AutoML Using Pipeline Embeddings

Sharath Koorathota, Nian Yi

Introduction

As data scientists spend more and more time on comparing models and tuning parameters for models, we want to find a way to automate these process by using the information of the dataset we use. By doing this, we can shorten our time on finding good models. A good source to use here is the Kaggle competition data which allow us to train our embeddings from different resources of data.

Previous Work

OBOE, a collaborative filtering method for model selection and hyperparameter tuning,(Yang,Akimoto,Kim, Udell,2019) used a matrix of errors of supervised learning method to fit a model to learn the low-dimensional features so that we can predict the cross-validated errors when having new datasets.They also adopt a bilinear model to simplify the model. The system successfully solved two problems including how to choose an initial model within a time frame and how to improve on initial guess giving more sources by the new data frame(such as library or meta-data features).

OBOE system perform well compared to other AutoML methods by making a strong assumption about the bilinearity. The system is now works only for one type of loss metrics and only single algorithm. However, we want to use a pipeline to certain dataset instead of a single machine learning algorithm.

The Kaggle Competition Dataset

We will use the information and meta-data collected from kaggle competitions. To be specifically, the data type we need to use(audio,image,video and etc),the data format(csv,text and etc), the type of task we need to do(classification or regression), the description of the problem on kaggle competition, and what kind of libraries and models are used in the winning solutions.

Problem and Goal

Our project extends on time-constrained collaborative filtering methods to previously-unexplored datasets - specifically, using data and performance metrics from competitions featured on Kaggle, an online community of data scientists and machine learners. [CITATION] Past methods utilized transformations such as principal component analysis on error matrices to yield component data and model latent meta-features. Our work aims to develop learnable embeddings for model pipelines, instead of component matrices, that can be utilized for unseen datasets. Using performance metrics calculated for a subset of possible models, a test dataset can potentially receive both an embedding for the dataset characteristics (e.g. using types of features and size of dataset) and a pipeline embedding that captures information on how a large variety of models (and their associated hyperparameters) are expected to perform on the test dataset.

Our primary focus will be on the pipeline embeddings, and will be accomplished in four stages: (1) select a portion of the Kaggle dataset and model configurations (100 each) to calculate accuracy metrics for Kaggle datasets, (2) define whether the dataset requires a classification or regression solution, so that OBOE is able to consider this information when calculating the performance metric, (3) initialize and training pipeline embeddings (of 512 dimensions) to map model configurations to performance (similar to the offline phase in OBOE, and using the authors openly-available code), and (4) predict model accuracy, calculated using pipeline embeddings, on test datasets to make a recommendation on the top three models.

Evaluation

To evaluate our recommended model suggestions, we will first compare the performance (accuracy or mean squared loss) of our top three recommended algorithms to the top three performance across all possible models in our pool. In addition, we will also compare the model type of our top recommendation (e.g. K-nearest neighbor, CNN) to the winner of the Kaggle competitions. We expect that, because we will be using pre-determined model configurations defined by Yang et al, there will be a significant difference in accuracy from the Kaggle winner, but that our model performance metrics will rank highly from possible models in our pool.

References

Yang, Y Akimoto, DW Kim M Udell (2018), Oboe: Collaborative filtering for automl initialization.. *arXiv preprint arXiv:1808.03233*.