Ordered sets for Data Analysis

Aleksey Ryabykin

12/12/22

Datasets

I have chosen three datasets to check the base algorithm and my proposed one performances. All of them belong to the UCI Machine Learning Repository:

- Car dataset:

Instances: 1728 Features: 7

- Mushrooms dataset:

Instances: 8124 Features: 23

- Congressional voting records:

Instances: 435 Features: 17

Car Dataset

Mushrooms dataset

Congress Dataset

Initial algorithm

Assume that we want to make a prediction for description $x \subseteq M$ given the set of training examples $X_{\text{train}} \subseteq 2^M$ and the labels $y_x \in \{\text{False}, \text{True}\}$, corresponding to each $x \in X_{\text{train}}$.

First, we split all examples X_{train} to positive X_{pos} and negative X_{neg} examples:

$$X_{\text{pos}} = \{x \in X_{\text{train}} \mid y_x \text{ is True}\}, \quad X_{\text{neg}} = X \setminus X_{\text{pos}}.$$

To classify the description x we follow the procedure:

1. Count the number of counterexamples for positive examples:

   For each positive example $x_{\text{pos}} \in X_{\text{pos}}$ we compute the intersection $x \cap x_{\text{pos}}$. Then, we count the counterexamples for this intersection, that is the number of negative examples $x_{\text{neg}} \in X_{\text{neg}}$ containing intersection $x \cap x_{\text{pos}}$;

2. Dually, count the number of counterexamples for negative examples.

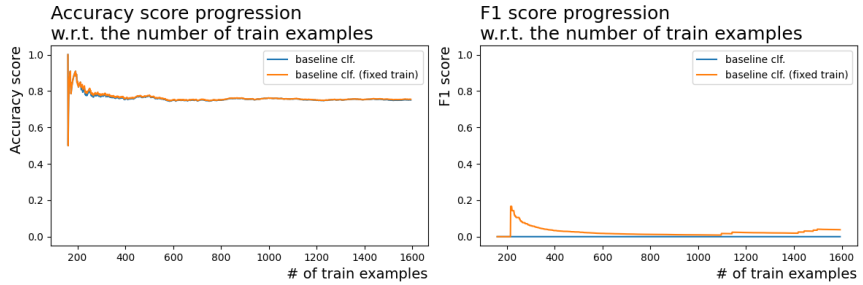
Finally, we compare the average number of counterexamples for positive and negative examples. We classify as being positive if the number of counterexamples for positive examples is smaller the one for negative examples.

Proposed improvements to the initial algorithm

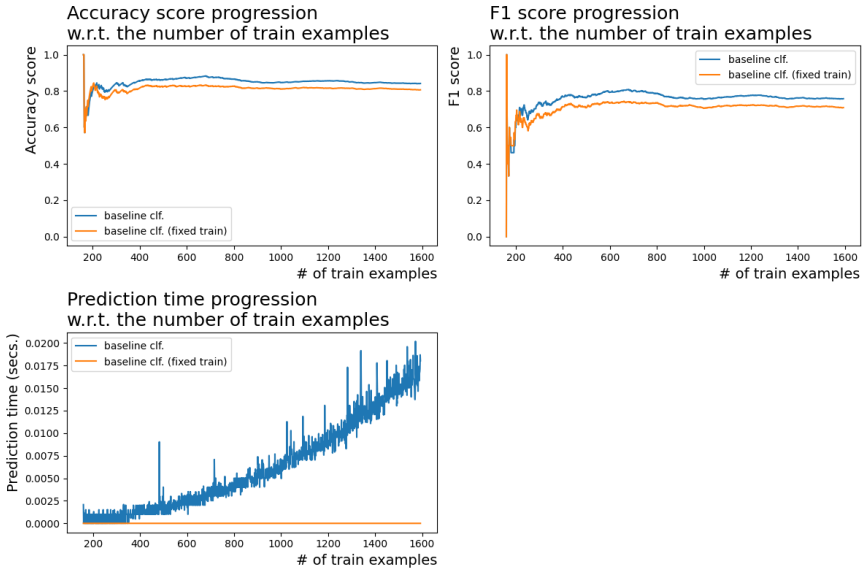
The core idea is the same as for the initial algorithm. But the chosen path to calculate the intersections is different with the previous one. The procedure can be expressed in terms of matrix multiplication of bitmasks that had been obtained after binarization.

Also, there is some bad influence of proposed in the initial algorithm normalization of counts of counterexamples.

Comparison



Initial algorithm



Improved algorithm

Results (Accuracy scores)

Data	Catboost CLF	DecisionTree CLF	RandomForest CLF	XBGRF CLF	LGBM CLF	Initial Algorithm	Improved Algorithm
Car	0.958	0.816	0.743	0.796	0.813	0.78	0.78
Mush rooms	1	0.926	0.917	0.932	0.927	0.93	0.93
Cong ress	0.952	0.966	0.961	0.971	0.957	0.7	0.92

Results (F1 scores)

Data	Catboost CLF	DecisionTree CLF	RandomForest CLF	XBGRF CLF	LGBM CLF	Initial Algorithm	Improved Algorithm
Car	0.921	0.838	0.829	0.844	0.865	0.18	0.78
Mush rooms	1	0.937	0.930	0.944	0.941	0.9	0.91
Cong ress	0.949	0.965	0.961	0.970	0.957	0.66	0.9

Results (Elapsed time)

Data	Catboost CLF	DecisionTree CLF	RandomForest CLF	XBGRF CLF	LGBM CLF	Initial Algorithm	Improved Algorithm
Car	16s	90ms	480ms	220ms	75ms	839ms	560ms
Mush rooms	17s	440ms	850ms	670ms	156ms	3min 13s	43.5s
Cong ress	1.1s	70ms	360ms	90ms	65ms	303ms	58ms

Results

Code reduction

≈50%

Chosen datasets

3

Time improvement

≈6.5x

Average quality improvement

≈23%

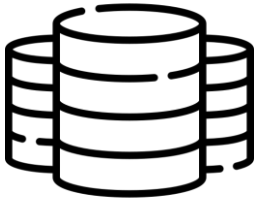
Number of comparing models

5

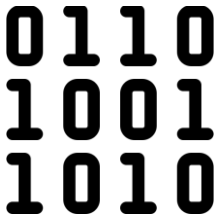
Pattern structures

0

Inferences



Little data



Binary

Interpretable