Leave-one-out kernel density estimates for outlier detection

Sevvandi Kandanaarachchi, Rob J. Hyndman

Leave-one-out kernel density estimates for outlier detection

Sevvandi Kandanaarachchi
RMIT University
Email: sevvandi.kandanaarachchi@rmit.edu.au
Corresponding author

Rob J. Hyndman
Monash University
Email: rob.hyndman@monash.edu

20 January 2021

Leave-one-out kernel density estimates for outlier detection

Abstract

This paper introduces *lookout*, a new approach to detect outliers using leave-one-out kernel density estimates and extreme value theory. Outlier detection methods that use kernel density estimates generally employ a user defined parameter to determine the bandwidth. Lookout uses persistent homology to construct a bandwidth suitable for outlier detection without any user input. We demonstrate the effectiveness of lookout on an extensive data repository by comparing its performance with other outlier detection methods based on extreme value theory. Furthermore, we introduce *outlier persistence*, a useful concept that explores the birth and the cessation of outliers with changing bandwidth and significance levels. The R package `lookout` implements this algorithm.

Keywords: anomaly detection, topological data analysis, persistent homology, extreme value theory, peak over thresholds, generalized Pareto distribution

1 Introduction

Outliers, anomalies and novelties are often interchangeably used to describe the same concept: data points that are unusual compared to the rest. The existence of multiple words to describe similar concepts arise from the growth of outlier detection and applications in multiple research areas. Indeed, outlier detection is used in diverse applications ranging from detecting security breaches in the Internet of Things networks to identifying extreme weather events. Consequently, it is important to develop robust techniques to detect outliers, which minimize costly false positives and dangerous false negatives.

This diverse literature can be divided into approaches that use probability densities to define outliers, and those that use distances to define outliers. Outlier detection methods that use probability densities treat outliers as observations that are very unlikely given the other observations. Outlier detection methods that use distances treat outliers as observations that lie far from other observations.

In this paper, we take a probability density approach to outlier detection. We propose a new outlier detection algorithm that we call *lookout*, which uses leave-one-out kernel density estimates to identify the most unlikely observations. We address the challenge of bandwidth selection by using persistent homology — a concept in topological data analysis — and use extreme value theory (EVT) to identify outliers based on their leave-one-out density estimates.

The main challenge in using kernel density estimates for outlier detection is the selection of the bandwidth. Schubert, Zimek & Kriegel (2014) employ kernel density estimates to detect outliers using k-nearest neighbor distances where k is a user-specified parameter, which determines bandwidth. Qin et al. (2019) employ kernel density estimates to detect outliers in streaming data. They too have a radius parameter, which is equivalent to the bandwidth that needs to be specified by the user. Tang & He (2017) use reverse and shared k nearest neighbors to compute kernel density estimates and identify outliers. They also have a user defined parameter k that denotes the reverse k nearest neighbors. The oddstream algorithm (Talagala et al. 2020) computes kernel density estimates on a 2-dimensional projection defined by the first two principal components, and so bandwidths need to be selected. We avoid subjective user-choice, and the inappropriate use of bandwidths optimized for some other purpose, by proposing the use of persistent homology as a new tool for bandwidth selection.

Extreme Value Theory has been gaining popularity in outlier detection because of its rich, theoretical foundations. Burridge & Taylor (2006) used EVT to detect outliers in time series data. Clifton et al. (2014) used Generalized Pareto Distributions to model the tails in high-dimensional data and detect outliers. Other recent advances in outlier detection that use EVT include the stray (Talagala, Hyndman & Smith-Miles 2021), oddstream (Talagala et al. 2020) and HDoutliers (Wilkinson 2017) algorithms. Of these three methods, stray is an enhancement of HDoutliers and both use distances to detect outliers, while oddstream uses kernel density estimates to detect outliers in time series data. Our approach is closest to oddstream in that we also apply EVT to functions of kernel density estimates. However, we use a different functional, and we avoid the need for an outlier-free training set.

A brief introduction to persistent homology and EVT is given in Section 2. In Section 3 we introduce the algorithm *lookout* and the concept of outlier persistence, which explores the birth and death of outliers with changing bandwidth. We show examples illustrating the usefulness of outlier persistence and conduct experiments using synthetic data to evaluate the performance of lookout in Section 4. Using an extensive data repository of real datasets, we compare the performance of lookout to HDoutliers and stray in Section 5.

We have produced an R package `lookout` (Kandanaarachchi & Hyndman 2021) containing this algorithm. In addition, all examples in this paper are available in the supplementary material at https://github.com/sevvandi/supplementary_material/tree/master/lookout.

We have used the R packages `TDAstats` (Wadhwa et al. 2018) and `ggtda` (Brunson, Wadhwa & Scott 2020) for all TDA and persistent homology computations and related graphs. We have used the R package `evd` (Stephenson 2002) to fit the Generalized Pareto Distribution.

2 Mathematical background

In this Section we briefly go through three different topics:

1. topological data analysis and persistent homology;
2. extreme value theory and the peaks-over-threshold approach; and
3. kernel density estimation.

2.1 Topological data analysis and persistent homology

Topological data analysis is the study of data using topological constructs. It is about inferring high dimensional structure from low dimensional representations such as points and assembling discrete points to construct global structures (Ghrist 2008). Persistent homology is a method in algebraic topology that computes topological features of a space that persist across multiple scales or spatial resolutions. These features include connected components, topological circles and trapped volumes. Features that persist for a wider range of spatial resolutions represent robust, intrinsic features of the data while features that sporadically change are perturbations resulting from noise. Persistent homology has been used in a wide variety of applications including biology (Topaz, Ziegelmeier & Halverson 2015), computer graphics (Carlsson et al. 2008) and engineering (Perea & Harer 2015). In this Section we will give a brief overview of persistent homology without delving into the mathematical details. Readers are referred to Ghrist (2008) and Carlsson (2009) for an overview and Wasserman (2018) for a statistical viewpoint on the subject.

Simplicial complex

Consider a data cloud representing a collection of points. This set of points is used to construct a graph where the points are considered vertices and the edges are determined by the distance between the points. Given a proximity parameter ε, two vertices are connected by an edge if the distance between these two points is less than or equal to ε. Starting from this graph, a simplicial complex — a space built from simple pieces — is constructed. A simplicial complex is a finite set

of k-simplices, where k denotes the dimension. To give some examples, a point is a 0-simplex, an edge a 1-simplex, a triangle a 2-simplex, and a tetrahedron a 3-simplex. Suppose S denotes a simplicial complex that includes a k-simplex γ. Then all non-empty subsets of $\beta \subset \gamma$ are also included in S. For example, if S contains a triangle pqr, then the edges pq, qr and rs, and the vertices p, q and r, are also in S.

The *Vietoris-Rips* complex and the *Cech* complex are two types of k-simplicial complexes. We will construct a Vietoris-Rips complex from the data cloud as it is more computationally efficient than the Cech complex (Ghrist 2008). Given a set of points and a proximity parameter $\varepsilon > 0$, $k + 1$ points within a distance of ε to each other form a k-simplex. For example, consider 5 points p, q, r, s and t and suppose the distance between any two points except t is less than ε. Then we can construct the edges pq, pr, ps, qr, qs and rs. From the edges pq, qr and rp we can construct the triangle pqr, from pq, qs and sp the triangle pqs and so on, because the distance between any two points p, q, r and s is bounded by ε. By constructing the 4 triangles pqr, qrs, rsp and spq we can construct the tetrahedron $pqrs$. The vertex t is not connected to this 3-simplex because the distance between t and the other vertices is greater than ε. The simplicial complex resulting from these 5 points consists of the tetrahedron $pqrs$ and all the subset k-simplices and the vertex t. Figure 1 shows this simplicial complex on the left and another example on the right.
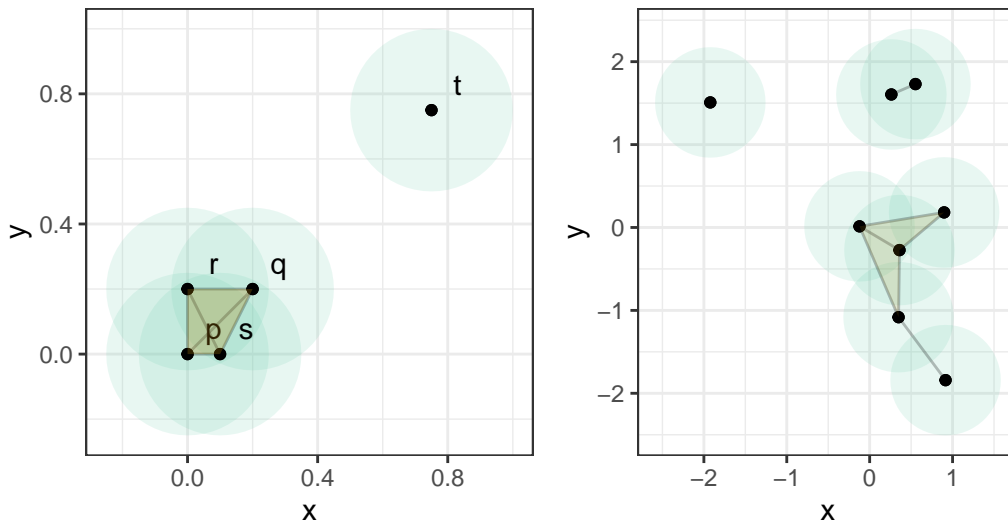

Figure 1: *The figure on the left shows the points p, q, r, s and t with a proximity parameter $\varepsilon = 0.5$ and the resulting Rips complex consisting of the tetrahedron pqrs, triangles pqr, qrs, rsp, pqs, edges pq, qr, rs, sp, qs, pr and vertices p, q, r, s and t. The figure on the right shows 8 points and the resulting Rips complex with $\varepsilon = 4/3$.*

Persistent homology

Given a point cloud of data, the resulting Rips complex depends on the value of the proximity parameter ε. As we increase ε, topological features such as connected components and holes
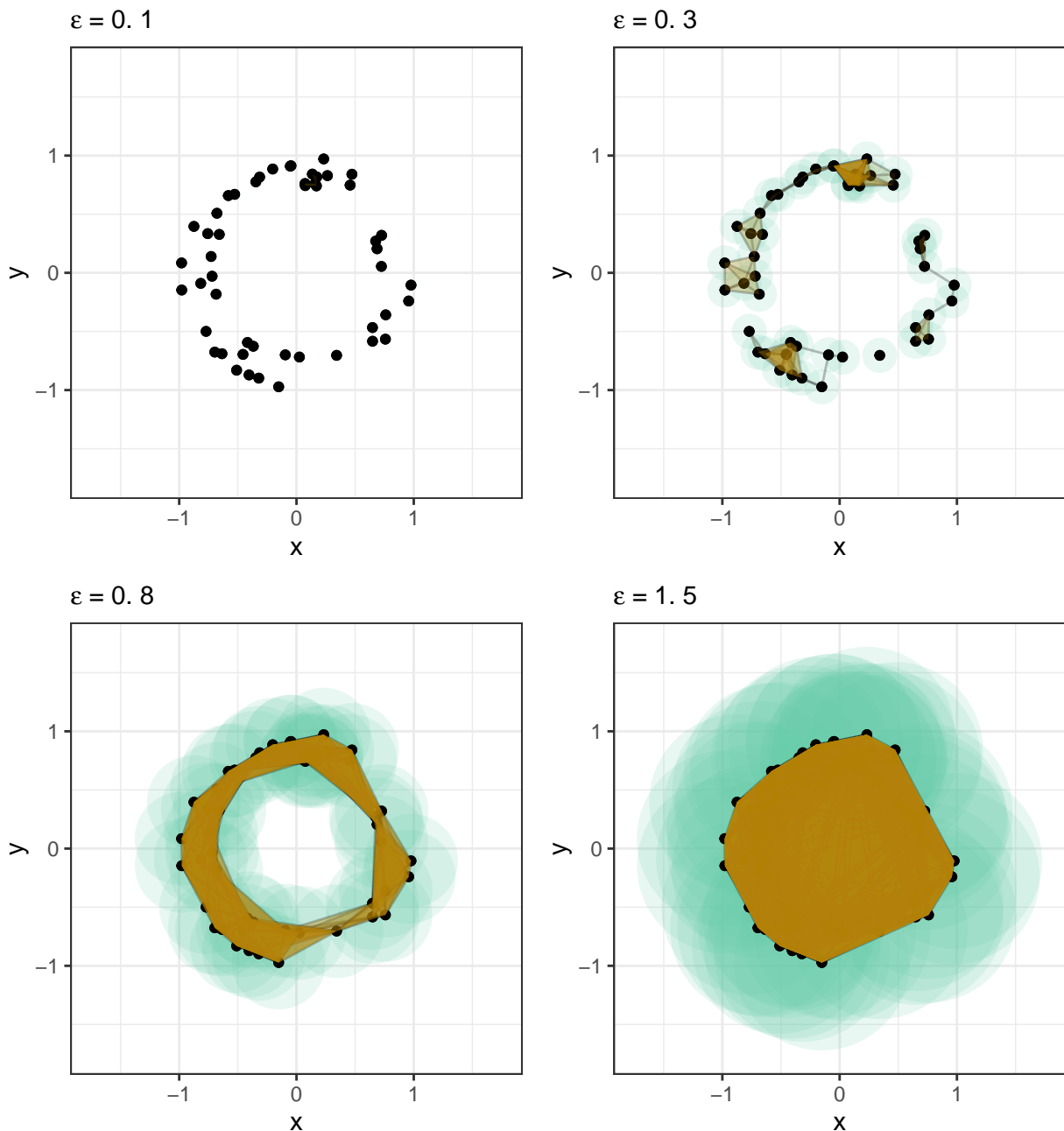
Figure 2: *Rips complexes resulting from different ε values.*

appear and disappear. This is the focus of persistent homology. For example, in Figure 2, we start with a large number of connected components (top-left) and as ε increases to 0.8 the number of connected components merge and decrease to 1 (bottom-left). Around this ε value a hole appears and as ε increases to 1.5, it too disappears (bottom-right). The appearances and disappearances of these topological features are referred to as births and deaths and are illustrated using a *barcode* or a *persistence diagram*.

Figure 3 shows the barcode and the persistence diagram of the point cloud shown in Figure 2. The barcode consists of a set of horizontal line segments, each line segment denoting a feature

that starts at its birth diameter and ends at its death diameter. These line segments are grouped by their dimension. The orange lines in Figure 3 denote the zero dimensional holes, which are connected components and blue lines denote one dimensional holes. The longer blue line which is born at 0.53 and dies at 1.35 corresponds to the hole at the center of the point cloud in Figure 2. Such features that continue for a large range of ε, represent structural properties of the data that are of interest to us. The same information is presented differently in the persistence diagram, where the birth and the death of each feature is denoted by a point. Points away from the diagonal inform about the structure of the data while points closer to the diagonal are perturbations related to noise.

Betti numbers denoted by β_k give the number of k-dimensional holes of a simplicial complex. The number of connected components is given by β_0 and the number of one-dimensional holes is given by β_1. The number of two-dimensional holes, which are trapped volumes or voids is given by β_2. The Betti numbers can also be computed from the barcode by drawing a vertical line at the desired diameter and counting the points of intersection for each dimension. For example, in Figure 3 the Betti numbers corresponding to the simplicial complex at diameter 0.5 are $\beta_0 = 1$ and $\beta_1 = 0$.
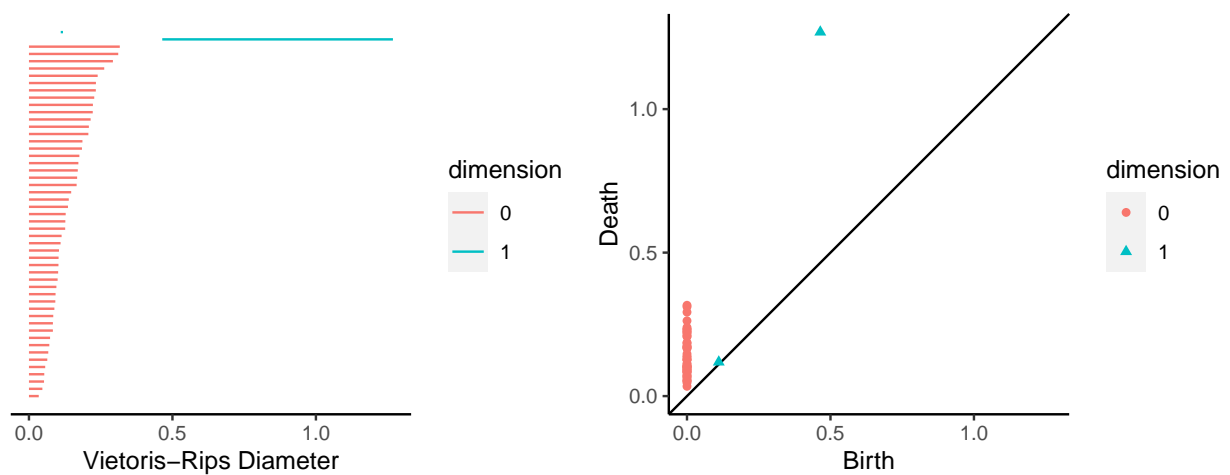

Figure 3: *The graph on the left shows the barcode of the point cloud in Figure 2 and the graph on the right shows the persistence diagram.*

The question arises: which ε is most representative of the structure of the data cloud?

2.2 Extreme value theory

Extreme Value Analysis (EVA) is used to model rare, extremal events, which often have a big societal impact such as a 100-year flood. EVA is used in many industries including hydrology, finance and insurance (R.D. Reiss & M.Thomas 2001). For example, setting premiums so that massive insurance losses are mitigated is just as important, if not more than achieving profits

on average. Due to its applicability, Extreme Value Theory (EVT) has also been used in outlier detection (Wilkinson 2017; Talagala et al. 2020). In this section we will give a brief introduction to EVA using the notation in Coles (2001).

Consider n independent and identically distributed random variables $X_1, \ldots, X_n$ with a distribution function $F(x) = P\{X \le x\}$. Then the maximum of these n random variables is

$$M_n = \max\{X_1, \ldots, X_n\}.$$

If F is known, the distribution of M_n can be derived for any value of n as

$$
\begin{aligned}
P\{M_n \le z\} &= P\{X_1 \le z, \ldots, X_n \le z\}, \\
&= P\{X_1 \le z\} \times \cdots \times P\{X_n \le z\}, \\
&= (F(z))^n.
\end{aligned}
$$

However, F is not known in practice. This gap is filled by Extreme Value Theory, which studies approximate families of models for F^n so that extremes can be modeled and uncertainty quantified. Similar to the Central Limit Theorem for means, the Fisher-Tippet-Gnedenko Theorem, also known as the the extreme value theorem or the extremal types theorem states that under certain conditions, a scaled maximum $\frac{M_n - a_n}{b_n}$, if a_n and b_n exist, have certain limit distributions.

Theorem 2.1 (Fisher-Tippett-Gnedenko). *If there exist sequences such that*

$$P\left\{\frac{(M_n - a_n)}{b_n} \le z\right\} \to G(z) \quad as \quad n \to \infty,$$

where G is a non-degenerate distribution function, then G belongs to one of the following families:

$$I : G(z) = \exp\left(-\exp\left[-\left(\frac{z-b}{a}\right)\right]\right), \quad -\infty < z < \infty,$$

$$II : G(z) = \begin{cases} 0, & z \le b, \\ \exp\left(-\left(\frac{z-b}{a}\right)^{-\alpha}\right), & z > b, \end{cases}$$

$$III : G(z) = \begin{cases} \exp\left(-\left(-\left[\frac{z-b}{a}\right]\right)^{\alpha}\right), & z < b, \\ 1, & z \ge b, \end{cases}$$

for parameters a, b and α where $a, \alpha > 0$.

The distribution types I, II and III are known as Gumbel, Fréchet and Weibull families respectively. These three families of distributions can be further combined into a single family by using the following distribution function known as the Generalized Extreme Value (GEV) distribution,

$$G(z) = \exp\left\{-\left[1 + \xi\left(\frac{z-\mu}{\sigma}\right)\right]^{-1/\xi}\right\},$$

where the domain of the function is $\{z : 1 + \xi(z-\mu)/\sigma > 0\}$ and the parameters $\mu, \xi \in \mathbb{R}$ and $\sigma > 0$. The parameter μ is called the location parameter and σ is called the scale parameter. The parameter ξ is called the shape parameter and it determines the behavior of the tails and differentiates between the Gumbel, Fréchet and Weibull distributions. When $\xi = 0$ we obtain a Gumbel distribution with exponentially decaying tails. When $\xi < 0$ we get a Weibull distribution with a finite upper end and when $\xi > 0$ we get a Fréchet family of distributions with polynomially decaying tails.

Block maxima approach

The GEV distributions are used to study extremes using the block maxima method. That is, observations are divided into blocks of equal length and the set of block maxima are modeled by fitting a GEV distribution. For example, one might consider yearly blocks to model the maximum daily temperature. However, this is considered a wasteful approach with respect to the data, because many data points including extremes maybe discarded by only considering the block maximum. For example, a given year may have multiple days of extreme temperatures that are higher than the maximum temperature of another year. But the block maxima approach would discard these other extremal temperatures and only consider the maximum temperature of each year.

This problem is remedied by regarding the observations that exceed a high threshold as extremes. This approach is known as the Peaks Over Threshold (POT) approach and uses the Generalized Pareto Distribution (GPD).

The Generalized Pareto Distribution and the POT approach

As previously, consider n independent and identically distributed random variables $X_1, \ldots, X_n$ with a distribution function F. For a high threshold value u, if all $X_i > u$ are regarded as extremes, we can write the conditional probability of extreme events as

$$P\{X > u + y \mid X > u\} = \frac{1 - F(u+y)}{1 - F(u)}, \quad y > 0,$$

giving us

$$P\left\{X \leq u + y \mid X > u\right\} = \frac{F(u+y) - F(u)}{1 - F(u)}, \quad y > 0.$$

The distribution function

$$F_u(y) = P\left\{X \leq u + y \mid X > u\right\},$$

describes the *exceedances* above the threshold u. If F is known we could compute this probability. However, as F is not known in practice we use approximations based on the Generalized Pareto Distribution (Pickands 1975).

Theorem 2.2 (Pickands). *Let $X_1, X_2, \ldots, X_n$ be a sequence of independent random variables with a common distribution function F, and let*

$$M_n = \max\{X_1, \ldots, X_n\}.$$

Suppose F satisfies Theorem 2.1 so that for large n

$$P\{M_n \leq z\} \approx G(z),$$

where

$$G(z) = \exp\left\{-\left[1 + \xi\left(\frac{z - \mu}{\sigma}\right)\right]^{-1/\xi}\right\},$$

for some $\mu, \xi \in \mathbb{R}$ and $\sigma > 0$. Then for large enough u, the distribution function of $(X - u)$ conditional on $X > u$, is approximately

$$H(y) = 1 - \left(1 + \frac{\xi y}{\sigma_u}\right)^{-1/\xi}, \tag{1}$$

where the domain of H is $\{y : y > 0 \text{ and } (1 + \xi y)/\sigma_u > 0\}$, where $\sigma_u = \sigma + \xi(u - \mu)$.

The family of distributions defined by equation (1) is called the **Generalized Pareto Distribution** (GPD). We note that the GPD parameters are determined from the associated GEV parameters. In particular, the shape parameter ξ is the same in both distributions.

For a chosen threshold u, the parameters of the GPD can be estimated by standard maximum likelihood techniques. As an analytical solution that maximizes the likelihood does not exist, numerical techniques are used to arrive at an approximate solution. We use the R package `evd` to fit a GPD using the POT approach.

2.3 Kernel density estimation

In this Section we review some basics of kernel density estimation. Consider a univariate kernel density estimator for a sample $x_1, x_2, \ldots, x_n \in \mathbb{R}$ given by

$$\hat{f}(x; h) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) = \frac{1}{n} \sum_{i=1}^{n} K_h(x - x_i),$$

where K denotes the kernel and $K_h(t) = K(t/h)/h$. The Gaussian kernel is given by $K(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2)$, and the Epanechnikov kernel is given by $K(x) = \frac{3}{4}\left(1 - x^2\right)_+$. In order to match the variances of different kernels, we consider the centralized second order moment of K $\mu_2(K) = \int z^2 K(z)dz$, and scale K such that $\mu_2(\tilde{K}) = 1$. By doing a change of variables for the Epanechnikov kernel we obtain

$$\tilde{K}(x) = \frac{3}{4\sqrt{5}} \left(1 - \frac{x^2}{5}\right)_+,$$

giving $\mu_2(\tilde{K}) = 1$.

The leave-one-out kernel density estimator is given by

$$\hat{f}_{-j}(x; h) = \frac{1}{(n-1)h} \sum_{i \neq j} K\left(\frac{x - x_i}{h}\right),$$

where x_j is left out. We further simplify it and obtain

$$
\begin{aligned}
\hat{f}_{-j}(x_j; h) &= \frac{1}{(n-1)h} \sum_{i \neq j} K\left(\frac{x_j - x_i}{h}\right), \\
&= \frac{1}{(n-1)h} \left(\sum_{i=1}^{n} K\left(\frac{x_j - x_i}{h}\right) - K\left(\frac{x_j - x_j}{h}\right)\right), \\
&= \frac{1}{(n-1)h} \left(\sum_{i=1}^{n} K\left(\frac{x_j - x_i}{h}\right) - K(0)\right), \\
&= \frac{n}{n-1} \hat{f}(x_j; h) - \frac{1}{(n-1)h} K(0).
\end{aligned}
$$

As such, it is easy to compute the leave-one-out estimate at x_j by leaving out x_j using $\hat{f}(x_j; h)$.

Next, we consider the multivariate kernel estimator

$$\hat{f}(\boldsymbol{x}; \boldsymbol{H}) = \frac{1}{n|\boldsymbol{H}|^{1/2}} \sum_{i=1}^{n} K\left(\boldsymbol{H}^{-1/2}(\boldsymbol{x} - \boldsymbol{x}_i)\right) = \frac{1}{n} \sum_{i=1}^{n} K_{\boldsymbol{H}}(\boldsymbol{x} - \boldsymbol{x}_i),$$

where $\mathbf{x} \in \mathbb{R}^p$, $\mathbf{H}$ denotes a $p \times p$ positive definite bandwidth matrix and $K_{\mathbf{H}}(\mathbf{z}) = \mathbf{H}^{-1/2} K(\mathbf{H}^{-1/2}\mathbf{z})$. The multivariate Gaussian kernel is given by

$$K(\mathbf{x}) = \frac{1}{(2\pi)^{p/2}} \exp\left(-\frac{1}{2}\|\mathbf{x}\|^2\right),$$

and the multivariate scaled Epanechnikov kernel by

$$K(\mathbf{x}) = \frac{p+2}{c_p}\left(1 - \frac{\|\mathbf{x}\|^2}{5}\right)_+,$$

where c_p is the volume of the unit sphere in $\mathbb{R}^p$.

3 Methodology

3.1 Bandwidth selection using TDA

The main challenge in using kernel density estimates for outlier detection is selecting the appropriate bandwidth. There is a large body of literature on kernel density estimation and bandwidth selection (Scott 1994; Wang & Scott 2019) that focuses on computing density estimates that represent the data as accurately as possible, where measures of accuracy have certain asymptotic properties. The most common measure of performance that evaluates kernel density estimates is the mean integrated squared error (MISE). However, our goal is somewhat different as we are interested in finding outliers in the data and thus do not interest ourselves in finding a good representation for the rest of the data. The optimal bandwidth which minimizes the MISE is generally not suitable for outlier detection. Often this bandwidth is too small and causes the kernel density estimates of the boundary and near-boundary points to be too low and similar to those of the outliers. In high dimensions this problem is exacerbated due to the sparsity of the data. Thus, we need a bandwidth that assists outlier detection. A too small bandwidth causes everything to be outliers, while too large a bandwidth hides outliers with other points.

To select a bandwidth for outlier detection, we use the barcode discussed in Section 2.1. First we construct the barcode of the data cloud for dimension zero using the Vietoris-Rips diameter. From the barcode we obtain the sequence of death diameters for the connected components, i.e. the sequence of diameters at which connected components merge into one another and disappear. By construction this is an increasing sequence as seen in Figure 3. Let us denote this sequence by $\{d_i\}_{i=1}^N$, that is we consider the barcode to have N horizontal line segments for dimension 0. To understand this sequence better we consider an example and compare $\{d_i\}_{i=1}^N$ with k nearest neighbor distances for different k. Figure 4 shows an annulus with some

points in the center, its barcode, and the violin plot of different distances. The set $\{d_i\}_{i=1}^N$ is denoted as TDA in Figure 4 with KNN distances for $k \in \{1, 5, 10\}$ denoted by NN_1, NN_5 and NN_10 respectively. We see that the TDA Rips diameters have a non-zero intersection with KNN distances for each k. That is, we can think of the TDA distances as a subset of the union of different KNN distances. Consequently, we can use TDA distances without having to select the parameter k for KNN distances.
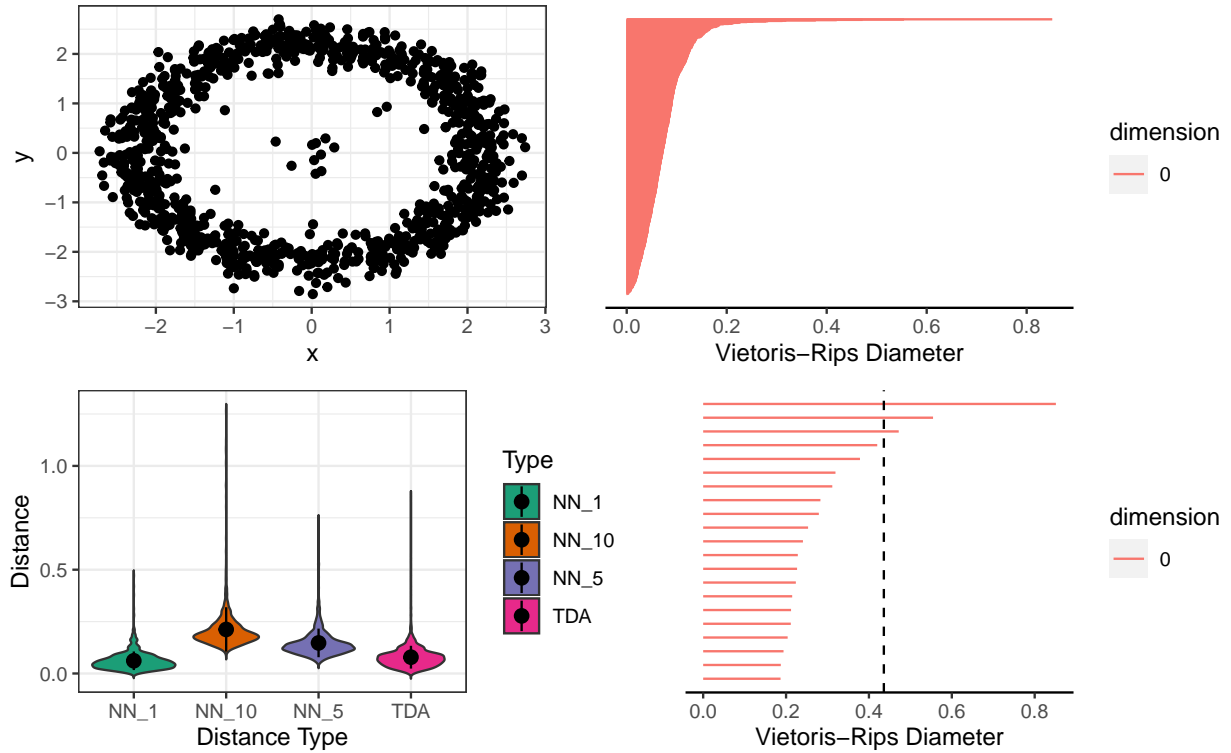

Figure 4: *An annulus with some points in the center on the top left. Its TDA barcode on the top right and the violin plot of TDA death diameters and KNN distances at the bottom.*

The plot on the bottom right in Figure 4 shows an enlarged version of the top part of the barcode, which contains the largest Rips diameters. A vertical dashed line is drawn at diameter 0.436, the second largest Rips diameter. As the diameter increases from 0.436 till 1.017, which is the maximum diameter, the number of connected components stay the same. For this point cloud, $(0.436, 1.017)$ is the largest diameter range for which the number of components stay the same. As such, it signifies a global structural property of the point cloud. We want to take this structure into account when selecting the bandwidth. Therefore, we choose a diameter that gives rise to persisting features, which in our case are connected components. We consider the Rips diameter intervals (d_i, d_{i+1}) for all i, and find the largest interval by computing successive differences

$$\Delta d_i = d_{i+1} - d_i, \quad \text{for} \quad i \in \{1, \cdots, N-1\}.$$

We choose the Rips diameter d_i that maximizes Δd_i, where

$$i_* = \arg\max\{\Delta d_i\}_{i=1}^{N-1},$$
$$d_* = d_{i_*}, \tag{2}$$

as our chosen Rips diameter. We use this diameter d_* to compute the bandwidth of the scaled Epanechnikov kernel.
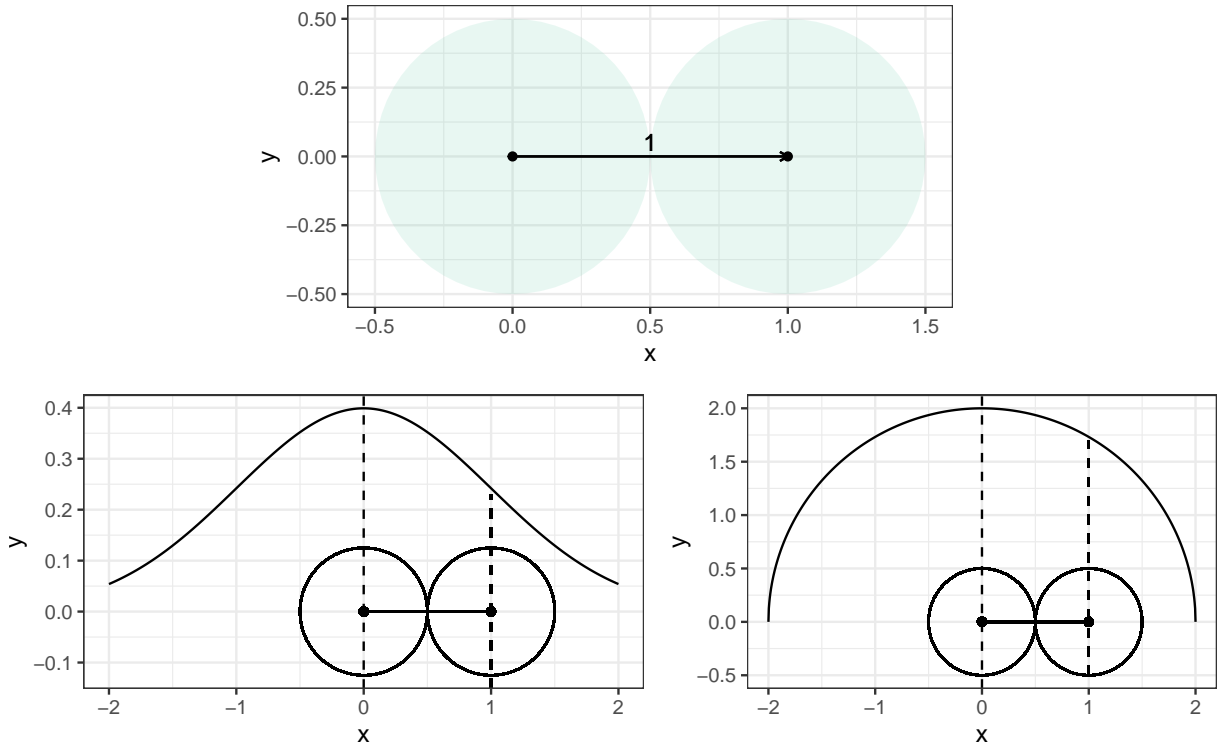

Figure 5: *The plot on the top shows two points that are $\varepsilon = 1$ distance away from each other. The plots at the bottom show a Gaussian and an Epanechnikov kernel, with bandwidth 1 and 2 respectively. The choice of bandwidth makes each point contribute to the other's kernel density estimate.*

Consider two points in the point cloud that are $\varepsilon = 1$ distance away as shown in Figure 5. Suppose we want these two points to contribute to each others kernel density estimates. This is illustrated in the bottom row of Figure 5. This can be achieved by using a range of bandwidth values. Let us consider a 1-dimensional dataset. For an Epanechnikov kernel, any bandwidth $h > \varepsilon$ would suffice. But, a larger bandwidth will make the contribution of the neighboring point higher than a smaller bandwidth. For a Gaussian kernel, there is no such restriction as any h would include the other point. However, a small $h << \varepsilon$ would make the contribution of the other point quite small. As such, we choose $h = \varepsilon$ for a Gaussian kernel where ε denotes a particular choice of the Rips diameter. For the Epanechnikov kernel, we choose $h = \sqrt{5}\varepsilon$, so that it matches the variance of the Gaussian kernel.

For higher dimensions, we use a bandwidth matrix of the form $H = hI$, and assign

$$h = d_*^{2/p}, \tag{3}$$

giving us

$$
\begin{aligned}
\|H^{-1/2}(x - x_i)\|^2 &= (x - x_i)^T \left(H^{-1/2}\right)^T H^{-1/2}(x - x_i), \\
&= (x - x_i)^T H^{-1}(x - x_i), \\
&= \frac{1}{h^p}(x - x_i)^T(x - x_i), \\
&= \frac{1}{d_*^2}\|x - x_i\|^2, \tag{4}
\end{aligned}
$$

where d_* is the Rips diameter given in equation (2). By choosing this bandwidth we make sure that points within a distance of d_* contribute to the kernel density estimate of x. This diameter results in the following leave-one-out kernel density estimate

$$
\begin{aligned}
\hat{f}_{-j}(x_j; H) &= \frac{n}{n-1}\hat{f}(x_j; H) - \frac{1}{(n-1)|H|^{1/2}}K(0), \\
&== \frac{n}{n-1}\hat{f}(x_j; H) - \frac{1}{(n-1)d_*}K(0). \tag{5}
\end{aligned}
$$

3.2 Algorithm *lookout*

Now we have all the building blocks necessary to describe the algorithm *lookout*. Consider an $N \times p$ data matrix X with N observations in $\mathbb{R}^p$. It is customary in outlier detection to scale the data so that all attributes contribute equally to outlier detection. Min-Max normalization, which scales each attribute to $[0, 1]$ has been shown effective compared to other normalization techniques (Kandanaarachchi et al. 2019a). As such, we normalize the data using Min-Max. To accommodate datasets that do not need to be normalized, we make normalization a user option.

To determine the bandwidth for kernel density estimation, we use the Rips diameter d_* as specified in equation (2) and use the bandwidth matrix $H = hI$ with $h = (d_*)^{2/p}$ as in equation (3). We use the scaled Epanechnikov kernel to compute kernel density estimates and leave-one-out kernel density estimates as in equation (5).

Let us denote the kernel density estimate of x_i by y_i and the leave-one-out kde of x_i (by leaving out x_i) by y_{-i}. As we want to model outliers as extremes, we use $-\log(y_i)$ and $-\log(y_{-i})$ so that smaller kde values are mapped to higher values and vice versa. Then we fit a Generalized Pareto Distribution to $-\log(y_i)$ using the POT approach discussed in Section 2.2. We use the

90th percentile as the threshold for the POT approach as recommended by Bommier (2014). Using the fitted GPD parameters, μ, σ and ξ we find the probability of the leave-one-out kde values $-\log(y_{-i})$, that is, we compute $P\left(-\log(y_{-i})|\mu,\sigma,\xi\right)$. Then, using a pre-defined cut-off probability α, we declare points with $P\left(-\log(y_{-i})|\mu,\sigma,\xi\right) < \alpha$ as outliers. We summarize these steps in Algorithm 1.

Algorithm 1: *lookout.*

---

input : The data matrix X, parameters α and *unitize*.

output: The outliers, the GPD probabilities of all points, GPD parameters and bandwidth

1 If *unitize = TRUE*, then normalize the data so that each column is scaled to $[0,1]$.
2 Construct the persistence homology barcode of the data.
3 Find d_* as in equation (2).
4 Using $h = (d_*)^{2/p}$ and $H = hI$ compute kernel density estimates and leave-one-out kernel density estimates using the scaled Epanechnikov kernel.
5 Denote the kde of x_i by y_i and leave-one-out kde by y_{-i}.
6 Using the POT approach fit a GPD to $\{-\log(y_i)\}_{i=1}^{N}$ and find μ, σ and ξ.
7 Using the GPD parameters μ, σ and ξ find the probability of the leave-one-out kde values $\{-\log(y_{-i})\}_{i=1}^{N}$, i.e., $P\left(-\log(y_{-i})|\mu,\sigma,\xi\right)$ for all i.
8 If $P\left(-\log(y_{-i})|\mu,\sigma,\xi\right) < \alpha$, then declare x_i as an outlier.

The output probability of lookout is the GPD probability of the points, and as such low probabilities indicate outliers and high probabilities indicate normal points. In addition, as we take $-\log(y_i)$ and $-\log(y_{-i})$ to fit a GPD, the scaling factor of the kernel $K(x)$ does not affect the GPD parameters as it is just an offset. For the scaled Epanechnikov kernel

$$\log(K(x)) = \log\left(\frac{p+2}{c_p}\right) + \log\left(\left(1 - \frac{\|x\|^2}{5}\right)_+\right).$$

That is, the logarithm values of all points are shifted by an equal amount because of the scaling factor in $K(x)$. It is similar for the Gaussian kernel as well. Thus, the scaling factor of the kernel can be disregarded for the algorithm lookout.

The algorithm lookout has only 2 parameters, α and *unitize*. The parameter α determines the threshold for outlier detection and the parameter *unitize* gives the user an option to normalize the data. We set $\alpha = 0.05$ and *unitize* = TRUE as default parameter values. We use the Epanechnikov kernel in lookout due to ease of computation. However, any kernel can be incorporated as long as the variances are matched.

3.3 Outlier persistence

Lookout identifies outliers by selecting an appropriate bandwidth using TDA. If the bandwidth is changed, will the original outliers be still identified as outliers by lookout? We explore this question by varying bandwidth values in lookout. Similar work is discussed by Minnotte & Scott (1993), who introduce the Mode Tree, which tracks the modes of the kernel density estimates with changing bandwidth values. Another such work is the SiZer map (Chaudhuri & Marron 1999), a graphical device that studies features of curves for varying bandwidth values.
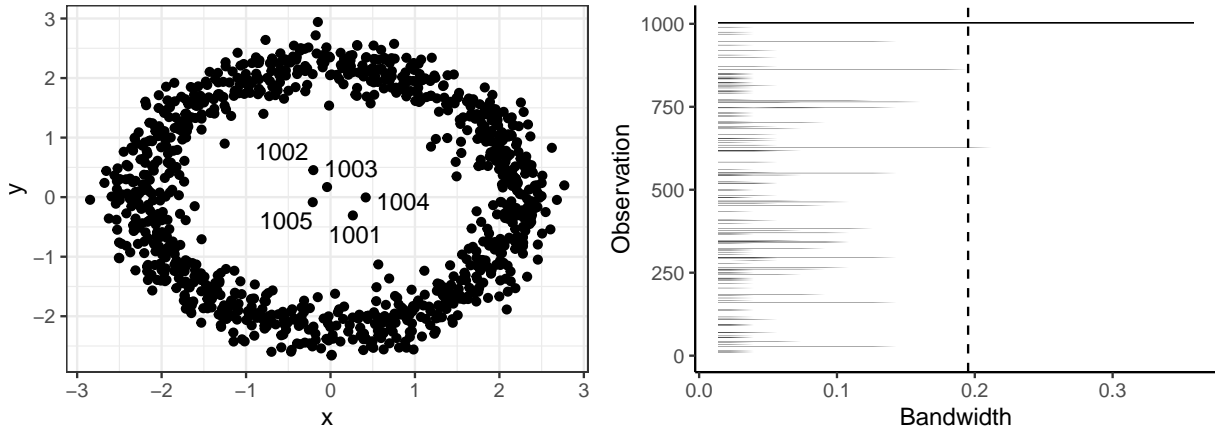

Figure 6: *Outliers at the center of the annulus in the left plot showing the outlier labels 1001-1005. The outlier persistence diagram on the right with the y-axis denoting the labels. The dashed line shows the lookout bandwidth d_*.*

If a point is identified as an outlier by the algorithm lookout for a range of bandwidth values, then it increases the validity of that point as an outlier. Consider an annulus with some points in the middle as shown in the left plot of Figure 6. The plot on the right, which is similar to a barcode, shows the outliers identified by lookout for different bandwidth values. Each horizontal line segment shows the range of Rips diameter values that has identified each point as an outlier. With some abuse of notation, we label the x axis "bandwidth", even though it actually represents the Rips diameter $d_* = h^{p/2}$, where the bandwidth matrix $\boldsymbol{H} = h\boldsymbol{I}$. This is motivated from equation (4), as we only consider points $\boldsymbol{x}_i$ within a distance d_* of every point $\boldsymbol{x}$ when computing kernel density estimates using the Epanechnikov kernel. In this plot, the y-axis corresponds to the point index. We call this plot *the outlier persistence diagram* signifying the link to topological data analysis.

In the example in Figure 6, we see that the points 1001-1005 are identified as outliers in the outlier persistence diagram for a large range of bandwidth values. The Rips diameter d_* selected by lookout is shown by a vertical dashed line. Many points are identified as outliers for small bandwidth values but do not continue to be outliers for long. These outliers are born at small

bandwidth values and after being an outlier for a relatively short period, die at small bandwidth values. Some points are never identified as outliers, even at small bandwidths.

By computing leave-one-out kernel density estimates using the bandwidth d_*, lookout fits a GPD to identify outliers. The outlier persistence diagram is a tool to observe the persistence of outliers with changing bandwidth values. As such, we only change the bandwidth values in this exercise and fix the GPD parameters. We use lookout to select these GPD parameters by using d_* as in equation (2). Furthermore, we use the death diameter sequence d_i to construct the set of bandwidth values used in this exercise. We use ℓ bandwidth values starting from the βth percentile of sequence $\{d_i\}$ ending at $\lambda \times \max_i d_i$. The parameters ℓ, β and λ are user-defined with default values $\ell = 20$, $\beta = 90$ and $\lambda = \sqrt{5}$. Increasing ℓ gives better granularity but increases the amount of computations. As the death diameters are tightly packed, the default value of 90th percentile gives a small enough starting bandwidth and $\sqrt{5} \max_i d_i$ gives a large ending bandwidth. We summarize these steps in Algorithm 2.

Algorithm 2: *outlier persistence for fixed α.*

input : The data matrix X, parameters α, *unitize* and bandwidth range parameters ℓ, β and λ

output : An $N \times \ell$ binary matrix Y where N denotes the number of observations and ℓ denotes the number bandwidth values with $y_{ik} = 1$ if the ith observation is identified as an outlier for bandwidth index k.

1 Initialize matrix Y to zero.

2 Run Algorithm 1 to determine the death diameter sequence $\{d_i\}$ and the GPD parameters μ_0, σ_0 and ξ_0.

3 Construct an equidistant bandwidth sequence of length ℓ starting from the βth percentile of $\{d_i\}$ to $\lambda \max_i d_i$. Call the bandwidth sequence $\{b_k\}_{k=1}^{\ell}$.

4 **for** *k from* 1 *to* ℓ **do**

5   Using $h = (b_k)^{2/p}$ and $H = hI$ compute kernel density estimates and leave-one-out kernel density estimates using the scaled Epanechnikov kernel.

6   Denote the kde of x_i by y_i and leave-one-out kde by y_{-i}.

7   Using the GPD parameters μ_0, σ_0 and ξ_0 find the GPD probability of the leave-one-out kde values $\{-\log(y_{-i})\}_{i=1}^{N}$, i.e., $P(-\log(y_{-i})|\mu_0, \sigma_0, \xi_0)$ for all i.

8   If $P(-\log(y_{-i})|\mu_0, \sigma_0, \xi_0) < \alpha$, then declare x_i as an outlier and let $y_{ik} = 1$.

Next, we extend this notion of persistence to include significance levels $\alpha \in \{0.01, 0.02, \ldots, 0.1\}$.

We define the *strength* of an outlier to be inversely proportional to its significance level. Let x_j be an outlier identified at significance level α, where α is the smallest significance level for which x_j

is identified as an outlier. Then

$$\text{strength}(\boldsymbol{x}_j) = \frac{0.11 - \alpha}{0.01}. \tag{6}$$

Thus, if a point is identified as an outlier with a significance level $\alpha = 0.01$, then it has strength 10, and an outlier with $\alpha = 0.1$ has strength 1. To compute persistence over significance levels, the only modification that needs to be done to Algorithm 2 is to fix $\alpha = 0.1$ and to record the minimum significance level if $P\left(-\log(y_{-i})|\mu_0, \sigma_0, \xi_0\right) < 0.1$. Then we can use equation (6) to compute its strength.
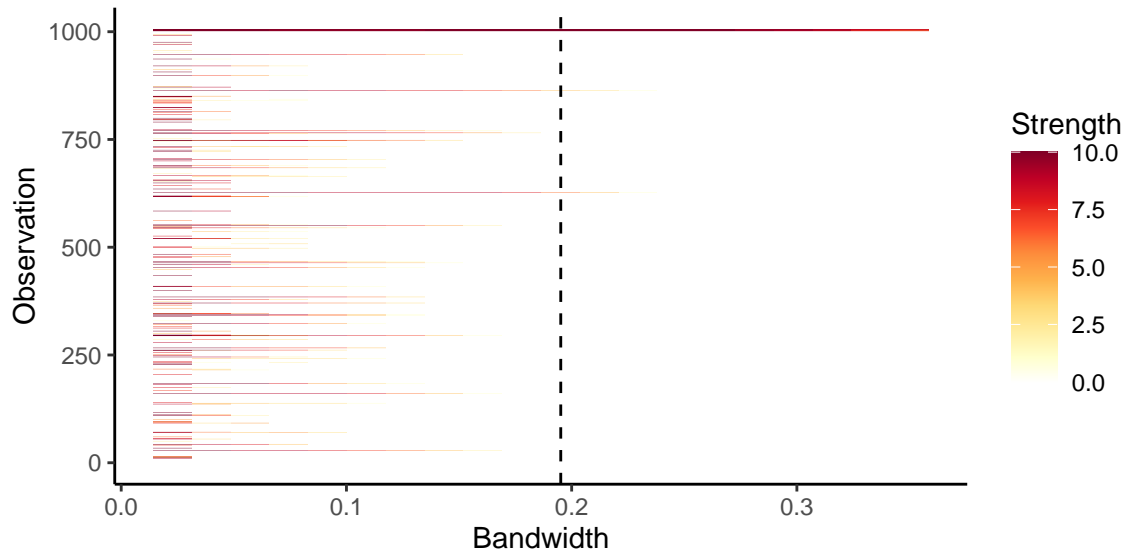

Figure 7: *Outlier persistence over different bandwidth values and their strengths. The dashed line corresponds to the lookout bandwidth d_*.*

Figure 7 shows the persistence of outliers over different bandwidth values and significance levels for the dataset in Figure 6. We see that points 1001-1005 are identified as outliers with high strength even for large bandwidths. This gives a comprehensive representation of outliers as it encapsulates the bandwidth as well as the strength, which corresponds to significance.

4 Experiments with synthetic data

In this Section we first explore outlier persistence using simple examples. Then we conduct experiments using synthetic data comparing lookout to HDoutliers (Wilkinson 2017) and stray (Talagala, Hyndman & Smith-Miles 2021), both of which use extreme value theory to detect outliers.
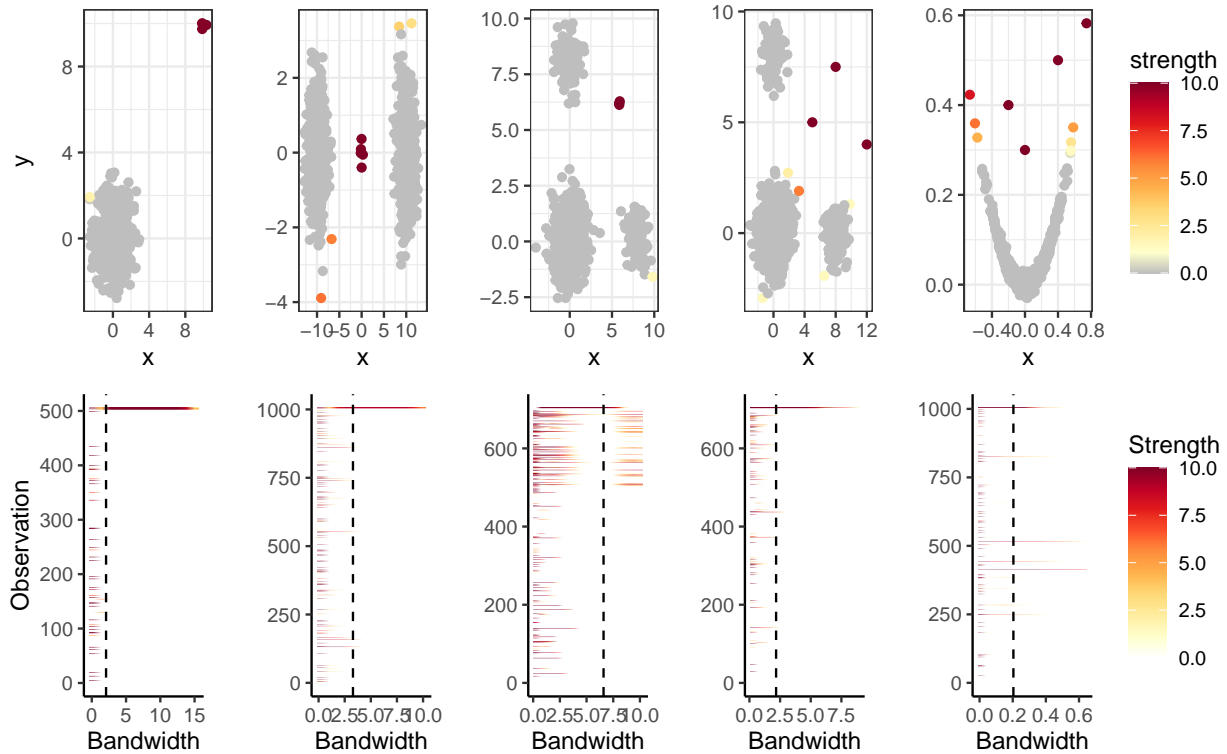
Figure 8: *Examples of outlier persistence and strength. The top row shows the data with the outliers identified by lookout with α = 0.1 colored by its strength. The bottom row shows the outlier persistence diagrams for these examples. The dashed line corresponds to the lookout bandwidth.*

4.1 Outlier persistence examples

Figure 8 shows 5 examples in $\mathbb{R}^2$, each showing the data, outliers identified by lookout, their strengths and the corresponding outlier persistence diagram. In each example, the outliers are placed at the end of the synthetic dataset, i.e. their observation indices are the highest. The dashed lines in the outlier persistence diagram indicate the bandwidth chosen by lookout. The graph at (row 1, column 1) show normally distributed data with 5 outliers in the top right corner, which are identified by lookout with high strength. Upon close inspection we see another point approximately at $(4, 0)$, which is slightly detached from the normal set of points, also being identified by lookout with low strength. The lookout probability of this point is 0.078, and as such it would only be identified as an outlier when $\alpha > 0.78$. We have used $\alpha = 0.1$ so that we can match the strengths of the outliers with the persistence diagram. The corresponding persistence diagram shows high strength outlier persistence for the points at the top right hand corner. Similarly, from other graphs in row 1, we see that outliers far away from high density regions are identified by lookout with high strength and points outside high density regions, but not so far away are identified with low strength. Some graphs, such as the graph in (row1, column2) is squashed horizontally making the x values more condensed than the y values. For the graph in (row 1, column 3) lookout has identified a cluster of outliers approximately at $(6, 6)$

with high strength as well as individual points slightly away from high density regions with low strength. The difference in graph at (row 1, column 4) is that it has single outliers instead of clusters, which are again identified by lookout. For the last graph on row 1, lookout has identified a cluster of outliers at $(-0.6, 0.35)$ with low strength and the remaining outliers with high strength. From the persistence diagrams we see that many points are identified as outliers for low bandwidth values, but only a small number of outliers persist until the bandwidth is large. In addition, some outliers persist at low strength values. For example, the persistence diagram at (row 2, column 5) has outliers around index 500 with low strength persisting until large bandwidth values.

From the examples in Figure 8 we see that lookout selects a bandwidth appropriate for outlier detection that is neither too small nor too large. In addition, lookout identifies outliers correctly. The outlier persistence diagram gives a snapshot of the outliers for varying bandwidths and significance levels increasing our understanding of the dataset and its outliers.

4.2 Comparison Study

In this section we conduct 3 experiments with synthetic data. Each experiment considers 2 data distributions; the main distribution and a handful of outliers, which are distributed differently. There are several iterations to each experiment. The iterations serve as a measure of the degree of outlyingness of the small sample. The outliers start off with the main distribution and slowly move out of the main distribution with each iteration. Consequently, in the initial iterations the points in the outlying distribution are not actual outliers as they are similar to the points in the main distribution, while in the later iterations they are quite different from the main distribution. We repeat each iteration 10 times to account for the randomness of the data generation process.

We compare the results of lookout with 2 other algorithms; HDoutliers (Wilkinson 2017) and stray (Talagala, Hyndman & Smith-Miles 2021), as both these algorithms use extreme value theory for outlier detection. For all three algorithms we use $\alpha = 0.05$. We use two evaluation metrics suited for imbalanced datasets: the F-measure and the geometric mean of sensitivity and specificity denoted by Gmean. The F-measure is defined as

$$\text{F-measure} = 2\frac{\text{precision} \times \text{recall}}{(\text{precision} + \text{recall})},$$

where

$$\text{precision} = \frac{tp}{tp + fp}, \quad \text{and} \quad \text{recall} = \frac{tp}{tp + fn},$$

where tp, fp and fn denote true positives (predicted = true, actual = true), false positives (predicted = true, actual = false) and false negatives (predicted = false, actual =true) respectively. The F-measure is undefined when both precision and recall are zero, which occurs when the true positives tp are zero. This happens when the outlier detection algorithm does not identify any correct outliers. We assign zero to the F-measure in such instances.

Sensitivity and specificity are similar evaluation metrics more frequently used in a medical diagnosis context:

$$\text{sensitivity} = \frac{tp}{tp + fn}, \quad \text{and} \quad \text{specificity} = \frac{tn}{tn + fp},$$

and

$$\text{Gmean} = \sqrt{\text{sensitivity} \times \text{specificity}},$$

where tn denotes the true negatives (predicted = false, actual = false). In fact, sensitivity and recall are two different terms denoting the same quantity of interest.

Experiment 1

For this experiment we consider two normally distributed samples in $\mathbb{R}^6$; one large and one small starting at the same location with the small sample slowly moving out in each iteration. The set of points belonging to the small sample are considered outliers. We consider 400 points in the bigger sample and 5 in the outlying sample, placed at indices 401 - 405. The points in the larger sample are distributed in each dimension as $\mathcal{N}(0, 1)$. The outliers differ from the normal points only in one dimension, i.e. they are distributed as $\mathcal{N}(2 + (i - 1)/2, 0.2)$, where i denotes the iteration. As such, in the first iteration the outliers are distributed as $\mathcal{N}(2, 0.2)$ in the first dimension and in the tenth iteration they are distributed as $\mathcal{N}(6.5, 0.2)$, showing that it has clearly moved out from the main distribution. Each iteration is repeated 10 times to account for randomness.

The top left graph of Figure 9 shows the first two dimensions of this experimental dataset in its last iteration and repetition with outliers identified by lookout shown in different colors. The top right graph shows the corresponding outlier persistence plot for this data with the dashed line denoting the lookout bandwidth. The graphs at the bottom show the performance of lookout, HDoutliers and stray with $\alpha = 0.05$. The mean F-measure of the 10 repetitions for each iteration is shown in the graph at the bottom left and the mean Gmean is shown at the bottom right. We see that for each iteration lookout surpasses HDoutliers and stray significantly.
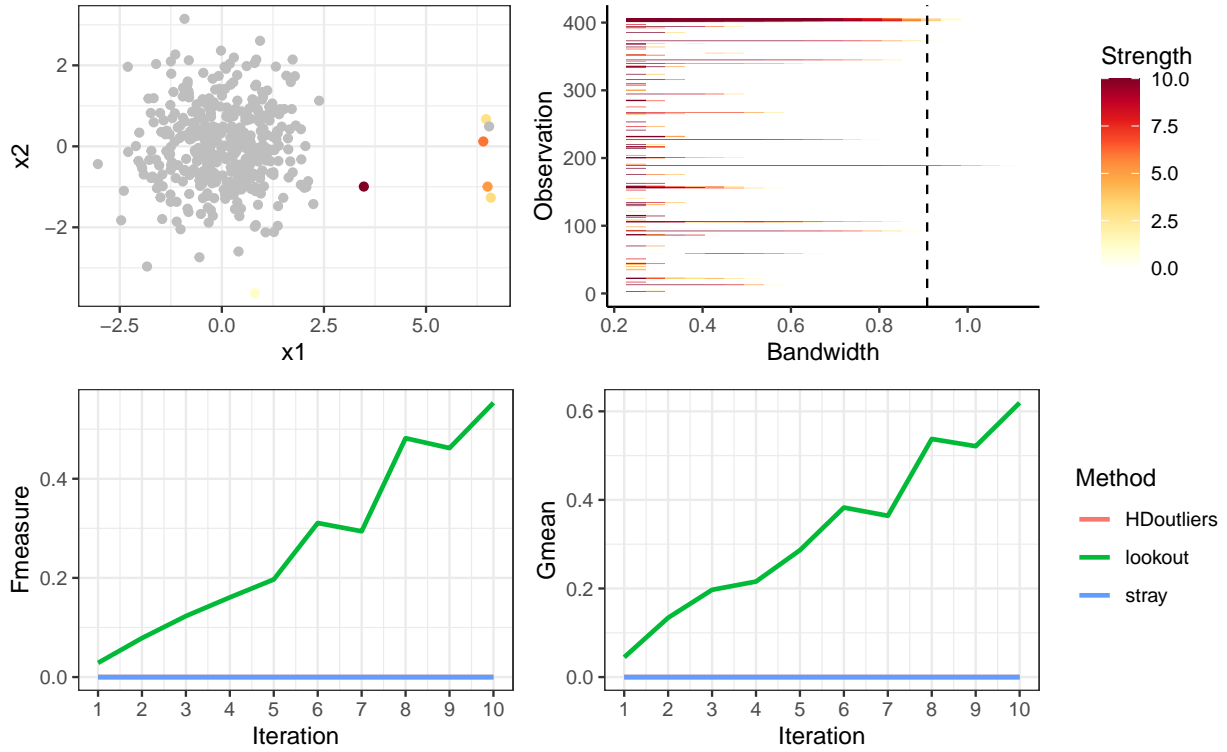
Figure 9: *Experiment 1 with outliers moving out from the normal distribution in $\mathbb{R}^6$. Graph on the top left shows the first two dimensions in the last iteration and last repetition. The graph on the top right shows the corresponding outlier persistence plot. The two graphs at the bottom show the mean F-measure and the Gmean of the 3 algorithms over 10 repetitions.*

Experiment 2

For this experiment we consider an annulus in $\mathbb{R}^2$ with outlying points moving into the center with each iteration. We consider 800 points in the annulus with 5 outlying points. The outliers are normally distributed and have a smaller standard deviation compared to the other points. The mean of the outliers in the ith iteration is $(5 - (i - 1)/2, 0)$, so that the outliers start at the right of the annulus with mean $(5, 0)$ and move in with each iteration, ending with mean $(0, 0)$. We repeat each iteration 10 times.

The graph at the top left of Figure 10 shows the points in the final iteration and repetition. The graph on the top right shows the outlier persistence for the final iteration and repetition. The outliers are placed at indices 801 - 805 and we see that they are not identified as outliers for small bandwidth values because the outliers are clustered together. This shows the importance of bandwidth selection for outlier detection when using kernel density estimates. The bandwidth selected by lookout is shown as a dashed line. The bottom two graphs show the mean F-measure and Gmean of the 3 algorithms for each iterations over 10 repetitions. We see that lookout identifies the outliers earlier compared stray and HDoutliers, which does not identify any outliers.
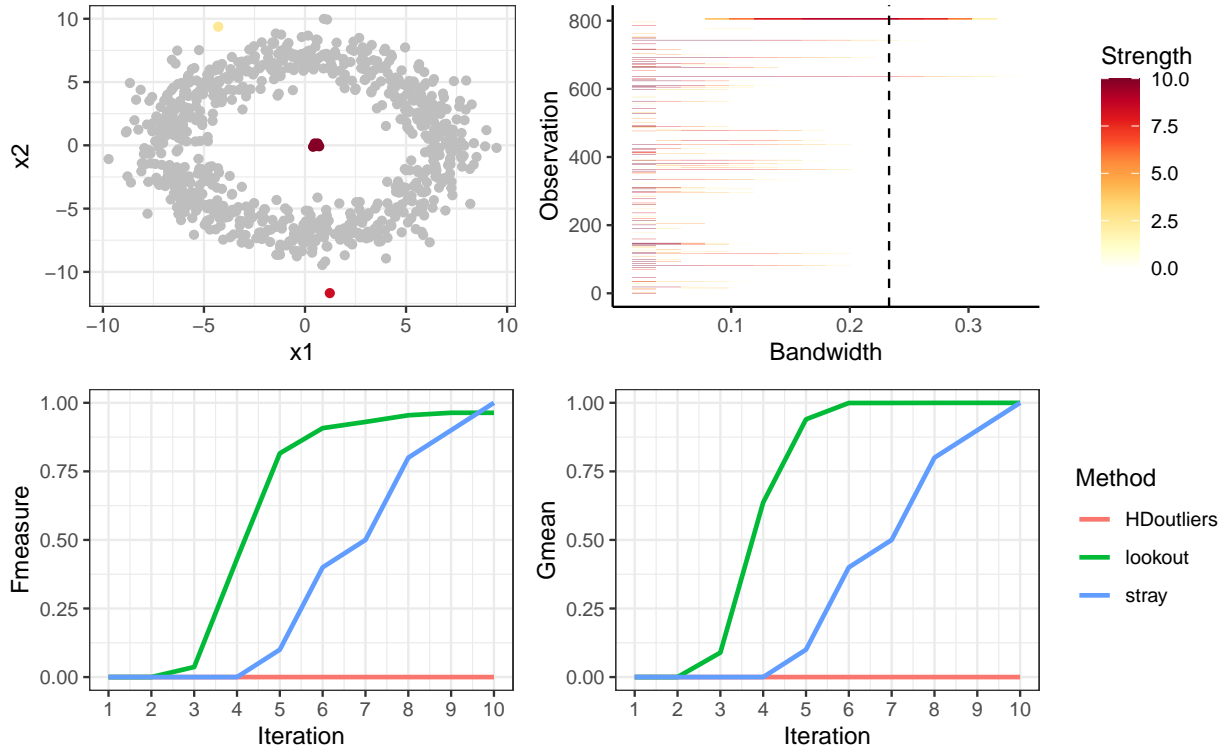
Figure 10: *Experiment 2 with outliers moving into the center of the annulus in $\mathbb{R}^2$. Graph on the top left shows the points from the last iteration and repetition. The graph on the top right shows the corresponding outlier persistence plot. The two graphs at the bottom show the mean F-measure and the Gmean of the 3 algorithms over 10 repetitions.*

Experiment 3

For this experiment we consider a unit cube in $\mathbb{R}^{20}$ with 500 points. Of these points 499 are uniformly distributed in each direction and the remaining point is considered an outlier. The outlier moves towards the point $(0.9, 0.9, \ldots, 0.9)$ with each iteration. For the ith iteration the first i coordinates of the outlier are each equal to 0.9 and the remaining coordinates are uniformly distributed in $(0, 1)$. The index of the outlier is 500. Each iteration is repeated 10 times with different randomizations.

The top graph in Figure 11 shows the outlier persistence for the last iteration and repetition. The dashed line shows the bandwidth chosen by lookout. The two graphs at the bottom show the comparison with HDoutliers and stray for each iteration averaged over the repetitions. We see that HDoutliers and stray do not identify the outlier for any iteration. Lookout clearly gives better performance and achieves an average Gmean of 0.999 from the 16th iteration onward.
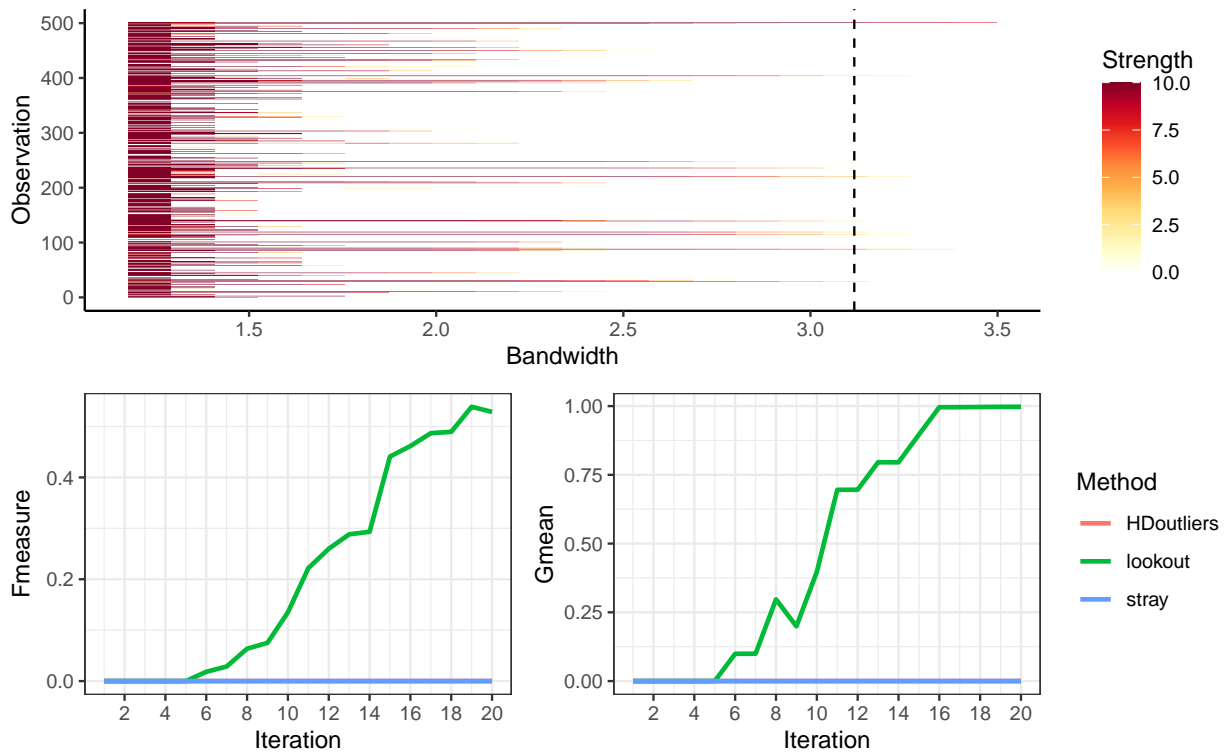
Figure 11: *Experiment 3 with an outlier moving to $(0.9, 0.9, \dots)$. Graph on the top shows the outlier persistence plot for the last iteration and repetition. The two graphs at the bottom show the mean F-measure and the Gmean of the 3 algorithms over 10 repetitions.*

5 Results on a data repository

For this Section we use the data repository used in Kandanaarachchi et al. (2019b). This repository has more than 12000 outlier detection datasets that were prepared from classification datasets. Dataset preparation involves downsampling the minority class in classification datasets, converting the categorical variables to numerical and accounting for missing values, all of which is detailed in Kandanaarachchi et al. (2020).

Figure 12 shows the letter-value plots (Hofmann, Wickham & Kafadar 2017) of performance differences between (a) lookout and HDoutliers and (b) lookout and stray, using Gmean and Fmeasure after removing the entries that have zero values for all three algorithms. Letter-value plots enhance traditional box-plots by including more detailed information making them suitable for large datasets. The letter-value plots in Figure 12 are area adjusted, that is, the area of each box represents the number of points/datasets in it. The median is represented by a white line and each black box represents a fourth of the population denoted by F in the legend. The next box represents a eighth denoted by an E and each successive box represents half of the previous one.
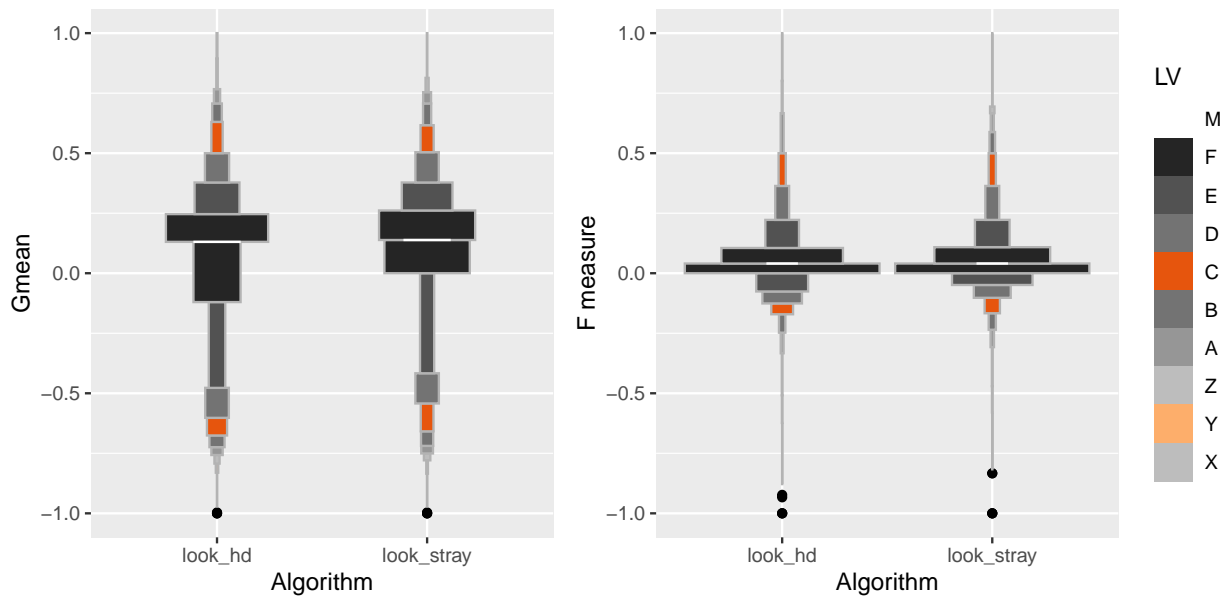
Figure 12: *Letter value plots of performance differences between 1. lookout and HDoutliers, 2. lookout and stray using Gmean and Fmeasure.*

Table 1: *Summary statistics for comparing lookout with the HDoutliers and stray algorithms.*

Statistic	lookout - HDoutliers	lookout - stray
Fmeasure		
Median	0.1307	0.1385
Mean	0.0487	0.0838
95% Confidence Interval	(0.0385,0.0589)	(0.0743,0.0932)
Gmean		
Median	0.0405	0.0406
Mean	0.0711	0.0768
95% Confidence Interval	(0.0662,0.0760)	(0.0721,0.0815)

From the graphs in Figure 12 we see that the total area of the letter-value plots above zero is larger than the area below zero. This signifies that more datasets have positive Gmean and F-measure values for lookout - HDoutliers and lookout - stray performance values compared to the negative values. This is confirmed by the results in Table 1. We see that for both Gmean and F-measure the median, mean and the 95% confidence interval from Student's t-tests are away from zero. In fact, the Gmean has median values 0.1307 and 0.1385 for lookout - HDoutliers and lookout - stray respectively. Similarly, the corresponding F-measure values are 0.0405 and 0.0406. Given that both Gmean and F-measure are bounded by 1, this shows that lookout gives better performance than HDoutliers or stray.

6 Conclusions

Outlier detection methods that use kernel density estimates generally employ a user defined parameter to construct the bandwidth. Selecting a bandwidth for outlier detection is different from selecting a bandwidth for general data representation, because the goal is to make outliers have lower density estimates compared to the non-outliers. In addition, it is a challenge to select an appropriate bandwidth in high dimensions. To make outliers have lower density estimates compared to the rest, a reasonably large bandwidth needs to be chosen. We introduced an algorithm called *lookout* that uses persistent homology to select the bandwidth. Lookout uses leave-one-out kernel density estimates and EVT to detect outliers. We compared the performance of lookout with HDoutliers and stray, two algorithms that use EVT to detect outliers on experimental data and on a large data repository and showed that lookout achieves better performance.

We also introduced *outlier persistence*, a concept that explores the birth and death of outliers with changing bandwidth and significance values. Outlier persistence gives a bigger picture of outliers, taking a step back from fixed parameter values. It explores the bandwidth and significance parameters and highlights the outliers that persist over a range of bandwidth values and their significance levels. It is a useful measure that increases our understanding of outliers.

7 Supplementary materials

The R package `lookout` is available at https://github.com/sevvandi/lookout. The outlier detection data repository used in Section 5 is available at Kandanaarachchi et al. (2019b). The programming scripts used in Sections 4 and 5 are available at https://github.com/sevvandi/supplementary_material/tree/master/lookout.

References

Bommier, E (2014). "Peaks-over-threshold modelling of environmental data". MA thesis. Department of Mathematics, Uppsala University.

Brunson, JC, R Wadhwa & J Scott (2020). *ggtda: 'ggplot2' Extension to Visualize Topological Persistence*. R package version 0.1.0. https://github.com/rrrlw/ggtda.

Burridge, P & AM Taylor (2006). Additive outlier detection via extreme-value theory. *Journal of Time Series Analysis* **27**(5), 685–701.

Carlsson, G (2009). Topology and data. *Bulletin of the American Mathematical Society* **46**(2), 255–308.

Carlsson, G, T Ishkhanov, V De Silva & A Zomorodian (2008). On the local behavior of spaces of natural images. *International journal of computer vision* **76**(1), 1–12.

Chaudhuri, P & JS Marron (1999). SiZer for Exploration of Structures in Curves. *Journal of the American Statistical Association* **94**(447), 807.

Clifton, DA, L Clifton, S Hugueny & L Tarassenko (2014). Extending the generalised pareto distribution for novelty detection in high-dimensional spaces. *Journal of Signal Processing Systems* **74**(3), 323–339.

Coles, S (2001). *An introduction to statistical modeling of extreme values*. Vol. 208. Springer.

Ghrist, R (2008). Barcodes: the persistent topology of data. *Bulletin of the American Mathematical Society* **45**(1), 61–75.

Hofmann, H, H Wickham & K Kafadar (2017). Letter-Value Plots: Boxplots for Large Data. *Journal of Computational and Graphical Statistics* **26**(3), 469–477.

Kandanaarachchi, S & RJ Hyndman (2021). *lookout: Leave One Out Kernel Density Estimates for Outlier Detection*. R package version 0.0.0.9000. https://github.com/sevvandi/lookout.

Kandanaarachchi, S, MA Muñoz, RJ Hyndman & K Smith-Miles (2019a). On normalization and algorithm selection for unsupervised outlier detection. *Data Mining and Knowledge Discovery*, 1–46.

Kandanaarachchi, S, MA Muñoz, RJ Hyndman & K Smith-Miles (2020). On normalization and algorithm selection for unsupervised outlier detection. *Data Mining and Knowledge Discovery* **34**(2), 309–354.

Kandanaarachchi, S, MA Munoz Acosta, K Smith-Miles & RJ Hyndman (2019b). *Datasets for outlier detection*. https://doi.org/10.26180/5c6253c0b3323.

Minnotte, MC & DW Scott (1993). The Mode Tree: A Tool for Visualization of Nonparametric Density Features. *Journal of Computational and Graphical Statistics* **2**(1), 51.

Perea, JA & J Harer (2015). Sliding windows and persistence: An application of topological methods to signal analysis. *Foundations of Computational Mathematics* **15**(3), 799–838.

Pickands, J (1975). Statistical Inference Using Extreme Order Statistics. *The Annals of Statistics* **3**(1), 119–131.

Qin, X, L Cao, EA Rundensteiner & S Madden (2019). Scalable kernel density estimation-based local outlier detection over large data streams. *Advances in Database Technology - EDBT* **2019-March**, 421–432.

R.D. Reiss & M.Thomas (2001). *Statistical Analysis of Extreme Values with Applications to Insurance, Finance, Hydrology and Other Fields*. 3rd ed. Birkhäuser, p. 516.

Schubert, E, A Zimek & HP Kriegel (2014). Generalized outlier detection with flexible kernel density estimates. *SIAM International Conference on Data Mining 2014, SDM 2014* **2**, 542–550.

Scott, DW (1994). *Multivariate Density Estimation: Theory, Practice, and Visualization.* Vol. 89. 425, p. 359.

Stephenson, AG (2002). evd: Extreme Value Distributions. *R News* **2**(2), 31–32. `https://CRAN.R-project.org/doc/Rnews/`.

Talagala, PD, RJ Hyndman & K Smith-Miles (2021). Anomaly Detection in High-Dimensional Data. *Journal of Computational and Graphical Statistics*. in press.

Talagala, PD, RJ Hyndman, K Smith-Miles, S Kandanaarachchi & MA Muñoz (2020). Anomaly Detection in Streaming Nonstationary Temporal Data. *Journal of Computational and Graphical Statistics* **29**(1), 13–27.

Tang, B & H He (2017). A local density-based approach for outlier detection. *Neurocomputing* **241**, 171–180.

Topaz, CM, L Ziegelmeier & T Halverson (2015). Topological data analysis of biological aggregation models. *PloS ONE* **10**(5), e0126383.

Wadhwa, RR, DFK Williamson, A Dhawan & JG Scott (2018). TDAstats: R pipeline for computing persistent homology in topological data analysis. *Journal of Open Source Software* **3**(28), 860. `https://doi.org/10.21105/joss.00860`.

Wang, Z & DW Scott (2019). Nonparametric density estimation for high-dimensional data—Algorithms and applications. *Wiley Interdisciplinary Reviews: Computational Statistics* **11**(4). arXiv: 1904.00176.

Wasserman, L (2018). Topological data analysis. *Annual Review of Statistics and Its Application* **5**, 501–532.

Wilkinson, L (2017). Visualizing big data outliers through distributed aggregation. *IEEE transactions on visualization and computer graphics* **24**(1), 256–266.