life expectancy, poverty and unemployment rate for the year of 2013.
- The Forbes Global 2,000 dataset released by Forbes, which is an annual dataset ranking top 2,000 companies of the World. The ranking is based on a mix of four

[7]http://probcomp.csail.mit.edu/bayesdb/

Final edited form was published in "2015 IEEE/ACM 2nd International Symposium on Big Data Computing. Limassol 2015", S. 21-30, ISBN 978-0-7695-5696-3
http://dx.doi.org/10.1109/BDC.2015.38

metrics: sales, profit, assets, and market value. The dataset also contains two categorical attributes, namely country and industry.

For the purpose of the experiments we introduced missing values to the attribute of interest given a threshold of missingness. The entity attribute e for the macroeconomics dataset is *country*, while for the Forbes it is *company*.

For the country macroeconomics dataset we choose poverty as the target attribute a_+. On the other hand, we choose market value as a_+. We compare the results of machines learning and Web table lookup imputation under various sparseness. Intuitively machine learning approaches do not perform well when the sparseness is high due to the lack of the training dataset. In general, since poverty is a highly complex metric that is dependent on many more factors, it is not possible to obtain a precise model. On the other hand, there's an abundant source of macroeconomic data which is also available in the Web table corpus.

B. Measures of Performance

We show the results of experiments under various sparseness thresholds. The original tables are kept intact to be used as the ground truth for the evaluation of accuracy and coverage. The accuracy, which is 100 minus the sum of percent errors divided by the number of tuples with missing values (Mean Absolute Percent Error), was calculated using formula 2. Note that, when predictions errors are very large, the resulting accuracy will be negative. We currently apply experiments to a_+ with continuous numerical values only. The main goal obviously is to maximize accuracy by minimizing the average error.

$$percent\ error = \frac{(imputed - original)}{original} * 100 \quad (1)$$

$$accuracy = 100 - \frac{\sum percent\_error}{\#\ of\ tuples\ with\ missing\ values} \quad (2)$$

C. Hybrid Imputation Approach

In a first experiment we evaluated just the Web-based imputation approach: Therefore, we increased the coverage threshold for the Web-based approach from 10% to 90% in steps of 10 percentage points for the countries dataset and from 8.3% to 83% in steps of 8.3 percentage for the companies dataset. As it can be seen from 5a for the company dataset, the overall accuracy increases dramatically. Also note the negative accuracy value when the dataset is very sparse. This is due to the impact of very large prediction errors on the overall result which we discuss in VI-B Although with less impact on the overall process, the same trend can be seen for the country dataset. This is due to the fact, that not every country reports about their poverty rate and the amount of data we could obtain by using the web based lookup was less than 70% for the whole dataset. Obviously we could use the obtained imputation model to predict the remaining 30% of the missing information. However, since we did not have ground truth to compare the accuracy of the obtained results, we have not included them in this experiment.

In a further evaluation, we investigated the accuracy of the overall hybrid approach: To simulate MNAR characteristics, we first intentionally removed the market values of top companies regarding the market value to introduce bias to the dataset. We then applied the pure model-based approach to impute the missing values. As a result, the R^2 value for the model, which is a statistical measure of how close the data are to the fitted regression line, is 0.31 only. The resulting imputed / predicted values and the difference to the original values are shown in Figure 4a). Next, we applied our novel hybrid approach, to substitute a small fraction of the missing values (10%) with a high accuracy using a Web-based lookup. The remaining values are than estimated by applying a model-based approach (in this case linear regression) using the already existing and the additional 10% as training data. The result, is plotted in Figure 4b. It can be seen, that by first imputing just a small portion (10%) of the missing values by Web lookup, the accuracy of the overall imputation increases greatly. In other words, otherwise poor imputation models due to the lack of enough training data or MNAR characteristics, can be improved to a great extend by applying the hybrid imputation approach.

For the country dataset, we have applied the tests in the same manner. However, the effect of using the hybrid approach was not as effective as it was for the company dataset. In order to get a better predictive power than the model-based imputation, we had to impute at 35% of the missing values from the Web. As a result, the R^2 value of the model has increased from 0.23 to 0.52. We then used again the already existing and the additional 35% as training for the model-based approach. The results are shown in 4c and 4d. It can clearly be seen, that using hybrid approach increases the accuracy of the overall imputation process.

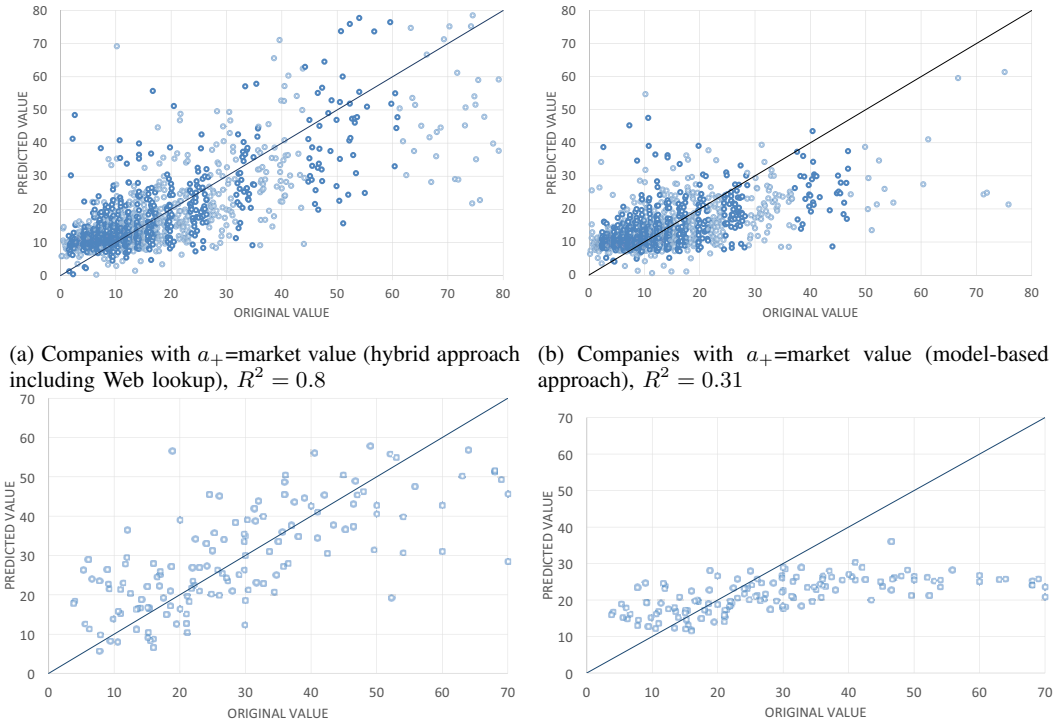
VII. Related Work

In the related work section we talk about the existing body of research on Web-based imputation, Entity Augmentation and traditional data imputation techniques used in statistics.

A. Web-based Imputation

To the best of our knowledge, there is only one research project that aims at data imputation using the Web data: the WebPut project [13] leverages traditional search methods together with the capabilities of Web search engines towards the goal of completing missing attribute values in relational tables. The main idea is to formulate an effective Web search query based on the existing and missing tuples

Final edited form was published in "2015 IEEE/ACM 2nd International Symposium on Big Data Computing. Limassol 2015", S. 21-30, ISBN 978-0-7695-5696-3
http://dx.doi.org/10.1109/BDC.2015.38



(a) Companies with a_+=market value (hybrid approach including Web lookup), $R^2 = 0.8$

(b) Companies with a_+=market value (model-based approach), $R^2 = 0.31$

(c) Countries with a_+=poverty (hybrid approach including Web lookup), $R^2 = 0.52$

(d) Countries with a_+=poverty (model-based approach), $R^2 = 0.23$

Figure 4: Original versus predicted values

and then parse the retrieved Web documents using C-DI (Co-occurrence based data imputation) and P-DI (Pattern based data imputation). The authors employ what they call as confidence-based schema in order to rank values returned from each imputation query and thus choose the one with the highest confidence as the candidate for the missing attribute. In addition, estimated values are used in the following iterations to find remaining missing values. Whereas WebPut focuses on substituting individual missing value by formulating patterns and parsing unstructured Web documents we are indexing semi-structured Web tables in an offline phase which are later searched using keyword queries. While the WebPut approach may reach a better recall since it is querying the whole Web, the precision and accuracy will be much lower since we are able to impute a series of missing value by using the same data source.

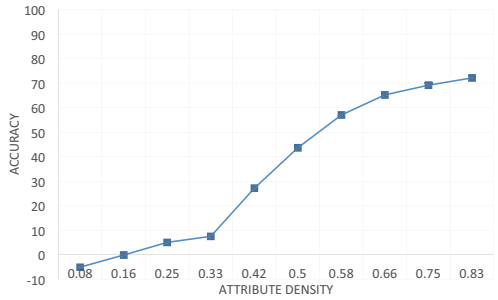
B. Entity Augmentation

Our hybrid imputation approach builds upon methods for automatic, Web table-based entity augmentation. A notable first example of such work is [3], which described a set of basic operators that facilitate the integration of many structured data sources from the Web. One of these operators, called `Extend`, attempts to find matching Web tables for a requested attribute and an existing table. InfoGather [16] improved the state of the art especially by identifying more candidate tables than the naïve matching approach, by introducing Web table similarity measures and identifying tables indirectly matching the query through them. In a follow-up paper [18], the system was extended to explicitly assign labels for time and units of measurements to tables, allowing for more targeted retrieval of specific attribute variants. [7] extends those systems by allowing top-k entity augmentation instead of single answer augmentation. This gives users a new way to deal with the ambiguity of Web data-based query results by offering alternative solutions.

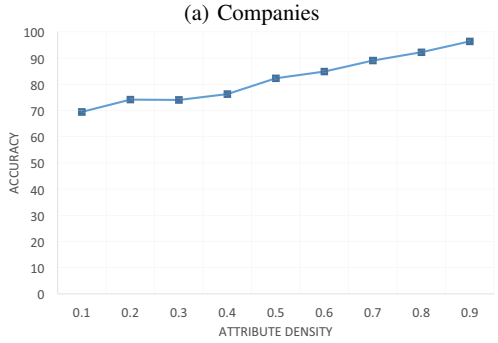
C. Data imputation in statistics

Data imputation has been widely studied in statistics. Works in this area can be divided into to categories: substitute-based and model-based data imputation. Techniques from the first category try to find a substitute value for the missing one from the same data set. In [9] the authors presented nine sub-approaches falling into this category, for example ”the most common attribute value” or a ”closest fit”. Well-known examples for the latter sub-approach are kNN and association rules that both try to find a “close-fit” value from a similar context. Model-based data imputation approaches form the second category. These approaches use predictive modelling based on the dataset to estimate values

Final edited form was published in "2015 IEEE/ACM 2nd International Symposium on Big Data Computing. Limassol 2015", S. 21-30, ISBN 978-0-7695-5696-3
http://dx.doi.org/10.1109/BDC.2015.38



(a) Companies



(b) Countries

Figure 5: Accuracy for increasing attribute density obtained by Web-lookups

for the missing ones. Here several techniques have been developed tackling different attribute types, such as [14] for continuous attributes and [2] for discrete attributes.

VIII. Conclusion

In this paper we proposed a novel imputation technique that leverages that abundant source of freely available Web tables. Therefore, we developed a Web table retrieval, matching and imputation system that is able to find missing values in millions of documents. Based on that, we proposed a novel hybrid data imputation strategy that takes into account the characteristics of an incomplete dataset and based on that chooses the best imputation approach. We showed the effectiveness of our approach in terms of accuracy by conducting several preliminary experiments based on a large Web table corpus. Currently work in progress is the automation of the model selection step. We are aware that there might be better performing models for the given datasets and we will address these issues in the automatic model selection step of the proposed approach. Furthermore, we are working on the implementation of more advanced Web table matching and scoring methods. In addition, we are planning to apply the proposed method to other datasets to further investigate the influence of different data characteristics.

Acknowledgment

This research has been funded by the European Commission through the Erasmus Mundus Joint Doctorate "Information Technologies for Business Intelligence - Doctoral College" (IT4BI-DC).

References

[1] D. P. Ballou and H. L. Pazer. Modeling Data and Process Quality in Multi-Input, Multi-Output Information Systems. *Management Science*, 31(2):150–162, 1985.

[2] J. Barnard and D. B. Rubin. Small-sample degrees of freedom with multiple imputation. *Biometrika*, 86(4):pp. 948–955, 1999.

[3] M. J. Cafarella, A. Halevy, and N. Khoussainova. Data integration for the relational web. *VLDB*, pages 1090–1101, 2009.

[4] J. Cohen, B. Dolan, M. Dunlap, J. M. Hellerstein, and C. Welton. Mad skills: New analysis practices for big data. *Proc. VLDB Endow.*, 2(2):1481–1492, Aug. 2009.

[5] E. Crestan and P. Pantel. Web-scale table census and classification. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 545–554, New York, NY, USA, 2011. ACM.

[6] J. Eberius, M. Thiele, K. Braunschweig, and W. Lehner. Drillbeyond: Enabling business analysts to explore the web of open data. *PVLDB*, 5(12):1978–1981, 2012.

[7] J. Eberius, M. Thiele, K. Braunschweig, and W. Lehner. Top-k entity augmentation using consistent set covering. In *Conference on Scientific and Statistical Database Management, SSDBM '15, San Diego, US, June 28 - July 01*, 2015.

[8] J. Eberius, C. Werner, M. Thiele, K. Braunschweig, L. Dannecker, and W. Lehner. Deexcelerator: A framework for extracting relational data from partially structured documents. In *Proceedings of the 22Nd ACM International Conference on Conference on Information & Knowledge Management*, CIKM '13, pages 2477–2480, New York, NY, USA, 2013.

[9] J. W. Grzymala-Busse and M. Hu. A comparison of several approaches to missing attribute values in data mining. In *Revised Papers from the Second International Conference on Rough Sets and Current Trends in Computing*, RSCTC '00, pages 378–385, London, UK, UK, 2001. Springer-Verlag.

[10] M. A. Hall. Correlation-based feature selection for machine learning. Technical report, 1999.

[11] J. M. Jerez, I. Molina, P. J. Garca-Laencina, E. Alba, N. Ribelles, M. Martn, and L. Franco. Missing data imputation using statistical and machine learning methods in a real breast cancer problem. *Artificial Intelligence in Medicine*, 50(2):105 – 115, 2010.

[12] K. Lakshminarayan, S. A. Harp, R. P. Goldman, and T. Samad. Imputation of missing data using machine learning techniques. In E. Simoudis, J. Han, and U. M. Fayyad, editors, *KDD*, pages 140–145. AAAI Press, 1996.

[13] Z. Li, M. A. Sharaf, L. Sitbon, S. Sadiq, M. Indulska, and X. Zhou. Webput: Efficient web-based data imputation. In *Proceedings of the 13th International Conference on Web Information Systems Engineering*, WISE'12, pages 243–256, Berlin, Heidelberg, 2012. Springer-Verlag.

Final edited form was published in "2015 IEEE/ACM 2nd International Symposium on Big Data Computing. Limassol 2015", S. 21-30, ISBN 978-0-7695-5696-3
http://dx.doi.org/10.1109/BDC.2015.38

[14] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[15] S. Sarawagi and S. Chakrabarti. Open-domain quantity queries on web tables: Annotation, response, and consensus models. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 711–720, New York, NY, USA, 2014. ACM.

[16] M. Yakout, K. Ganjam, K. Chakrabarti, and S. Chaudhuri. Infogather: entity augmentation and attribute discovery by holistic matching with web tables. In *SIGMOD*, pages 97–108, 2012.

[17] X. Yin, W. Tan, and C. Liu. Facto: a fact lookup engine based on web tables. In *WWW*, pages 507–516, 2011.

[18] M. Zhang and K. Chakrabarti. Infogather+: semantic matching and annotation of numeric and time-varying attributes in web tables. In *SIGMOD*, pages 145–156, 2013.