OSDA. Project. Data Analysis

The project was done by Grigory Kryukov.

Data

We will work with data on the game of Tic-Tac-Toe. The datasets are taken from the files attached to the task. In addition, in the last part, we will independently generate a dataset in which the properties of the binary relation (asymmetry and transitivity) will be indicated.

Algorithms

All algorithms take as input an array of metrics and classes for the training sample and an array of metrics for the test sample and return the prediction of the test sample classes.

1. The basic algorithm of LazyFCA. The method is implemented on the basis of generators.

For each of the objects from the plus-context, we find the intersection of its description and the description of the object from the test sample. If it does not occur in any of the descriptions of negative examples, then we increase by 1 the number of "arguments in favor of the fact that the object from the test sample is positive".

For each of the objects from the negative context, we find the intersection of its description and the description of the object from the test sample. If it does not occur in any of the descriptions of the plus examples, then we increase by 1 the number of "arguments in favor of the fact that the object from the test sample is negative."

Next, we compare the number of "arguments". If there are more "arguments" in favor of the fact that the test sample object is positive, we accept y_pred[i] = '+'. If there are more "arguments" in favor of the fact that the test sample object is negative, we accept y_pred[i] = '-'. If the number of arguments is equal, then we accept "+" with probability p, which can be passed as an argument.

This algorithm was described in the description of the task, so I will not repeat the formulas.

2. Average intersection cardinality.

Calculates the average intersection cardinality of all plus-context objects with the test sample object and the average intersection cardinality of all minus-context objects with the test sample object. We accept the hypothesis for which the average intersection cardinality is greater.

$$avg_{+} = \frac{\Sigma|g' \cap g_{i}^{+}|}{|G_{+}|}, avg_{-} = \frac{\Sigma|g' \cap g_{i}^{-}|}{|G_{-}|}$$

$$avg_{+} > avg_{-} \Rightarrow returns\ +$$

$$avg_{+} \leq avg_{-} \Rightarrow returns\ -$$

3. The intersection cardinality of k nearest objects.

Let's define the proximity of vectors as the cardinality of their intersection (the greater the power of the intersection, the closer the objects are). For each of the plus and minus contexts, we will find the k objects closest to the object from the test sample, and compare the average capacities for the k nearest objects. We accept the hypothesis for which the average intersection cardinality is greater.

$$Choose\ g_{1}^{+},..., g_{k}^{+}\ from\ G_{+}: |g' \cap g_{i}^{+}| \rightarrow max$$

$$Choose\ g_{1}^{-},..., g_{k}^{-}\ from\ G_{-}: |g' \cap g_{i}^{-}| \rightarrow max$$

$$avg_{+} = \frac{\Sigma_{i=1}^{k}|g' \cap g_{i}^{+}|}{k}, avg_{-} = \frac{\Sigma_{i=1}^{k}|g' \cap g_{i}^{-}|}{k}$$

$$avg_{+} > avg_{-} \Rightarrow returns\ +$$

$$avg_{+} \leq avg_{-} \Rightarrow returns\ -$$

4. Advanced LazyFCA algorithm.

This is an ensemble of algorithms # 1 and # 3. First, we apply algorithm # 1. If the difference modulo the "arguments" in favor of the fact that the object of the test sample takes a positive or negative value reaches the value d, then we accept this result. Otherwise, if the number of arguments in favor of each of the classes is approximately equal, we apply the solution using algorithm # 3.

The metric results on the datasets.

Let's find the average running time and the average value of quality metrics for all implemented algorithms.

We will use 10 datasets for training and testing, which are presented in the materials for the task (Tic-Tac-Toe).

I choose k=10 and d=5 as parameters for algorithms #3 and #4.

	baseline_FCA	avg_cardinality	topk_cardinality	smart_FCA
Accuracy	0.9405	0.6588	0.9896	0.9532
True Positive	0.6535	0.4278	0.6524	0.6535
True Negative	0.2871	0.2310	0.3372	0.2998
False Positive	0.0595	0.1155	0.0093	0.0468
False Negative	0.0000	0.2257	0.0011	0.0000
TPR	1.0000	0.6545	0.9983	1.0000
TNR	0.8303	0.6668	0.9736	0.8668
NPV	0.1697	0.3332	0.0264	0.1332
FPR	1.0000	0.5084	0.9970	1.0000
FDR	0.0834	0.2131	0.0142	0.0668
Precision	0.9166	0.7869	0.9858	0.9332
Recall	1.0000	0.6545	0.9983	1.0000
Running time	20.5779	0.2790	0.2910	20.5914

The algorithm "Intersection cardinality of k nearest objects" shows the best results in the accuracy of the forecast. It is worth noting that algorithms # 1 and # 4 also showed good results, but at the same time they have a relatively long running time.

Cross-validation and selection of optimal parameter values

As we saw earlier, algorithm #3 shows the best results on quality metrics, while it is much faster than algorithms #1 and #4. Therefore, in this section we will look for the optimal value of the parameter k only for the algorithm "Intersection cardinality of k nearest objects".

Let's assume that type I and type II errors lead to the same losses, so we focus only on the accuracy metric. We will use cross-validation (10-Fold).

Accuracy on different parameter values:

k	Accur (TTT)	Accur (Assym)	Accur (Transit)
1	0.488	1	0.964
2	0.513	1	0.965
3	0.659	0.999	0.973
4	0.79	0.998	0.97
5	0.87	0.996	0.979
6	0.933	0.98	0.965
7	0.961	0.976	0.976
8	0.968	0.956	0.968
9	0.98	0.95	0.966
10	0.984	0.932	0.954
11	0.991	0.925	0.946
12	0.989	0.909	0.947
13	0.991	0.9	0.946
14	0.987	0.889	0.942
15	0.987	0.886	0.95
16	0.986	0.883	0.943
17	0.969	0.885	0.945
18	0.981	0.883	0.946
19	0.98	0.884	0.949

TTT - Tic-Tac-Toe datasets
Asymm - Checking the asymmetry of the binary relation
Transit - Checking the transitivity of the binary relation

Conclusion

As you can see, different values of the parameter k are optimal for different tasks.

To determine the outcome of the tic-tac-toe game, the optimal values were k=11 and k=13.

To determine whether the binary relation is asymmetric, k=1, k=2 were optimal: with these parameter values in our experiments, the forecast was always correct.

To determine whether a binary relation is transitive, k=5 turned out to be optimal.

It turns out that for each task, the optimal value of k may be different. In any case, the method shows very decent results, in most experiments the accuracy of the forecast exceeds 95%.