# A One-Class Classification decision Tree based on kernel density estimation

Parag Goyal (IIT2018164), Bhavya Girotra (IIT2018167), Yash Katiyar (IIT2018170)

B-Tech IT, 6th Semester, Indian Institute of Information Technology Allahabad

**Abstract -** *One-class Classification (OCC) is an area of machine learning which addresses prediction based on unbalanced datasets. Basically, OCC algorithms achieve training by means of a single class sample, with potentially some additional counter-examples. The current OCC models give satisfaction in terms of performance, but there is an increasing need for the development of interpretable models. In the present work, we propose a one-class model which addresses concerns of both performance and interpretability. Our hybrid OCC method relies on density estimation as part of a tree-based learning algorithm, called One-Class decision Tree (OC-Tree).*

*Within a greedy and recursive approach, our proposal rests on kernel density estimation to split a data subset on the basis of one or several intervals of interest. Thus, the OC-Tree encloses data within hyper-rectangles of interest which can be described by a set of rules. Against state-of-the-art methods such as Cluster Support Vector Data Description (ClusterSVDD), One-Class Support Vector Machine (OCSVM) and Isolation Forest (iForest), the OC-Tree performs favorably on a range of benchmark datasets. Furthermore, we propose a real medical application for which the OC-Tree has demonstrated its effectiveness, through the ability to tackle interpretable diagnosis aid based on unbalanced datasets.*

## I. Introduction

As precious assets of knowledge extraction, data are massively collected in the fields of industry and research, day by day. Though valuable, the prolif-eration of data requires attention upon processing. In particular, unbalanced datasets may be hardly addressed through the classical scheme of multi-class prediction. The practice of One-Class Classification (OCC) has been developed within this consideration.

OCC is of major concern in several domains where it may be expensive and/or technically difficult to collect data on a range of behaviors or phenomena. As a matter of fact, one-class classifiers are trained on a single class sample, in the possible presence of a few counter-examples. The resulting models allow to predict target (or positive)patterns and to reject outlier (or negative) ones. Basically, OCC is pursued for outlier (or anomaly) detection.

One-Class Support Vector Machine (OCSVM) and Support Vector Data Description (SVDD) are among the most common OCC methods [4, 5]. OCSVM aims at finding the hyper-plane that separates the target instances from the origin with the wider margin, while SVDD aims at enclosing these instances within a single hyper-sphere of minimal volume.

Kernel Density Estimation (KDE) is another approach which can address OCC intuitively, in computing the non-parametric estimation of a sample distribution. Thresholded at a given level of confidence, this estimation is used to reject any instance located beyond the decision boundary thus established. However, KDE loses in performance and readability towards high dimensional samples. In the present work, we tackle OCC through a hybrid method, called One-Class decision Tree

(OC-Tree), which is intended to combine the benefits of the standard decision tree and KDE.

## II. Motivation

The current methods of OCC give satisfaction, but that is without counting on the advent of explainable artificial intelligence which opens new research horizons for machine learning in encouraging the development of interpretable models. In this regard, some methods have been developed as post-hoc explainers on the predictions of classifiers. But a great challenge remains the development of interpretable models by nature, which provide simultaneously high levels of performance. This challenge is the major source of motivation for the present work.

## III. Algorithm

In a divide and conquer spirit, the implementation of our one-class tree rests on successive density estimations to raise target areas as hyper-rectangles of interest. We assess the relevance of a subdivision against an information gain criterion adapted to OCC issues proposed by [20]. Let us consider $\chi \subset \mathbf{R^d}$ a hyper-rectangle of dimensions d including target training instances. Let us note $\mathbf{A = \{a_1, a_2, \ldots, a_d\}}$ the set of attributes and $\mathbf{X = \{x_1, x_2, \ldots, x_n\}}$ the set of instances. The goal of our proposition is the division of the initial hyper-rectangle $\chi$ in (non necessarily adjacent) sub-spaces $\chi_{ti}$, represented by tree nodes $\mathbf{t_i}$, in absence of counter-examples.

Let us denote as $A_t$ the set of eligible attributes for division at a given node t. Thus, $\mathbf{A_t \subseteq A}$. We note $\mathbf{A_t = \{a_1', a_2', \ldots a_{lt}'\}}$, $\mathbf{l_t}$ being the number of eligible attributes at node **t**, with $\mathbf{l_t \leq d}$ accordingly. At each node t, the algorithm searches the attribute $\mathbf{a_j' \in A_t}$ which best cuts the initial sub-space $\chi_t$ into one or several sub-space(s) $\chi_{ti}$ such that:

$$\chi_{ti} = \{x \in \chi_t : L_{ti} \leq x^{aj'} \leq R_{ti}\} \quad (1)$$

$x^{aj'}$ is the value of instance x for attribute $\mathbf{a_j'}$; $\mathbf{L_{ti}}$ and $\mathbf{R_{ti}}$ are respectively the left and right bounds of the closed sub-intervals raised to split the current node **t** in target nodes $\mathbf{t_i}$, based on attribute $\mathbf{a_j'}$. For each attribute $\mathbf{a_j' \in A_t}$, the algorithm achieves the following steps, at a given node t.

1. Check if the attribute is still eligible and compute the related Kernel Density Estimation (KDE), i.e., an estimation of the probability density function $\hat{f}_j(x)$ based on the available training instances.

2.Divide the space $\chi_t$, based on the modes of $\hat{f}j$ (x).

3. The quality of the division is assessed by the computation of the impurity of the resulting nodes deriving from division.

At each iteration, the attribute that achieves the best purity score is selected to split the current node t in child nodes. If necessary, some branches are pre-pruned in order to preserve the interpretability of the tree. The algorithm is run recursively; termination occurs under some stopping conditions.

## IV. Experimental Protocol

We compared the OC-Tree with three reference methods, namely the ClusterSVDD, One-Class Support Vector Machine (OCSVM) and Isolation Forest (iForest). The comparison of the OC-Tree with ClusterSVDD is highly relevant since both methods pursue similar objectives, i.e., enclosing data within one or several hyper-rectangle(s) and hyper-sphere(s) respectively. ClusterSVDD requires that two parameters should be optimized on a dataset: $\boldsymbol{\nu}$ and **k** which constitute respectively, the

upper bound on the fraction of instances lying outside the decision boundary and the supposed number of clusters.

OCSVM is a standard OCC method to which a comparison is thus worth considering. We considered a gaussian kernel for this method, and we optimized $\boldsymbol{\nu}$ which pursues the same objective as in ClusterSVDD and OC-Tree. Thus, to ensure a fair comparison, we adjusted this parameter in the same way that we did for ClusterSVDD. Finally, a method like iForest provides a relevant benchmark since it is of the same nature as OC-Tree, i.e., a tree-based method, but built in a very different way. Indeed, this ensemble technique aims at the development of decision trees based on a random choice of attributes and thresholds. If the average path length skimmed in the trees is low (resp. high), an instance is predicted as outlier (resp. target). We used the standard parameter settings for this method, since it was shown that the performances are ensured to be quite optimal with such settings.

## V. Benchmark Datasets

In absence of benchmark data for OCC, it is standard practice to convert multi-class problems into one-class ones for evaluation purposes. We thus considered a set of benchmark datasets, where each instance belongs to a class $\mathbf{c_i}$ among a set of C. The relevancy of OC-Tree and of the reference methods on these datasets was assessed in two distinct ways.

A. All the instances, whatever their class, were considered as the representatives of the same class. We injected in this dataset a certain percentage of additional outliers following a uniform distribution.(Approach A)

B. We adopted the one vs rest strategy which consists of considering a class $\mathbf{c_i} \in \mathbf{C}$ as a target one and the others as outliers. In this case, the outliers injected in a given data subset were randomly picked among the representatives of the outlier classes, i.e., $\boldsymbol{C \backslash c_i}$ .(Approach B)

Whether through approach A or B, the resulting dataset was split in a way that two thirds constituted a training set, while the remaining was kept as a test set.

## VI. Model Selection

The OC-Tree and some reference methods rely on a certain number of parameters that have to be adjusted appropriately. This parameter tuning was achieved through a 10-fold Cross-Validation (10-fold CV) procedure, based on the values presented in Table 3. Note that we conducted a grid search in the case where we had to optimize two parameters. The range of values for parameter k, i.e., the number of clusters in ClusterSVDD, has been differentiated depending on the considered dataset and the approach under which the datasets were addressed. More particularly in regards to approach B, it appeared to us reasonable to set a range of [1, 5] as possible values for parameter k, regardless of the considered dataset. Indeed, in this case, each class of the multi-class problem is considered for OCC. Thus, intuitively, one would expect that data are concentrated within a small number of target groupings but at the same time, the presence of a single class may reveal a structure of data different from the one observed in the case of a multi-class problem. That is why k may present higher values than those considered with approach A for some datasets.

Thus, except for iForest, each algorithm was tuned through a CV procedure, in search of the model which presents the best performance in the sense of the F1-score. The model selection was naturally achieved on the training set extracted from each dataset. The selected models were finally assessed against a test set.

## VII. Results

It appears that the OC-Tree performs favorably in comparison to the other reference methods. The improvements achieved against iForest may be explained by the fact that the latter method is properly intended for anomaly detection, and may thus have slightly lower performances when the proportion of outliers in the training set is low [21], especially for proportions of 2% and 5%. Moreover, compared to iForest, the OC-Tree seems globally to better handle the ionosphere and satimage datasets. Actually, the ionosphere dataset has a quite diffuse distribution of data along some dimensions, which involves that some normal instances may lie far away from the others. As it is built on a random choice of attributes, the iForest method is likely to detect these instances as outliers. On the opposite, the OC-Tree is built on attributes which concentrate the instances, so the ones lying outside these concentrations may be really perceived as outliers. As regards the satimage dataset, the low proportion of outliers in such a high dimensional dataset may have disadvantaged the iForest method, with a difference in terms of F1-score that can reach 5%. As regards the performances of OCSVM, they are in some cases lower than OC-Tree, which may be explained by the fact that OCSVM encloses data within a single boundary and can thus not exactly adjust to the structure of data. Finally, as mentioned previously, ClusterSVDD may be sensitive to noise, which explains why the OC-Tree provides better results in some cases.

| Dataset | Noise level | ClusterSVDD | OCSVM | iForest | OC-Tree | |
|---|---|---|---|---|---|---|
| Australian | 2% | 0.986 - 0.921 | 0.995 - 0.908 | 1.000 - 0.926 | 1.000 - 0.965 | (+) |
| | 5% | 0.973 - 0.926 | 0.977 - 0.922 | 1.000 - 0.922 | 1.000 - 0.970 | (+) |
| | 10% | 0.900 - 0.960 | 0.916 - 0.960 | 1.000 - 0.991* | 0.900 - 0.996 | |
| | 15% | 0.877 - 0.961 | 0.882 - 0.935 | 0.955 - 1.000 | 1.000 - 0.961 | (+) |
| Diabetes | 2% | 1.000 - 0.941 | 1.000 - 0.941 | 1.000 - 0.874 | 0.992 - 0.965 | (+) |
| | 5% | 0.998 - 0.965* | 0.988 - 0.957 | 0.996 - 0.914 | 0.992 - 0.911 | |
| | 10% | 0.932 - 0.972 | 0.975 - 0.933 | 0.996 - 0.952 | 0.980 - 0.952 | |
| | 15% | 0.974 - 0.897 | 0.974 - 0.901 | 0.965 - 0.988 | 0.992 - 0.945 | |
| Ionosphere | 2% | 0.972 - 0.914 | 0.972 - 0.897 | 0.981 - 0.879 | 1.000 - 1.000 | (+) |
| | 5% | 0.938 - 0.913 | 0.937 - 0.904 | 0.937 - 0.904 | 1.000 - 1.000 | (+) |
| | 10% | 0.884 - 0.939 | 0.880 - 0.904 | 0.904 - 0.912 | 0.983 - 1.000 | (+) |
| | 15% | 0.828 - 0.946 | 0.824 - 0.920 | 0.832 - 0.929 | 0.982 - 1.000 | (+) |
| Iris | 2% | 1.000 - 0.902 | 1.000 - 0.961 | 1.000 - 0.922 | 1.000 - 0.941 | |
| | 5% | 0.977 - 0.860 | 0.980 - 0.960 | 0.978 - 0.900 | 0.943 - 1.000 | (+) |
| | 10% | 0.979 - 0.920 | 1.000 - 0.940* | 0.958 - 0.920 | 0.978 - 0.900 | |
| | 15% | 0.902 - 0.920 | 0.889 - 0.960 | 0.889 - 0.960 | 0.862 - 1.000 | (+) |
| Satimage | 2% | 0.995 - 0.945 | 0.996 - 0.957 | 0.996 - 0.890 | 0.996 - 0.968 | (+) |
| | 5% | 0.986 - 0.974 | 0.986 - 0.971 | 0.984 - 0.914 | 0.979 - 0.981 | (+) |
| | 10% | 0.991 - 0.981 | 0.977 - 0.946 | 0.952 - 0.922 | 0.981 - 0.969 | |
| | 15% | 0.990 - 0.963 | 0.966 - 0.937 | 0.907 - 0.934 | 0.980 - 0.968 | |
| Segment | 2% | 0.999 - 0.963 | 0.999 - 0.974 | 1.000 - 0.912 | 1.000 - 1.000 | (+) |
| | 5% | 0.974 - 0.970 | 0.978 - 0.970 | 1.000 - 0.940 | 1.000 - 1.000 | (+) |
| | 10% | 0.927 - 0.980 | 0.930 - 0.979 | 1.000 - 0.991 | 0.993 - 1.000 | (+) |
| | 15% | 0.898 - 0.970 | 0.928 - 0.946 | 0.976 - 1.000* | 0.872 - 0.996 | |

| Dataset | Noise level | ClusterSVDD | OCSVM | iForest | OC-Tree | |
|---|---|---|---|---|---|---|
| Australian (-1) | 2% | 0.984 - 0.945 | 0.983 - 0.914 | 0.991 - 0.875 | 0.992 - 0.977 | (+) |
| | 5% | 0.936 - 0.936 | 0.935 - 0.920 | 0.959 - 0.928 | 0.945 - 0.960 | (+) |
| | 10% | 0.902 - 0.960 | 0.919 - 0.912 | 0.941 - 0.896 | 0.890 - 0.968 | (+) |
| | 15% | 0.819 - 0.934 | 0.822 - 0.917 | 0.886 - 0.901 | 0.834 - 1.000 | (+) |
| Australian (+1) | 2% | 0.980 - 0.951 | 0.980 - 0.951 | 0.980 - 0.951 | 0.990 - 0.980 | (+) |
| | 5% | 0.950 - 0.950 | 0.950 - 0.941 | 0.949 - 0.921 | 0.943 - 0.990 | (+) |
| | 10% | 0.906 - 0.950 | 0.932 - 0.950 | 0.947 - 0.891 | 0.901 - 0.990 | (+) |
| | 15% | 0.881 - 0.960 | 0.872 - 0.950 | 0.855 - 0.940 | 0.860 - 0.980 | |
| Diabetes (-1) | 2% | 0.978 - 0.978* | 0.977 - 0.966* | 0.976 - 0.910 | 0.976 - 0.910 | |
| | 5% | 0.957 - 0.978* | 0.956 - 0.967* | 0.954 - 0.922* | 0.952 - 0.878 | |
| | 10% | 0.944 - 0.934 | 0.926 - 0.956 | 0.928 - 0.846 | 0.926 - 0.956 | (+) |
| | 15% | 0.863 - 0.921 | 0.876 - 0.955 | 0.874 - 0.933 | 0.862 - 0.910 | |
| Diabetes (+1) | 2% | 0.981 - 0.933 | 0.980 - 0.903 | 0.980 - 0.879 | 0.982 - 0.988 | (+) |
| | 5% | 0.945 - 0.951 | 0.956 - 0.927 | 0.955 - 0.909 | 0.942 - 0.982 | (+) |
| | 10% | 0.895 - 0.962 | 0.905 - 0.956 | 0.909 - 0.938 | 0.881 - 0.975 | |
| | 15% | 0.853 - 0.938 | 0.858 - 0.938 | 0.871 - 0.919 | 0.856 - 0.963 | (+) |
| Ionosphere (-1) | 2% | 0.974 - 0.881 | 0.967 - 0.690 | 0.971 - 0.810 | 0.977 - 1.000 | (+) |
| | 5% | 0.946 - 0.833 | 0.935 - 0.690 | 0.946 - 0.833 | 0.955 - 1.000 | (+) |
| | 10% | 0.872 - 0.829 | 0.857 - 0.732 | 0.872 - 0.829 | 0.943 - 0.805 | (+) |
| | 15% | 0.889 - 0.930 | 0.861 - 0.721 | 0.895 - 0.791 | 0.905 - 0.884 | (+) |
| Ionosphere (+1) | 2% | 1.000 - 0.960 | 1.000 - 0.960 | 1.000 - 0.893 | 0.986 - 0.973 | (+) |
| | 5% | 0.973 - 0.973 | 0.986 - 0.946 | 0.986 - 0.919 | 0.947 - 0.959 | |
| | 10% | 0.972 - 0.932 | 0.973 - 0.959 | 0.956 - 0.878 | 0.973 - 0.973 | (+) |
| | 15% | 0.920 - 0.958* | 0.909 - 0.972 | 0.920 - 0.958* | 0.861 - 0.944 | |
| Iris (1) | 2% | 1.000 - 1.000* | 1.000 - 1.000 | 1.000 - 0.941* | 1.000 - 0.882 | |
| | 5% | 0.933 - 0.824 | 1.000 - 0.824 | 1.000 - 0.882 | 1.000 - 0.882 | (+) |
| | 10% | 0.938 - 0.833 | 0.938 - 0.833 | 1.000 - 1.000* | 1.000 - 0.889 | (+) |
| | 15% | 0.941 - 0.842 | 0.941 - 0.842 | 1.000 - 1.000* | 1.000 - 0.895 | |
| Iris (2) | 2% | 1.000 - 1.000 | 1.000 - 1.000 | 1.000 - 0.824 | 1.000 - 1.000 | (+) |
| | 5% | 0.944 - 1.000 | 0.944 - 1.000 | 0.944 - 1.000 | 0.944 - 1.000 | (+) |
| | 10% | 1.000 - 1.000 | 0.947 - 1.000 | 1.000 - 1.000 | 1.000 - 1.000 | (+) |
| | 15% | 1.000 - 1.000* | 0.947 - 0.947 | 1.000 - 1.000* | 0.947 - 0.947 | |
| Iris (3) | 2% | 1.000 - 0.824 | 1.000 - 0.824 | 1.000 - 0.824 | 1.000 - 1.000 | (+) |
| | 5% | 0.941 - 0.941* | 0.933 - 0.824 | 0.933 - 0.824 | 0.938 - 0.882 | |
| | 10% | 1.000 - 1.000* | 1.000 - 0.722 | 1.000 - 1.000* | 1.000 - 0.833 | |
| | 15% | 0.929 - 0.684 | 1.000 - 0.789 | 1.000 - 0.895 | 1.000 - 0.895 | (+) |