AlphaClean: Automatic Generation of Data Cleaning Pipelines

CS520 Literature Review Report

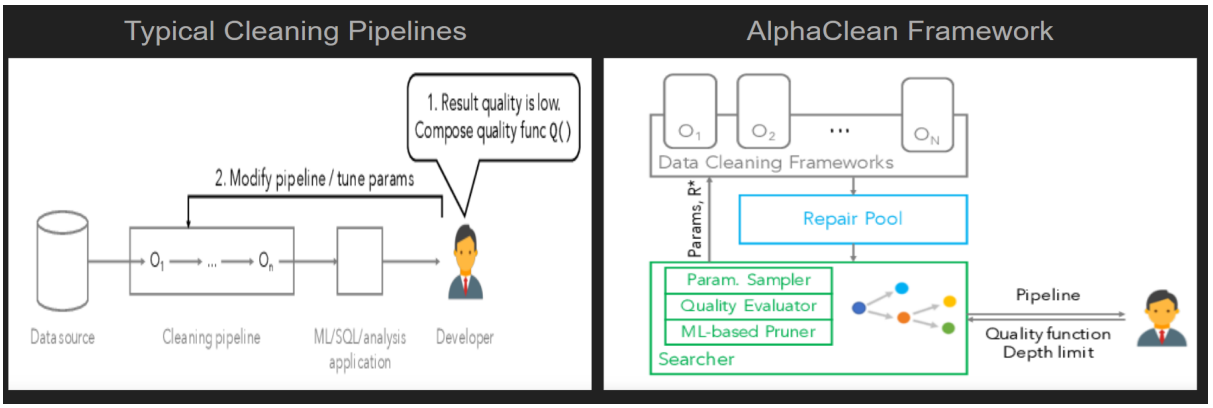
Geongu Park, Sowmya Nethagani, Khiem T. Truong

1. ABSTRACT

Nowadays, it is known that most data analysts spend up to 80% of the analysis time during data cleaning and preparation. Therefore, they desire a high-quality automatic data cleaning pipeline before they jump into the analysis phase. Not only do analysts have options to use such python libraries like pandas or existing data cleaning pipelines to use in hand-written scripts for data cleaning but also have frameworks that provide automated data cleaning processes for them. This paper introduces an automatic generation of data cleaning pipelines system called AlphaClean that finds the optimal cleaning pipeline of up to 9 times higher quality than directly applying the aforementioned methods.

2. LITERATURE REVIEW

Whenever people work on the raw dataset, they can observe a lot of incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. Therefore, the data cleaning process is necessary to avoid the poor performance of models when trained using uncleaned data. In other words, data cleaning is a process to remove unnecessary data from the dataset that could lead to a potential reduction in the performance of models. To achieve this, analysts used to design a hand-written script, but nowadays, this trend is changing to use automated data cleaning libraries that build and tune structured pipelines themselves. The AlphaClean that the paper introduces is one of them that finds solutions up to 9 times better quality than directly applying state-of-art parameter tuning methodologies, which is absolutely stronger in underperforming data cleaning methods and redundancy in the data cleaning library.



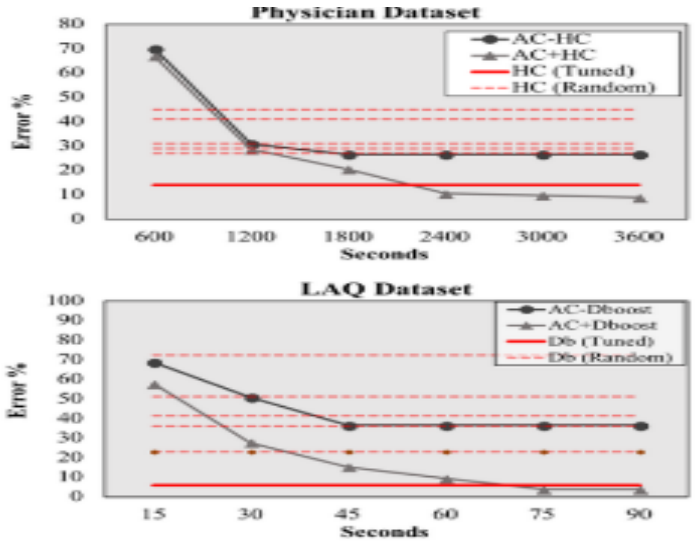
As it can be seen from the above figure about typical cleaning pipelines, human-in-the-loop is necessary to have a good quality result. Such a human-in-the-loop process includes developing an evaluation function to discover the potential quality problems and manually fixing the errors that occur from the data cleaning pipeline. Therefore, it is such a huge workload for a developer to manually address them to find the optimal solution due to the fact that the space required to process the operator pipelines is exponentially growing. However, as it is obvious in the figure of Alphaclean Framework, there is no human interaction during the finding of the optimal data cleaning pipeline process. By automating this human-in-the-loop process and looking for the optimal solution, AlphaClean provides users with what they desire and saves their time by huge. Basically, AlphaClean is an Application Programming Interface (API) that connects users and the existing data cleaning frameworks. In simple English, there are multiple candidate data cleaning frameworks, and AlphaClean automates the process to choose the best candidate to include in the optimal solution. In the AlphaClean system, the finding of the optimal pipeline is done by the Searcher operation. The Searcher can be divided into 3 sub-operations that are parameter sampler, quality evaluator, and machine learning-based pruner. Outside of Searcher, there exists storage called Repair Pool that stores the outputs of the results from the quality evaluator. The workflow of the AlphaClean is as follows. Based on the figure above, a thread is assigned to each framework, and they operate based on the parameter values provided by the parameter sampler, and its outputs are appended to the repair pool. Afterward, the quality of the outputs in the repair pool is then computed with the quality evaluator component. Finally, the ML-based pruner uses the computational result of the quality evaluator performed on the parameter sampler outputs in the repair pool to remove or extend the set of candidate cleaning pipelines to deliver the optimal pipeline to the user.

3. SEARCH ALGORITHM

Here we will be seeing about our system enhancement. Firstly, there is a Parameter sampler the users will specify two types of parameter properties such as Attribute Name Parameter and Threshold Parameter for which the AlphaClean will apply search optimizations. In the Attribute name parameter, the AlphaClean will work out the domain of permitted values which will reflect the database attribute. When there are attribute names that do not belong to the quality function then the AlphaClean will clean up the parameter space. Some examples of numeric parameters are thresholds, inference parameters, and confidence bounds. AlphaClean wipes out the areas from most to least restrictive.AlphaClean will first have the sample as rec=0 and then will lower the threshold. Secondly, there is parallelization where composing and evaluating Q(s′(R)) is the most expensive search operation, even with

incremental evaluation. So we parallelize data partitions and candidate pipelines. There are two types of parallelization that are search parallelism and data parallelism. For the Search Parallelism, the pipelines and their qualities will be sent and received by the worker who recomputes the pipeline results. Which will ensure the priority queue is distributed evenly among the workers for the next iteration. In data parallelism, many large datasets will naturally partition themselves like the timestamp or region. The ultimate goal is that we actually partition the data set so that the number of errors is limited to a small number of records. Most of the partitioning functions can also be defined by the users also. In our current implementation to partition the input relation by row, user-specified blocking rules are used. Thirdly, there is learning of the pruning rules where the goal is to develop data and quality function-dependent pruning rules. For that, AlphaClean uses data parallelism to run parallel searches for each block of the dataset, producing an optimized cleaning pipeline for each block. AlphaClean develops the ideal cleaning pipeline for each block based on a set of training instances. Positive training is defined as a conditional assignment c in a block's optimal cleaning plan s*. AlphaClean uses a Logistic Regression classifier that prefers false positives to false negatives. This is done by training the model and increasing the prediction threshold until there are no False Negatives. Featurization is a technique in which each conditional assignment is transformed into a feature vector. The options point AlphaClean to the data cleaning methods and parameter settings that have shown the most promise in earlier blocks. AlphaClean uses a linear classifier since it is easy to train. We may be able to automatically learn the features themselves if we apply a deep learning strategy to a big enough number of cleaning tasks that share a common set of data manipulations.

4. EXPERIMENTS



The major purpose of this study is to compare AlphaClean to modern black box hyper-parameter tuning techniques and to understand the instances that occur. Then, in comparison to data cleaning systems like HoloClean, show the possibilities of a search-based strategy. There is a comparison with contemporary black box hyper-parameter tuning algorithms. The datasets used here for data cleaning are Hospital, London Air Quality (LAQ), and Physician. To test Alpha Clean's performance, it was compared with modern black box hyper-parameter tuning algorithms such as HoloClean. The tests are performed on datasets (Hos, LAQ, Phys) that were used in prior data cleaning benchmarks, each with different sizes and cleaning needs. In the above figure, AlphaClean is compared against single standalone systems that address functional dependencies (Holoclean HC) and numerical errors (DBoost). The result shows that AlphaClean can support both errors type and work with a variety of frameworks. Even Though a single data cleaning method can optimize the quality on its own, HC got 86% while AC without HC got 73%, but together, using AC+HC gets the result to 91%, showing how AC help addresses the weak spots of the methods, a similar thing is shown with the result of other systems.

5. CONCLUSION

Over time, data cleaning methods have become more complicated. Data cleansing pipeline optimization differs greatly from machine learning pipeline tuning, according to this paper. Each pipeline in AlphaClean is treated as a conditional assignment operation. This is the collection of all conditional assignment pipelines in a well-posed search space. Repairs are canonicalized by AlphaClean as conditional assignment procedures. Their results are fed into a pool of conditional assignments. Advances in planning and optimization can tackle a variety of data cleansing challenges, but they are counter-intuitive.