The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Dataset Evolver: An Interactive Feature Engineering Notebook

Fatemeh Nargesian, Udayan Khurana,[2] Tejaswini Pedapati,[2]
Horst Samulowitz,[2] Deepak Turaga,[2]
[1]University of Toronto, fnargesian@cs.toronto.edu,
[2]IBM Research, {ukhurana, tejaswinip, samulowitz, turaga}@us.ibm.com

Abstract

We present DATASET EVOLVER, an interactive Jupyter notebook-based tool to support data scientists perform feature engineering for classification tasks. It provides users with suggestions on new features to construct, based on automated feature engineering algorithms. Users can navigate the given choices in different ways, validate the impact, and selectively accept the suggestions. DATASET EVOLVER is a pluggable feature engineering framework where several exploration strategies could be added. It currently includes meta-learning based exploration and reinforcement learning based exploration. The suggested features are constructed using well-defined mathematical functions and are easily interpretable. Our system provides a mixed-initiative system of a user being assisted by an automated agent to efficiently and effectively solve the complex problem of feature engineering. It reduces the effort of a data scientist from hours to minutes.

Feature Engineering

Feature engineering is the task of constructing new features for a dataset by applying transformation functions (e.g. arithmetic and aggregate operators) on existing features with the goal of improving prediction performance. It is often a lengthy process of trial and error, relying heavily on the domain expertise of the data scientist.

Feature engineering is usually coupled with model selection and hyperparameter optimization. In order to assist users to choose learning algorithms and set hyperparameters (including feature selection and preprocessing algorithms) to optimize performance, tools such as Auto-sklearn (Feurer et al. 2015), Hyperopt-sklearn (Komer, Bergstra, and Eliasmith 2014), and Auto-weka (Thornton et al. 2013) have been developed on sklearn library and Weka. Auto-weka leverages Bayesian optimization to fit a probabilistic model that captures the relationship between hyperparameters and the measured performance (Feurer et al. 2015). Auto-sklearn takes into account past performance of algorithms and parameters on similar datasets and constructs ensembles from the models evaluated during the past optimizations. On the other hand, Hyperopt-sklearn adopts an evaluation-based online search strategy to explore the space of possible configurations of classification and feature selection algorithms (Komer, Bergstra, and Eliasmith 2014). All tools mentioned above consider feature preprocessing and feature selection (e.g. scaling and Principal Component Analysis) as parameters during hyperparameter optimization; however, they do not explicitly provide any solutions for efficient and effective interpretable feature engineering.

Building a mixed-initiative data science environment requires feature engineering techniques that allow exploring and pruning of the space of new features interactively without compromising the quality of constructed features. Within DATASET EVOLVER, we include two search and pruning strategies based on meta-learning (Learning Feature Engineering) and reinforcement learning (Cognito). These provide different levels of control and interaction with users, based on their preferences and expertise.

Learning Feature Engineering (LFE) performs feature space exploration using a meta-learning predictor (Nargesian et al. 2017). It predicts effective transformations for features without relying on model evaluation or explicit feature expansion and selection. To achieve that, for each transformation function LFE leverages a Multi-Layer Perceptron classifier, that given class labels predicts whether the transformation and feature(s) that it should be applied to derive an effective feature. LFE suggests the promising transformations, by combining the predictions across these classifiers. To generalize across datasets, LFE captures the correlations between feature values and class labels in a data structure called *Quantile Sketch Array*, which is a stack of fixed-size distribution-based summary of feature values per class label. The empirical evaluation of LFE on a large number of datasets has shown its effectiveness in constructing new features. Moreover, the implementation of LFE integrated with DATASET EVOLVER has the average response time of one millisecond for recommending transformations per feature.

Cognito provides a search-based automation to feature engineering (Khurana et al. 2016; Khurana, Samulowitz, and Turaga 2018). It is based on the hierarchical exploration of a *transformation tree* which is a directed acyclic graph, where nodes are versions of a given dataset and edges are transformations. To steer exploration, Cognito uses model validation score as feedback. Cognito combines greedy heuristic exploration strategies, handcrafted based on the understanding of the human approach to the process, with strategies

Table 1: DATASET EVOLVER Feature Engineering API.

Methods	Description
`FEExplorer`(dataframe)	Initializes a feature engineering environment.
`explorer`(mode, vis)	Specifies a feature exploration strategy. mode = '*linear*' invokes LFE and mode = '*tree*' invokes Cognito (default: *linear*). vis = *True* or *False* is used to display visualization (default: *False*).
`first`(vis)	Displays the best transformation for a feature. Each new feature is juxtaposed next to the original feature and visualized if vis = *True*.
`next`(vis)	Displays the recommended features one after another.
`add_features`(transformation_list)	Applies selected transformations on features and adds constructed features to the dataset.
`auto_add_features()`	Adds all suggested features to the dataset automatically.
`compare`(base_dataset, new_dataset)	Displays the evaluation matrices of the original and transformed dataset.

inspired from reinforcement learning to effectively play the exploration-exploitation trade-off within a constrained budget. During the hierarchical exploration of transformation trees, Cognito supports compositions of transformations. In each step, users can observe the transformations that were applied and their corresponding impact on prediction performance. This makes Cognito a transparent technique to users.

While LFE is prediction-based, Cognito is exploration-based. In our ongoing work, we combine the prediction-based and exploration-based techniques in a two-way positive-feedback loop. The feature suggestions by LFE can guide Congito to bias the search in the right direction, while Cognito's model validation scores help LFE refine its feature engineering patterns and improve its prediction power.

System Overview

DATASET EVOLVER is implemented in a Jupyter notebook. Data scientists working in DATASET EVOLVER environment can perform arbitrary data science tasks such as missing value impuation and model training, and when necessary ask DATASET EVOLVER for feature engineering suggestions.

As demonstrated in Table 1, DATASET EVOLVER offers two different exploration strategies, *linear* (LFE) and *tree* (Cognito). Upon choosing linear mode, DATASET EVOLVER computes feature recommendations and orders them based on their scores. The user can then navigate through these suggestions using the `next` method. To help the user understand the impact of the recommended transformation on the data, DATASET EVOLVER visualizes the correlation between feature values and class labels before and after applying the transformation. The user can then add the recommended features of choice to the dataset by invoking `add_features` or ask DATASET EVOLVER to pick the best features by invoking `add_auto_features`. Furthermore, by choosing *tree* mode, DATASET EVOLVER visualizes the hierarchy of features and users can perform an online exploration of the hierarchy using various classification algorithms and tree search strategies.

Demonstration

We will demonstrate a fully functional implementation of DATASET EVOLVER and highlight its feature engineering functionalities through different scenarios. Figure 1 shows a feature engineering scenario in DATASET EVOLVER. We will show how users can load a dataset of interest, in DATASET EVOLVER environment; ask for transformation

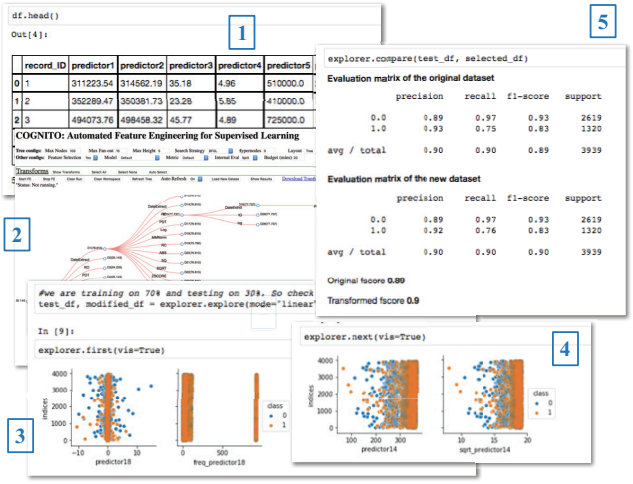

Figure 1: DATASET EVOLVER Snapshots.

suggestions by LFE or Cognito; and inspect the impact of recommended transformations on classification quality using visualization and quantitative statistics. The screencast video of DATASET EVOLVER can be found at: https://www.youtube.com/watch?v=4T8KaeOn-2Y.

References

Feurer, M.; Klein, A.; Eggensperger, K.; Springenberg, J. T.; Blum, M.; and Hutter, F. 2015. Efficient and robust automated machine learning. In *NIPS*, 2755–2763.

Khurana, U.; Turaga, D.; Samulowitz, H.; and Parthasarathy, S. 2016. Cognito: Automated Feature Engineering for Supervised Learning. In *ICDM*, 1304–1307.

Khurana, U.; Samulowitz, H.; and Turaga, D. 2018. Feature engineering for predictive modeling using reinforcement learning. In *Proceedings of AAAI (to appear)*.

Komer, B.; Bergstra, J.; and Eliasmith, C. 2014. Hyperopt-sklearn: Automatic hyperparameter configuration for scikit-learn. *ICML AutoML Workshop*.

Nargesian, F.; Samulowitz, H.; Khurana, U.; Khalil, E. B.; and Turaga, D. 2017. Learning feature engineering for classification. In *IJCAI*, 2529–2535.

Thornton, C.; Hutter, F.; Hoos, H. H.; and Leyton-Brown, K. 2013. Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. In *SIGKDD*, 847–855.