CAFE: Constraint-Aware Feature Extraction from Large Databases

Mahdi Esmailoghli
TU Berlin
Berlin, Germany
esmailoghli@tu-berlin.de

Ziawasch Abedjan
TU Berlin
Berlin, Germany
abedjan@tu-berlin.de

1. INTRODUCTION

Data scientists spend 80% of their time on pre-processing tasks, such as feature extraction, and just 20% on training the machine learning (ML) model [1]. While there is already a body of research on how to extract features from a given dataset [3] a rather neglected problem is the availability of useful features in the first place. Oftentimes, the data scientists have to enrich their datasets with more features from other sources to obtain reasonable prediction models.

An approach for such a system would be to search for similar tables in an external data source and to join all the relevant columns as the new feature set [5]. But this approach suffers from two main issues: first, adding all possible features to the data will lead to a high dimensional dataset suffering from the curse of dimensionality [4], and second, the solution might add features that are considered low-quality or misleading by the user. At the same time, users may want to uphold certain constraints on the new features, such as consistency of values, interpretability, or fairness that are necessary to have a high quality data for the downstream ML task. As each of these constraints might require arbitrary different filtering routines, a system is required that is extendable with regard to constraints, filtering routines, and their optimal execution order. Current solutions [2, 5] focus on enriching the dataset without considering the impact of the added information on the downstream ML models and how they correlate with the target values.

2. CAFE

To solve the two aforementioned problems, we propose Cafe, which extracts features from a given database and filters them according to given user constraints while maximizing the accuracy of the downstream ML task. Our proposed method maps the given constraints to low-level pruning strategies and creates an optimal execution plan similar to a DBMS optimizer.

Cafe uses an inverted index to calculate column similarity and find candidate tables with potential feature candidates.

To enforce high-level user-constraints but maintain optimization potential, we propose a two-stage definition for constraints. We distinguish high-level constraints that bundle low-level pruning strategies. We already include default pruning strategies for common user constraints.

The execution order of the pruning strategies affects the runtime and downstream ML accuracy. The goal is to achieve maximum accuracy and optimize the runtime at the same time. For this purpose, we propose a sample-based estimation approach for the execution planner of Cafe. The execution planner distinguishes two types of strategy relationships: independent (the order does not change the final result) and dependent (the order will affect the final dataset and its accuracy). Currently, the execution planner uses a random sample of the retrieved columns to predicts runtime, accuracy, and number of affected features by the strategies. Then Cafe calculates average runtime spent on each affected features: $t_i = \frac{r_i}{c_i}$ where r_i and c_i are the runtime and number of affected features of strategy i, respectively.

Using the estimated metadata, first, the planner sorts all the strategies based on the accuracy gain. Then the planner identifies chains of consecutive independent strategies. Within those chains, Cafe can change the order without affecting the final set of features and sort them by t_i.

3. CHALLENGES AND FUTURE WORKS

So far, we defined strategies that reduce the dimensionality of the data by removing features. Finding the best execution plan on any arbitrary list of functions would be challenging. Another direction for future work is multi-query-column enrichment. Currently, the user selects one query column to retrieve the external information based on, but she might want to select more than one column. Selecting more that one query column needs more advanced indexing strategy to retrieve similar tables in an efficient way.

4. REFERENCES

[1] D. Deng et al. The data civilizer system. In *CIDR*, 2017.
[2] J. Eberius et al. Top-k entity augmentation using consistent set covering. In *SSDBM*, page 8, 2015.
[3] J. M. Kanter and K. Veeramachaneni. Deep feature synthesis: Towards automating data science endeavors. In *DSAA*, pages 1–10, 2015.
[4] M. Verleysen and D. François. The curse of dimensionality in data mining and time series prediction. In *IWANN*, pages 758–770, 2005.
[5] M. Yakout et al. Infogather: entity augmentation and attribute discovery by holistic matching with web tables. In *SIGMOD*, pages 97–108, 2012.

10th Annual Conference on Innovative Data Systems Research (CIDR '20) January 12-15, 2020, Amsterdam, Netherlands.