ANOMALY DETECTION IN AGGLOMERATIVELY BUILT DENOISED TREES

Nebahat Bozkus

Abstract

In the literature, identifying clusters of similar objects and identifying anomalies/outliers are often discussed together. At the stage of classification of objects, there may be such objects which are not sufficiently similar to others. At this point, a popular question arises: whether these objects should be labelled as anomalies or assigned to the nearest cluster. In this work, we propose a new anomaly detection technique – named LiftOut – which is based on a wavelet-like denoising method called "Lifting". The proposed method works on agglomeratively built trees. LiftOut first detects anomalies in trees, then it removes points labelled as anomalies from trees. The final stage of the algorithm is to find nodes in trees where classes should be placed. LiftOut is applied on real world scenarios, and its performance is compared with the DBSCAN algorithm. While DBSCAN attempt to create new small size clusters for closely placed anomalies, LiftOut algorithm catches high percentage of anomalies and true clusters.

Key words: outlier detection, classification, lifting

JEL Code: C10, C19, C63

Introduction

Clustering and anomaly/outlier detection are two related working area. While clustering finds different structures in a data set and organizes data according to the structures found, in anomaly detection, we try to identify points that significantly differ from the majority of the data. There is no single definition for anomalies; the definition of anomaly varies from application to application. The most general definition for anomaly is provided by Hawkins (1980): anomaly values are observations with different characteristics from other observations, and these observations are generated by different mechanism than the rest of the data.

Anomaly detection methods are divided as statistical methods, proximity-based methods and clustering-based methods. Statistical methods are also branched as parametric

and non-parametric methods, and proximity-based methods are further divided into distance-based, grid-based and density-based methods. Statistical methods make assumptions about the distribution of data or about the model that fits the data. Parametric methods assume that the distribution of the data is known, and parameters of the distribution is estimated using the data. Non-parametric methods do not need to make any assumptions on the distribution of data. The detailed review of statistical methods is available in Han et al. (2012). The proximity-based methods, on the other hand, label any object as an anomaly if the object is distantly placed from its neighbors. One distance-based method is presented by Knorr and Ng (1998). Their method is sensitive to two parameters given by the researcher: the maximum distance of an object to its neighborhood and the minimum number of objects in its neighborhood. Grid-based methods divide data space into multidimensional grids to detect anomalies (Han et al., 2012). If the diagonal length of the cell is well-defined, grid-based algorithms show high performances. The cost of the algorithm is an exponential function of the length of the data; the algorithm slows down as the size and number of data increase. Breunig et al. (2000) proposed a density-based algorithm called LOF (local outlier factor). LOF measures the density of k-neighborhood of an object to detect anomalies. LOF, however, is a computationally expensive algorithm. The final type of anomaly detection methods, clustering-based methods, assumes that "normal" objects belong to large and dense clusters while anomalies belong to small or sparse clusters, or these methods assume that anomalies are not belonged to any cluster. One of the well-known clustering-based methods is called PAM (partitioning around medoids; Kaufmann & Rousseeuw, 1987) algorithm which is based on k-medoids clustering algorithm. While PAM is an efficient method for small size data, the efficiency decreases as data size or number increases. Anomaly detection via a density-based clustering algorithm called DBSCAN was proposed by Ester et al. (1996).

Bozkus and Barber (2023) proposed a lifting-based classification algorithm that detects irregular classes on dendrograms. The algorithm places classes on nodes automatically by eliminating the user manual intervention. This algorithm does not cluster some of the observations, or it finds small-size clusters. Researchers define non-clustered observation(s) and objects in small-size clusters as anomalies using the anomaly definition of Barnett and Lewis (1978): a point or group of points which is inconsistent with the majority of observations is defined as anomaly. In this research, the lifting algorithm is applied to detect all observations that create inconsistency in classification structures before the final classification structure is assigned. Classes are reconstructed after anomalies are removed. The necessary parameters are automatically estimated by the algorithm itself.

1 Method

Wavelets is an orthonormal basis function, and it provides the multiscale division of a data set in Euclidean lattice (see e.g., Daubechies, 1992). Lifting provides the wavelet-like coefficients of a data set having a neighborhood structure. Early works of lifting consider data points placed in irregular Euclidean lattice. Later, Jansen et al. (2009) presented "lifting one coefficient at a time" (LOCAAT) algorithm which works on networks. The data points are assumed to be placed on nodes in networks. Binary trees are special networks, where each node is split into at most two internal nodes/leaves; thus, LOCAAT can be applicable on binary trees (dendrograms). Bozkus and Barber (2023) proposed a classification algorithm for binary trees based on the LOCAAT algorithm. In this work, an anomaly detection technique based on their method is presented.

1.1 LOCAAT algorithm

Noise-corrupted data is defined as $g(s_i)$ defined as

$$g(s_i) = h(s_i) + \varepsilon_i,\ i = 1, \dots, n \qquad (1)$$

where the s_i are nodes in a dendrogram, and $\varepsilon_i \sim N(0, \sigma^2)$ independently, and our interest is in $h(s_i)$. We suppose that the function $g(s)$ has the form

$$g(s) = \sum_{\ell=1}^{n} c_{n,\ell}\phi_{n,\ell}(s),$$

and the scale function is defined as

$$\phi_{n,\ell}(s_i) = \ \mathbb{1}\{\, i = \ell\},$$

where $i, \ell = 1, \dots n$, and the indicator function $\mathbb{1}\{\mathrm{A}\}$ takes the value one if A is true else zero. Thus, it is obvious that the observed function values are the initial scaling coefficients ($c_{n,\ell}$).

The signal g can be written using the lifting transform as

$$g(s) = \sum_{k=r+1}^{n} d_{j_k}\psi_{j_k}(s) + \sum_{k' \in S_r} c_{r,k'}\phi_{r,k'}(s),$$

where detail coefficient of point j_k , non-lifted points and wavelet functions are represented with d_{j_k}, r and ψ_{j_k}, respectively. In addition, non-lifted points are defined in $S_r \subset \{1, \dots, n\}\backslash\{j_k\}$, where $k = n, n-1, \dots, r+1$.

1.1.1 Forward lifting transform

LOCAAT algorithm selects a node from the tree; the difference between the selected node and its neighbors are computed. Then the node is removed from the tree, and its neighbors are updated, and the algorithm is repeated until r nodes left in the tree.

Scaling function at i^{th} node in stage k is defined as one ($k = n, n-1, \dots, r+1$ and $i = 1,2, \dots n$), and the number of non-lifted points, r, is set to 2 that is the suggested choice in early works.

LOCAAT algorithm is an iterative algorithm; it starts with stage $k = n$. The initial integrated function, $I_{k,i}$, is defined as the sum of the branch lengths between node i and its neighbors. The first node to be lifted (j_k) is the one having the smallest $I_{k,i}$, and neighbors of the lifted node j_k ($J_{k,i}$) are set as the fist degree neighbors. Observed function values are the initial scaling coefficients, $c_{k,i}$.

The predicted function value of the lifted node is defined as sum of its neighbors' weighted function values ($y_{k,j_k} = \sum_{i \in J_k} a_i^k c_{k,i}$). The weights ($a_i^k$) are the inverse distances which make the sum of the weights to 1 (Jansen et al., 2009).

The first stage of the lifting algorithm is the choice of the node-to-be-lifted, then detail coefficient of the lifted node j_k is computed. Details are the differences between the observed and the predicted function values ($d_{j_k} = c_{k,j_k} - y_{k,j_k}$). Later, initial integrated values and scaling coefficients for neighbors are updated using $I_{k-1,i} = I_{k,i} + a_i^k I_{k,j_k}$, $i \in J_k$, and $c_{k-1,i} = c_{k,i} + b_i^k d_{j_k}$, $i \in J_k$, where $b_i^k = {I_{k,j_k} I_{k-1,i}} \Big/ {\sum_{\ell \in J_k} I_{k-1,\ell}^2}$. Then the lifted node is removed from the tree, and the neighborhood structure of the node j_k is updated using the minimum spanning tree algorithm. After that the process is repeated for stage $k = n-1, n-2, \dots r+1$.

1.1.2 Denoising process

After applying lifting transformation to noise corrupted data, noisy detail coefficients are obtained. In the literature, one of the wavelet shrinkage techniques is applied to denoise detail coefficients. Then backward transformation is applied to get denoised function values.

Wavelet shrinkage methods assume that big detail coefficients include some noise next to real signal value, and small detail coefficients include only noise. It is assumed that noises are independently normally distributed with zero mean and the same σ^2 variance. If any detail coefficient is less than a threshold, it is set to zero. One choice of the threshold is the universal threshold (Donoho and Johnstone, 1995) defined as

$$\lambda = \sigma\sqrt{2\log n},$$

and the σ is estimated using the mean absolute deviation from the median (MEAD):

$$\text{MEAD}(d_k)=\text{mean}([|d_k - \text{medyan}(d_k)|]),$$

where d_k are the detail coefficients at k^{th} resolution level.

1.1.3 Artificial resolution levels

In the discrete wavelet transformation, detail coefficients are naturally grouped at resolution levels. Half of the coefficients are at the finest resolution level, and the half of the rest of the coefficients are at the second resolution level, and so on. This is not applicable for the lifting algorithm, but resolution levels can be artificially created.

Observations are located on leaves at binary trees/dendrograms. The closest two objects are merged via branches to generate a node on a tree. In each branching, at most two nodes/leaves are merged, and final two nodes unite to generate the root of the tree. In this research, the root is assigned to the first resolution level, then the nodes/leaves in the first branch are assigned to the second resolution level. Until all leaves are assigned to a resolution level, the process is repeated.

1.1.4 Backward lifting transform

The forward transform is followed by the denoising stage by the universal threshold, then the transform should be reversed to obtain the denoised estimate of function values. $\hat{g}$ of the function g. For $k = r+1, r+2, \dots, n$, first scaling coefficients of i^{th} neighbor at stage k are updated by

$$c_{k,i} = c_{k-1,i} - b_i^k d_{j_k},$$

then the scaling coefficients of the lifted node at stage k is predicted by

$$c_{k,j_k} = d_{j_k} + \sum_{i\in J_k} a_i^k c_{k,i}.$$

1.2 Anomaly detection and classification stage

The proposed anomaly detection algorithm is for hierarchically built trees with Ward's linkage and with Euclidean distances. Ward's linkage merges clusters which make minimum increase in the sum of squared errors. Ward's linkage is consistent with the proposed node value by Bozkus and Barber (2023).

1.2.1 One possible choice of a function value

To be able to apply the LOCAAT algorithm on dendrograms, neighborhood structure, branch lengths between nodes and a function value for each node are needed. Neighborhood structure and branch lengths between nodes are naturally achieved in the hierarchical clustering stage. A function value for each node is left to be able to apply the LOCAAT algorithm on dendrograms. Some of data sets may come with meaningful function values, but it is not always the case. Thus, a more general function value is needed which is applicable for all data sets. Bozkus and Barber (2023) proposed that one possible choice of a function value is compactness score which is defined as the average distance from the medoid of each cluster. Medoid of a cluster is set to the L_1-median proposed by Vardi and Zhang (2000).

1.2.2 Anomaly detection

After defining function value for each node, LOCAAT algorithm can be applied to dendrograms. Compactness scores are assigned to nodes on the tree, and detail coefficients are obtained via LOCAAT algorithm. First n detail coefficients belong to leaves; the next $n-2$ ones are for internal nodes, and the final detail coefficient is for the root.

In **Section 1.1.1**, predicted function values and detail coefficients are defined. If the detail coefficient of a node is negative, it means it is differentiated from its parent node; in other saying, the cluster at this node moves away from the cluster at its sibling node. If the detail coefficient of a node negatively increases, it has different features from other nodes that are assigned to the same cluster with it. To overcome with this issue, a threshold (α) can be defined. Nodes having smaller detail coefficients than α are labelled as nodes including anomaly points. Thus, the threshold α is a lower boundary for detail coefficients of nodes. One possible choice of α is

$$\alpha = Q_1 - |1{,}5 \times (Q_3 - Q_1)|,$$

where Q_1 and Q_3 are the first and third quantile of detail coefficients, respectively, and $|\cdot|$ is for the absolute value. To be able to assign a cluster to a node, it needs to have at least three offspring. If any node having the detail coefficient less than α includes at most two offspring, leaves under the node are defined as anomalies. Then, the points determined as anomalies are removed from the data set, and the tree is reconstructed with hierarchical clustering. The process is repeated until the number of anomalies found is zero.

1.2.3 Detection of clusters

After anomalies are removed from the data, detail coefficients are assigned to artificial levels. Then, detail coefficients are denoised using the procedure defined in Section 1.1.2. If the denoised detail coefficient of the root is greater than zero, there is a significant divergence in the data set. Thus, there are possible clusters in the data set. All nodes on the tree are checked. If a node with all its sub-nodes have denoised detail coefficients less than or equal to zero, a cluster is assigned to the node.

LOCAAT algorithm is not applied to the final r nodes, so a special rule needs to be defined for non-lifted nodes (detail coefficients for non-lifted points are not computed, but their function values are updated because of their neighbors, and they are always positive definite). If denoised detail coefficients of all sub-clusters of a non-lifted node are less than or equal to zero, the ratio of branch lengths should be checked (since prediction and update weights are computed using branch lengths). For each non-lifted node, the following ratio is computed:

$$\frac{(\text{sum of branch lengths from their child nodes})}{(\text{sum of branch lengths from their first degree neighbors})}.$$

If the defined ratio of a non-lifted node is greater than $2/3$, its sub-clusters are considered far enough from each other, so the clusters are assigned to child nodes of the non-lifted node. Otherwise, a cluster is assigned to the non-lifted node. Then the backward transformation is applied to obtain denoised function values. Anomaly detection and classification by the LOCAAT algorithm is called as "LiftOut".

2 Results

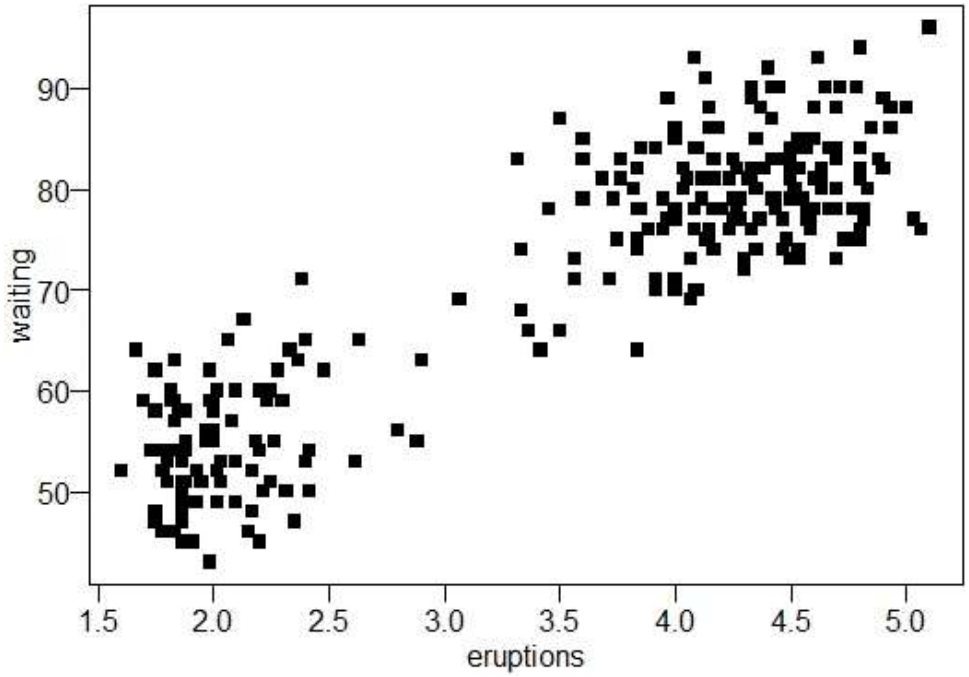
In this section, the performance of the LiftOut and DBSCAN are compared using a real-world example (due to page limit, only the result of real-world data is presented). Hierarchical clustering with Ward's linkage is applied on Euclidean distances. The MinPts parameter of DBSCAN is set to the recommended choice, 5. The proper choice of ϵ parameter is found as $1{,}5$ by checking the k-nearest-neighborhood plot as suggested by Hahsler et al. (2019). DBSCAN is available in R in dbscan package (Hahsler et al., 2019). The k-nearest-neighborhood plot is also obtained using kNNdist() function in dbscan library.

Old Faithful geyser data set

Azzalini and Bowman (1990) released a data set which was collected from the Old Faithful geyser in Yellowstone national park in Wyomind, USA. In this data set, there are two measurements: the waiting time between two consequent successful eruptions (labelled as

waiting) and duration of eruption (labelled as eruptions). The data set includes the measurements on 272 eruptions occurred in 1-15 August in 1985. The faithful data is available in R. Azzalini and Bowman (1990) and some other researchers who studied this data set suggest that there are two classes in faithful data. The scatter plot of faithful data is given in Fig. 1.

Fig. 1: Scatter plot of Old Faithful geyser data

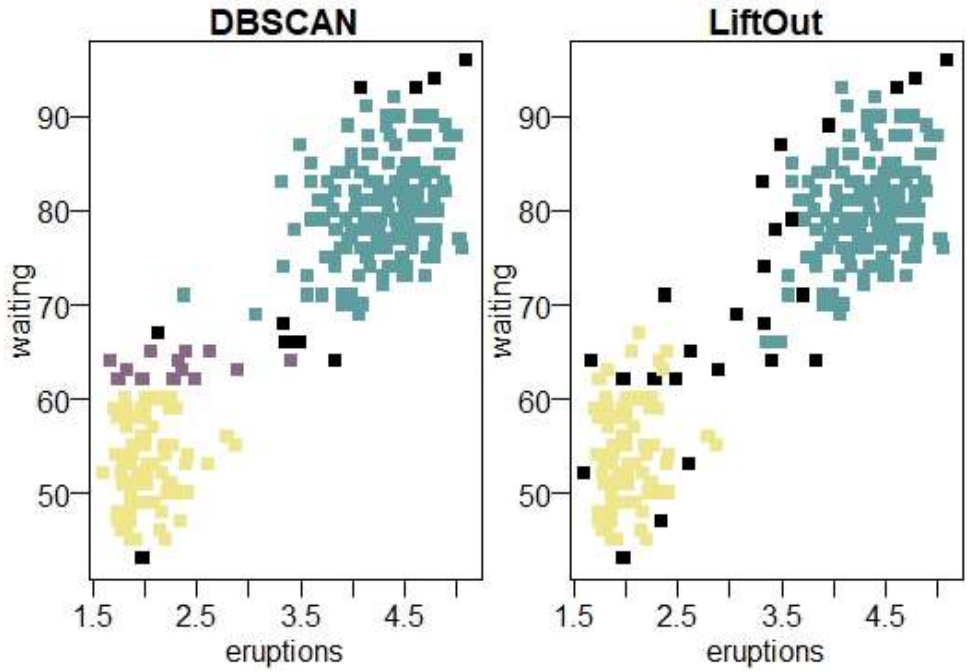


Source: Author

There are two obvious classes in the plot, but groups are not well separated and the dispersion of groups are high. Thus, detection of classes is not an easy task for the available classification techniques in the literature. After removing anomalies, classes may be differentiated easier by the classification techniques.

LiftOut and DBSCAN are applied to the Old Faithful geyser data, and both algorithms detect some objects as anomalies. Different classes and anomalies detected by LiftOut and DBSCAN are colored with different colors (Fig. 2). Anomalies are colored as black points. DBSCAN algorithm divides the data into three groups with some anomalies. DBSCAN creates a third group which includes the intersection points of two main groups. LiftOut, however, finds two main groups and some anomalies. The number of anomalies found in LiftOut (25 points) is higher than DBSCAN (10 points).

Fig. 2: Classification scheme found by LiftOut and DBSCAN algorithm



Source: Author

Conclusion

In this study, an alternative anomaly detection method is presented on denoised hierarchically built trees, and the classification scheme is found after removing anomalies. The proposed algorithm is called as LiftOut. A real-world example is presented to test the performance of the algorithm, and its performance is compared with the DBSCAN algorithm. DBSCAN algorithm divides the data set into more groups than the LiftOut. It forms another group from some of the data points labelled as anomalies by the LiftOut algorithm. LiftOut assigns observations into two groups (suggested number of groups in early researches) after removing anomalies. We should remember that DBSCAN algorithm needs user defined parameters, so the performance of the algorithm varies with different choice of parameters. The wrong choice of parameters decreases the performance of the DBSCAN, so the proper choice of parameters needs to be found before applying the algorithm itself. Both the anomaly detection and the classification part of the LiftOut algorithm works in high performances. However, LiftOut tempts to over clean the data.

References

Azzalini, A., & Bowman, A.W. (1990). A look at some data on the old faithful geyser.

Journal of the Royal Statistical Society Series C (Applied Statistics), 39(3), 357–365.

Barnett, V., & Lewis, T. (1984). *Outliers in statistical data (2nd ed.).* John Wiley & Sons.

Bozkus, N., & Barber, S. (2023). Automatic detection of the number of clusters by lifting. Submitted.

Breunig, M., Kriegel, H.P., Ng, R.T., & Sander, J. (2000). LOF: Identifying density-based local outliers. *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, 93–104.

Daubechies, I. (1992). *Ten lectures on wavelets*. Society for Industrial and Applied Mathematics, Philadelphia.

Donoho, D.L., & Johnstone, I.M. (1995). Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90(432), 1200–1224.

Ester, M., Kriegel, H.P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 226–231.

Hahsler, M., Piekenbrock, M., & Doran, D. (2019). dbscan: Fast density-based clustering with R. *Journal of Statistical Software*, 91(1), 1–30.

Han, J., Pei, J. & Kamber, M. (2012). *Data mining: concepts and techniques (3rd ed.).* Morgan Kaufmann.

Hawkins, D.M. (1980). *Identification of outliers*. Chapman and Hall.

Jansen, M., Nason, G.P., & Silverman, B.W. (2009). Multiscale methods for data on graphs and irregular multidimensional situations. *Journal of the Royal Statistical Society Series B (Statistical Methodology)*, 71(1), 97–125.

Kaufmann, L., & Rousseeuw, P. (1987). Clustering by means of medoids. *Data Analysis based on the L1-Norm and Related Methods*. 405–416.

Knorr, E.M., & Ng, R.T. (1998). Algorithms for mining distance-based outliers in large datasets. *Proceedings of the 24rd International Conference on Very Large Data Bases*, 392–403.

Vardi, Y., & Zhang, C.H. (2000). The multivariate L1-median and associated data depth. *Proceedings of the National Academy of Sciences*, 97(4), 1423–1426.

Contact

Nebahat Bozkus

Giresun University, Department of Statistics, Giresun 28200, Turkey.

nebahat.bozkus@giresun.edu.tr