Data Mining and Warehousing

VI Semester, Department of Information Technology,
Indian Institute of Information Technology, Allahabad, Prayagraj.

Assignment 4
A One-Class Classification decision Tree based on kernel density estimation

Group Members

Shubham Kumar IIT2018115

Raktim Bijoypuri IIT2018125

Aditya Kamble IIT2018126

Aakashdeep IIT2018128

Tejas Mane IIT2018135

Introduction:

As a major component of knowledge discovery, data is being collected at a huge scale. But availability of data is not sufficient as it needs processing to generate more accurate results. OCC is the technique which is helpful where data collection is a difficult task. One-Class Support Vector Machine (OCSVM) and Support Vector Data Description (SVDD) are among the most common OCC methods.

One-class Classification (OCC) is an area of machine learning which addresses prediction based on unbalanced datasets. The performance of present OCC models are quite good, but interpretable models need improvement. Hybrid OCC model developed by Philippe Fortemps satisfies both performance and interpretability problems with a greedy recursive approach. Against the traditional state of the art methods like 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.

Algorithm :

For each attribute $a_0 j \in A_t$, 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 χt, based on the modes of $fj(x)$.
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 (see Sec. 2.4). The algorithm is run recursively; termination occurs under some stopping conditions

At node t, division is executed based on $\hat{f}j(x)$, in four steps.

- Clipping KDE
- Revision
- Assessment
- Shrinking

Results :

Noise level	ClusterSVDD		OC-Tree	
	Precision	Recall	Precision	Recall
2%	0.998	0.917	0.998	0.985
5%	0.995	0.940	0.999	0.987

Table 1: Performance assessment table

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, especially for proportions of 2% and 5%. OC-Tree is built on attributes which concentrate the instances, so the ones lying outside these concentrations may be really perceived as outliers.