HA3: Learning to Rank. Report. Aisana Bolatbek, 201762234

Task definition

Obtain document ranking formula using machine learning methods

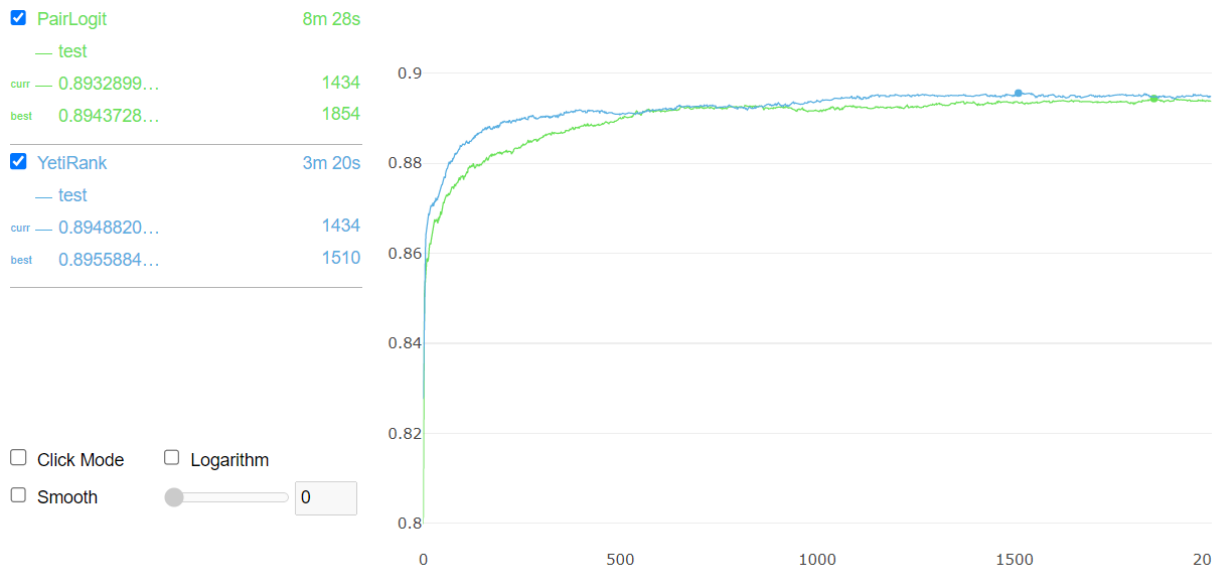
Data description

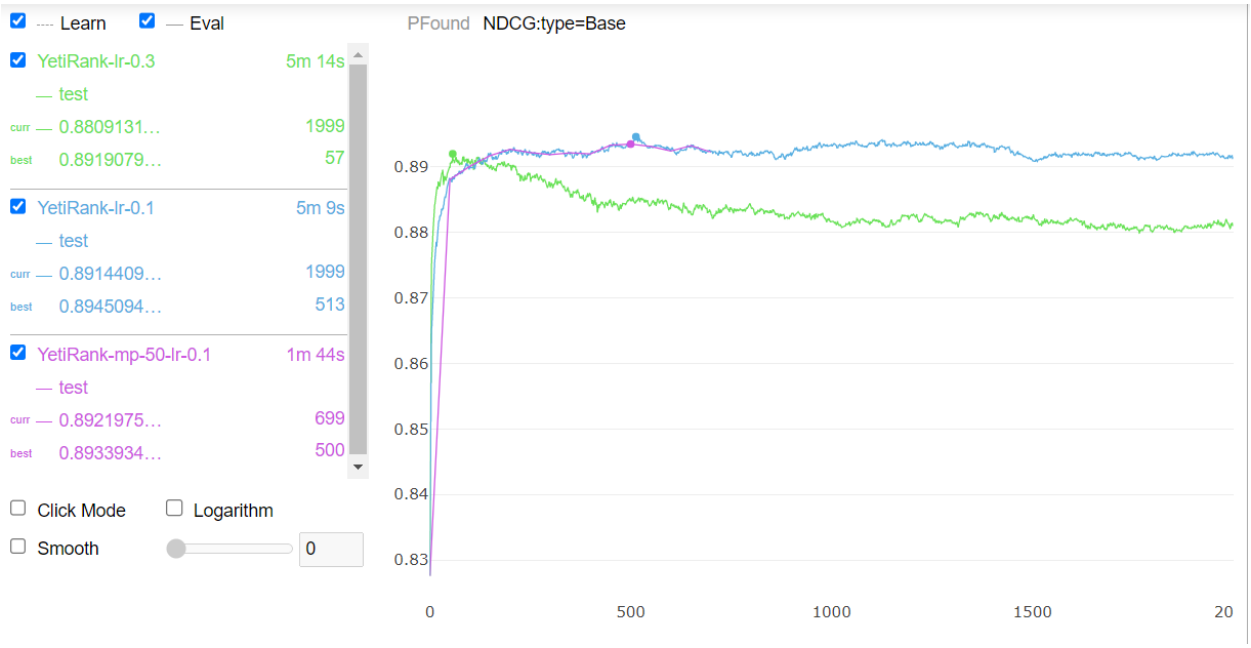
Real data from yandex: feature vectors of query-document pairs and relevance judgements, divided into training and testing sets. 245 features, that are either binary or continuous, taking values in range [0, 1]. Also wikiIR collection for the optional task.

Methods and tools

Python, Jupyter Notebook, Catboost

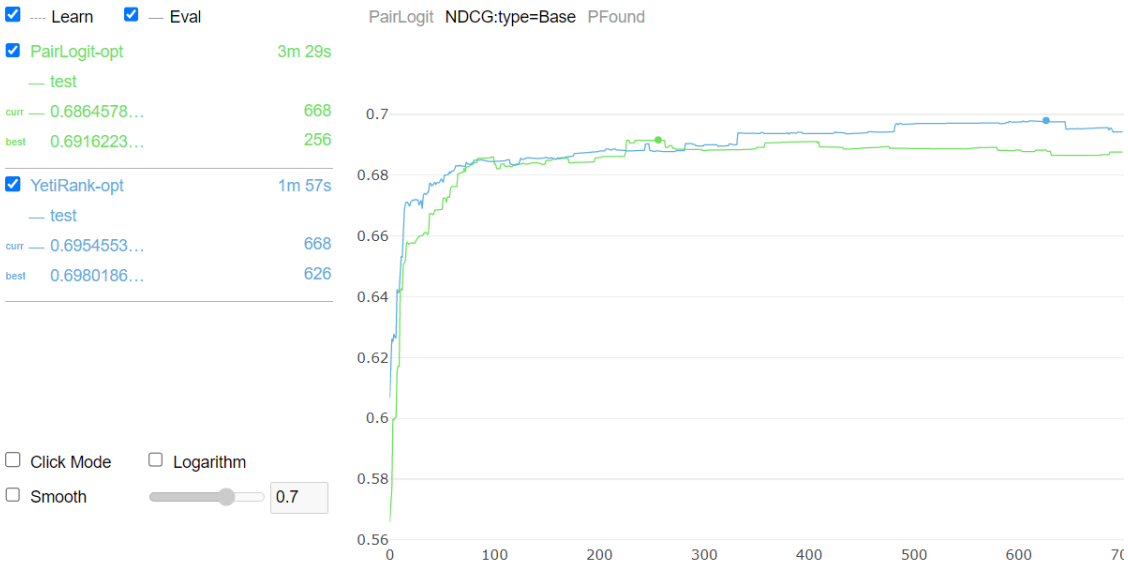
Results and interpretation





PairLogit is rather slow then YetiRank, but their results are quite similar. I tuned only two parameters: learning rate and metric period. With higher learning rate we got the best result

rather faster, but I believe that 0.3 was too high for YetiRank, since it started to return slightly worse results after reaching maximum (2 picture). Adjusting metric period helped to speed up training process of YetiRank (2 picture), but didn't do much for PairLogit. Eventually, I decided to choose default variants of 2 methods (1 picture).



For optional task I extracted such features: query and document length in words, # of matched q/d terms, phrase match (0/1), BM25 score. I got the best results mostly by 700 iterations, but I didn't experiment more than this. I believe that target metric is not great, but it doesn't seem to be underfitted. Maybe, it would be better to extract more features.

Analysis (including failures and possible improvements)

First, I re-formatted data into appropriate one. Then, started learning process. I didn't do a lot of experiments. Maybe I should investigate CatBoost tuning parameters more. Also, for optional task, result is not very good, but I could not say that there is underfitting, maybe, I should extract other features. Also, I coded myself to get features, maybe I should learn how to let Elastic to calculate all the scores.

Link to Github