A Scoring-based Approach for Data Preparator Suggestion

Lan Jiang, Gerardo Vitagliano, and Felix Naumann

Hasso Plattner Institute, University of Potsdam
`firstname.lastname@hpi.de`

Abstract. Self-service data preparation enables end users to prepare data by themselves. However, the plethora of possible data preparation steps can overwhelm the user. We introduce a score-based preparator ranking approach to propose preparator candidates in a context-specific manner. To this end, we give scoring functions for a selected set of preparators and outline future work towards a full-fledged data preparation system.

1 Preparator Suggestion

Data preparation (DP) is the process of cleaning and transforming raw data before serving them to downstream applications. This process iteratively applies a series of *preparators* on data. The traditional way of obtaining prepared data requires collaboration between IT specialists and end users. The former prepare data according to some specifications, whereas the latter, who are usually domain experts, consume these prepared data in the subsequent analytic applications. If questionable outcomes occur, the preparation pipeline is revisited and adapted accordingly. This modus operandi is often far from efficient, first because it produces communication overhead: the two sides have to go back and forth until eventually understanding each other. Second, the preparation itself is not trivial: it is reported to account for up to 80% of the time spent in the whole data analysis lifecycle [2]. To fill this gap, *self-service data preparation* has been proposed: it uses data profiles, intelligent data processing algorithms, and advanced user interfaces to allow easier exploratory data preparation by end users [5].

However, domain experts may not have enough technical skills or time to prepare their data, especially when the data comes with many dimensions and complicated schemata. For example, facing a dataset with hundreds of columns, one might hesitate about which columns might be merged into one, or not understand which parameters of a preparator to choose to consistently format a column, even provided with elaborate profiling information. We believe modern data preparation systems should be equipped with a *preparator suggestion* mechanism that guides users through data preparation steps. We describe a score-based preparator suggestion framework that proposes to the user a ranked list of suggested preparators for the next step in the pipeline. We present a selection of preparators and the corresponding heuristics to allow the framework to determine the applicability of these preparators in the current data context.

2 Related Work

Only few research projects attempt data preparator suggestion. Heer et al. proposed a framework for predictive interaction, which captures data transformation intentions by requiring users to make a few clicks on the content of interest [4]. The framework constructs a few patterns that each are able to transform these selected content, and suggest a group of possible transformations accordingly. Their approach is manifested in the commercial Trifacta tool. Guo et al. proposed a score-based approach that proactively suggests a handful of preparators selected out of eight candidates that each can transform tabular data towards a fixed target format [3]. The approach employs heuristics to reduce the search space. Both approaches can be used towards preparator suggestion but need further work to address scalability to many preparators.

An obvious approach to preparator suggestion is to train a model based on previous choices on similar data. Presistant exploits a meta-learning approach that trains a learner to suggest a set of preparators that improves the performance of classifiers on particular use cases [1]. It considers nine preparator candidates and aims at improving the result of five classifiers. MiningMart borrows the idea of case-based reasoning (CBR) for this purpose [7]. It stores the best-practice data preprocessing workflows developed by experienced users. As a new case arrives, the system calculates the similarity between the new case and each of the stored ones to select the closest one and applying its data preprocessing workflow to the new case.

Other works focus on traditional data cleaning tasks, such as repairing data errors [8], or consolidating inconsistent data formats [6]. We deem these techniques as necessary implementations for our individual preparators, such as finding outliers, or unifying data formats.

3 Scoring-based Suggestion Engine

We propose a scoring-based approach that suggests a ranked list of preparators based on the data and interaction context at present. These preparators come from the search space of parameterized preparator candidates. A *parameterized preparator* is a preparator whose parameters are pre-set with values. We introduce a heuristic metric to measure the applicability of each preparator. These metrics are designed under two general guiding ideas: after applying the suggested preparator, datasets (i) become more homogeneous and (ii) have metadata that is closer to the predefined target, if it is known. Given a data context, each preparator computes an applicability score based on the corresponding heuristic. The *decision engine* collects the scores and outputs those with highest scores. In future work, we will explore to use a user-defined target data schema to assist the decision engine.

A preparator is an atomic operation to transform data. We define a preparator as a tuple of precondition and signature. The *precondition* is the required metadata that validate a preparator. The *signature* represents the preparator with all its parameters. The signature of a selection of preparators is shown in Table 1.

We design an applicability measure for each of the preparators by using *profiles* and *statistics* derived from the data context. These measurements vary according to the particular characteristics of preparators. For example, we use the percentage of empty cells in a column to measure the applicability of the preparator Fill missing value. More exemplary scoring functions can be found in Table 1. To make these measurements comparable to each other, we normalize them to [0,1], where a higher score means the preparator is more applicable. The preparators suggested by our approach are ranked according to their applicability score.

Table 1. Data preparators and the applicability score functions.

Preparator	**Signature**	**Applicability measurement**
Delete column	`delete(col)`	$S_{DC} = \frac{\#\ null\ cells}{\#\ total\ cells}$
Split column	`split(col, sep, n)`	$S_{SC} = \frac{1}{n}\sum_i^n(similarity(val\_pat(c_i)))$, where $val\_pat(c_i)$ are the value patterns of generated column c_i, n is the number of new columns
Merge columns	`merge(col1, col2)`	$S_{MC} = (\#\ \text{interleaving tuples}) \times DT$, where $DT = \begin{cases} 1, \text{same data type} \\ 0, \text{different data type} \end{cases}$
Fill missing values	`fill(col, value)`	$S_{FM} = 1 - \frac{\#\ null\ cells}{\#\ total\ cells}$
Remove preamble	`remove_preamble()`	$S_{RP} = \begin{cases} 1, \text{has preamble} \\ 0, \text{has no preamble} \end{cases}$
Change value format	`reformat(col, srcFormat, tarFormat)`	$S_{CF} = \frac{\#\ failed\ cells}{\#\ total\ cells}$
Filter outliers	`filter_outlier(col)`	$S_{FI} = \frac{\#\ outlier\ cells}{\#\ total\ cells}$

Preparators with different parameter signatures are essentially different candidates. As potentially many values can be assigned to these parameters, the *search space* of suitable preparators may be large. Calculating all these candidates may cause latency, and is therefore not acceptable in interactive self-service data preparation systems.

We introduce three pruning rules to reduce the search space. First, we check whether the metadata precondition of a preparator satisfies the current valid metadata, and filter out it if any of the metadata do not match. For example, given a piece of metadata that indicates no empty cells in a column, the Fill missing values should be filtered out and not shown to the user. Second, the preparators have been conducted may affect the applicability of a preparator candidate for the next step. For example, if we have conducted Change value format, it is not likely to perform it again. We are seeking various heuristic measurements of using this type of pipeline contexts to reduce the search space. Last but not least, we use the previously computed scores to avoid unnecessary checks. For that, we need to store the score of each candidate that was calculated in a previous step. After computing the score of a candidate, we may directly skip the cached candidates with a lower score, as long as the object data part of these candidates were not affected by the previous performed preparators.

4 Conclusion and Future Work

Self-service data preparation aims at reducing DP expense and boosting efficiency of data-driven applications. Suggesting suitable preparation steps can further reduce time consumption. In this paper, we propose a score-based ranking approach for the preparator suggestion problem. We introduce a handful of preparator candidates that could be suggested by our approach, using heuristic scoring measurements.

A future solution to this problem may be to employ machine learning techniques: When provided with enough samples of input/output datasets, and corresponding DP pipelines, we may train ML models to predict appropriate preparators for the preparation situation at hand.

References

1. B. Bilalli, A. Abelló, T. Aluja-Banet, and R. Wrembel. Automated data preprocessing via meta-learning. In *Proceeedings of the International Conference on Model and Data Engineering*, pages 194–208, 2016.
2. T. Dasu and T. Johnson. *Exploratory Data Mining and Data Cleaning.* John Wiley, 2003.
3. P. J. Guo, S. Kandel, J. M. Hellerstein, and J. Heer. Proactive wrangling: mixed-initiative end-user programming of data transformation scripts. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 65–74, 2011.
4. J. Heer, J. M. Hellerstein, and S. Kandel. Predictive interaction for data transformation. In *Proceedings of the Conference on Innovative Data Systems Research (CIDR)*, 2015.
5. J. M. Hellerstein, J. Heer, and S. Kandel. Self-service data preparation: Research to practice. *IEEE Data Engineering Bulletin*, 41(2):23–34, 2018.
6. Z. Jin, M. R. Anderson, M. Cafarella, and H. Jagadish. Foofah: Transforming data by example. In *Proceedings of the International Conference on Management of Data (SIGMOD)*, pages 683–698, 2017.
7. K. Morik and M. Scholz. The MiningMart approach to knowledge discovery in databases. In *Intelligent technologies for information analysis*, pages 47–65. Springer, 2004.
8. T. Rekatsinas, X. Chu, I. F. Ilyas, and C. Ré. HoloClean: Holistic data repairs with probabilistic inference. *PVLDB*, 10(11):1190–1201, 2017.