PROJECT REPORT CS 848

HOLOCLEAN – SAMPLING ON DIMENSIONAL MODEL

Instructor: Ihab F. Ilyas

Contributors: Archit Shah, Ishank Jain, Marian Boktor

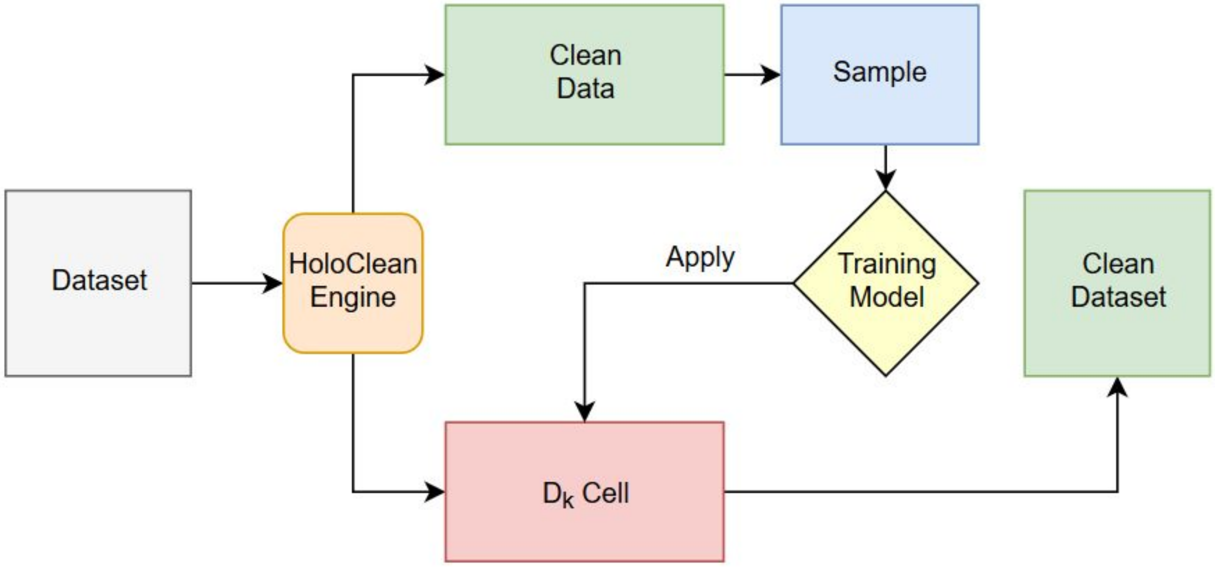

Fig.1 The flowchart of HoloClean cleaning process

1. Introduction

HoloClean is a framework for holistic data repairing driven by probabilistic inference. HoloClean unifies qualitative data repairing, which relies on integrity constraints or external data sources, with quantitative data repairing methods, which leverage statistical properties of the input data [1].

Our project builds upon the line of thought– “Allow data scientists to effectively communicate their domain knowledge in a declarative way to enable accurate analytics, predictions, and insights form noisy, incomplete, and erroneous data”. Our project leverages the “domain knowledge” of the dataset to sample training data in an effective manner from the Clean Data (refer Figure 1). In our project we intend to reduce the number of training points considered to train the neural network model which is used to clean dk cells, this helps us to bring down the memory utilized during the whole process and at the same time also reduce the runtime for the process. For our project we also looked into locality-sensitive hashing, edit distance based clustering and clustering based on column dependencies. We didn't proceed with the previous ideas because of higher cost to perform clustering.

2. Motivation

The present version of HoloClean generates multiple candidate values for each cell (training points) in the dataset as a part of domain. Domain values/candidates are alternate values which might take place of the ground truth. Once domain is generated the training points can then be used to train the model which is used to repair the dk cells (unclean cells). We are motivated to use our knowledge of schema, dimension table (holds attributes that don't vary much) to reduce the numbers of training points considered for training. Moreover, in the HoloClean paper [1], we learn that most of the time in the cleaning process is spent on generating feature tensor and training the model. We want to reduce the number of training points used in training the model using sampling technique and also reduce the time taken in cleaning.

Sampling because:

- Helps train the overall model with less number of records without severely compromising accuracy.
- Less number of records means less time for training and less memory consumption.

The Challenge:

- Selecting the samples that are representative of the whole dataset.
- The cost for selecting a representative sample can sometimes be more than just training the samples.

3. Method

In this section, we will explain what changes we made to HoloClean code files to achieve the desired goal. We have made changes to two different files in the HoloClean engine (Featurize_dataset.py and Domain.py).

1. The most important step in our model is to select attributes that represent the dimensions in the dataset. These are the attributes which do not vary a lot. For this purpose we analyse the columns involved in denial constraints

and check for dependencies between the columns. More often, these dimension columns are the ones which have been mentioned in the DC constraint files and show a strong Functional Dependency.
For example: We analysed the hospital dataset and chose ‘ProviderNumber’, ‘HospitalType’, ‘HospitalName’, ‘Address,’City’, ‘State’, ‘ZipCode’, ‘CountyName’ as our dimensional columns.

2. While generating domain, function generate_domain passes through each and every cell to generate the domain. We create a new variable “samp_ind” and we implement our logic to sample by updating this variable.
3. The core logic here is to sample only unique combinations of the dimensional columns. We identify the unique combinations of dimensional columns in the original dataframe, which in turn allows us to turn “samp_ind” = 1 for those tuple’s dimensional attributes and “samp_ind” = 0 for duplicate cells in dimensional attributes. For rest of the cells we put “samp_ind” = 1 as we do not want to filter them out from training.
4. The new “samp_ind” variable is concatenated along with the domain data and passed forward so that we can use it to sample the points at a later stage. This makes sure we sample only from training data and not the initial whole dataset.
5. Now the training points are passed to featurized_dataset to generate the feature tensor. Inside this function HoloClean parses each cell and its entire domain to generate the feature tensor. Here we identify which cells to consider for training from all the given cells with the help of “samp_ind” variable which had been set earlier.
6. The existing SQL query helps us decide on which cells the training should be done. It filters out the dk cells from training set, which could not be weak labelled. We updated this SQL query to filter out dimension table duplicates cells from going forward for training.
7. The engine further generates feature tensor and trains the HoloClean model normally. We have not made any changes to this part of the process.

We have worked on the following datasets in our project: Hospital, Food, and some more datasets (like npyd) which lack few files needed for configuration, but we managed to achieve a remarkable reduction in training time (up to 45%) with minimal compromises to the evaluation measures.

4. Evaluation

In this section we have evaluated the results we got from our sampling technique on couple of datasets.

Evaluation	**No Sampling**	**With Sampling**
Precision	1.00	0.96
Recall	0.46	0.47
F1-Score	0.63	0.63
Training points	6592	4661
Execution Time	87 sec	60 sec

Table 1. Reflects the results for Hospital dataset

- As a result of reduction in number of points considered for training, we observe drop in the execution time of cleaning process.
- Correct repairs increased with sampling technique, thus increasing the recall.
- Meanwhile in our model there were incorrect repairs (11), thus decreasing the precision.

Evaluation	**No Sampling**	**With Sampling**
Precision	0.64	0.65
Recall	0.47	0.47
F1-Score	0.54	0.55
Training points	24714	20418
Execution time	167 sec	133 sec

Table 2. Reflects the results for Food dataset

- We have observe slight increase in performance for precision. Total repairs decreased but correct repairs are the same making precision slightly increased.
- As a result of reduction in number of points considered for training, we observe drop in the execution time of cleaning process.
- Since we have improved slightly in precision we also gained in F1 score, meaning low false positives and low false negatives.

For the nypd dataset we observed the following results:

Evaluation	**No Sampling**	**With Sampling**
Time to fit repair model	1929.78 sec	1061.49 sec
Training points	67769	57486
Training accuracy	92.87	94.73

5. Future Work

We would like to make few enhancements to our current model, such that when we sample, the impact on the precision of the model is minimal. Further, we want to work on a sampling technique that can be used to sample directly from the clean data instead of sampling from training points, so that we can effectively pick up points from beginning, helping us to reduce the time and memory consumed for generating domain and features for cells that may not be useful. One another future task could be to automate the procedure of picking the dimension for the sampling task. The current sampling strategy could also be modified to add another layer of sampling for attributes that are not identified as dimensional.

Reference:

[1] Rekatsinas, T., Chu, X., Ilyas, I.F. and Ré, C., 2017. Holoclean: Holistic data repairs with probabilistic inference. *Proceedings of the VLDB Endowment*, *10*(11), pp.1190-1201.