HDiR: An R Package for Computation and Nonparametric Plug-in Estimation of Directional Highest Density Regions and General Level Sets

by Paula Saavedra-Nieves, Rosa M. Crujeiras

Abstract A deeper understanding of a distribution support, being able to determine regions of a certain (possibly high) probability content is an important task in several research fields. Package **HDiR** for R is designed for exact computation of directional (circular and spherical) highest density regions and density level sets when the density is fully known. Otherwise, **HDiR** implements nonparametric plug-in methods based on different kernel density estimates for reconstructing this kind of sets. Additionally, it also allows the computation and plug-in estimation of level sets for general functions (not necessarily a density). Some exploratory tools, such as suitably adapted distances and scatterplots, are also implemented. Two original datasets and spherical density models are used for illustrating **HDiR** functionalities.

1 An overview on directional general level sets and highest density regions

When analyzing a data distribution, it is often the case that for a deeper understanding of the modelling problem, it is interesting to determine regions on the density support exceeding a certain threshold on the density function values. These regions are known as density level sets and, if the density is unknown, such a task can be accomplished from a set estimation perspective. Set estimation deals with the problem of reconstructing a set (or estimating any of its features such as its boundary or its volume) from a random sample of points intimately related to it. Since Hartigan (1975) establishes the notion of clusters as connected components of a density level set, the reconstruction of this particular type of sets has been widely considered in the literature (mainly for densities supported on an Euclidean space). There are only very few contributions where density level set theory has been extended to more general domains such as the unit sphere or manifolds. Cuevas et al. (2006) consider the estimation of level sets for general functions (not necessarily a density) such as regression curves, providing some consistency theoretical results and showing a density level set on the sphere for illustration. More recently, the reconstruction of density level sets on manifolds is studied in Cholaquidis et al. (2022), who also presents some simulations illustrating the performance of their approach on the torus and on the sphere.

Let X be a random vector taking values on a $d-$dimensional unit sphere S^{d-1} with density f and $t > 0$, the goal of (directional) density level set estimation is to reconstruct the set

$$G_f(t) = \{x \in S^{d-1} : f(x) \geq t\}. \tag{1}$$

from a random sample of points $\mathcal{X}_n = \{X_1, \cdots, X_n\}$ of X when f is unknown. As an illustration, some (theoretical) level sets are shown in Figure 1 by representing $G_f(t)$ for a circular (left) and a spherical density (right) when three different values of the level t are chosen. The threshold t is represented through a dotted line for the circular case. Note that, if large values of t are considered, $G_f(t)$ coincides with the greatest modes of the circular/spherical distribution. However, for small values of t, the level set $G_f(t)$ is practically equal to the support of the distribution. Therefore, cluster definition via connected components in Hartigan (1975) is clearly related to the notion of mode. Note also that the computation of the number of modes considering the values of a density over a certain range of values for the level t, would enable the construction of a directional cluster tree. Azzalini and Torelli (2007) already present this statistical tool for Euclidean data. Moreover, the association between clusters and modes is the basis of modal clustering methodology (see Menardi, 2016 for a review on this topic). Most modal clustering algorithms are based on the application of a mode-seeking numerical method to the sample points and assigning the same cluster to those data that are iteratively shifted to the same limit value. Examples of such procedures include the mean shift algorithm that has been already studied in S^{d-1} (see, for instance, Chang-Chien et al., 2010 and Yang et al., 2014).

Despite a practitioner may be interested in determining this type of regions, the value of the level t could be (in principle) unknown in real situations. In practice, it is quite common to assume that the set in (1) must satisfy a probability content previously established. Following Box and Tiao

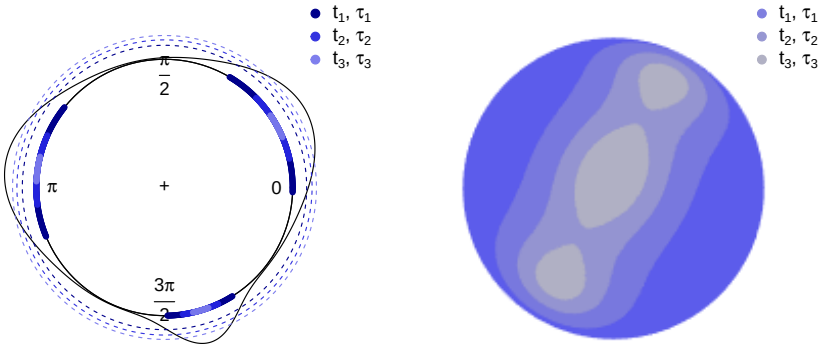

Figure 1: For a circular density (left) and a spherical density (right), level set $G_f(t)$ for $t = t_1$, $t = t_2$ and $t = t_3$ verifying $0 < t_1 < t_2 < t_3$. Equivalently, HDR $L(f_\tau)$ for $\tau = \tau_1 = 0.2$, $\tau = \tau_2 = 0.5$ and $\tau = \tau_3 = 0.8$.

(1973), Hyndman (1996) and, more recently, Azzalini and Torelli (2007), Saavedra-Nieves and Crujeiras (2021b) generalize the definition of HDRs from the Euclidean to the directional setting, providing a plug-in estimation method. Specifically, HDRs are a kind of density level sets where the set probability content is fixed instead of the level t. The estimation of HDRs involves further complexities given that the threshold must be computed from the previously fixed probability content. Formally, given $\tau \in (0, 1)$, the $100(1-\tau)\%$ HDR is the subset

$$L(f_\tau) = \{x \in S^{d-1} : f(x) \geq f_\tau\}, \tag{2}$$

where f_τ can be seen as the largest constant such that

$$\mathbb{P}(X \in L(f_\tau)) \geq 1 - \tau,$$

with respect to the distribution induced by f. Figure 1 also shows the HDR $L(f_\tau)$ for a circular and a spherical densities with three different values of τ. Note that, if large values of τ are considered, $L(f_\tau)$ is equal to the greatest modes and the most distinct clusters can be easily identified. However, for small values of τ, $L(f_\tau)$ is almost equal to the support of the distribution.

To sum up, given a value of t, the computation of the level set established in (1) (and of its connected components) is a quite simple mathematical task when f is known. Under this assumption and taking a fixed $\tau \in (0, 1)$, determining the HDR introduced in (2) presents a similar complexity but, in this case, it is additionally necessary to determine the threshold f_τ. In particular, numerical integration methods can be applied to solve that problem. However, when the density f is assumed to be unknown and a random sample $\mathcal{X}_n \in S^{d-1}$ generated from f is the only available information to reconstruct the set, nonparametric set estimation techniques such as plug-in methods must be considered in order to reconstruct the connected components of the set. Perhaps due to its practical importance, Euclidean HDRs plug-in algorithms based on kernel smoothing have been widely studied even solving the problem of selecting an appropriate smoothing parameter specifically devised for the HDR reconstruction (see Baíllo and Cuevas, 2006, Samworth and Wand, 2010 or Casa et al., 2020). In the directional setting, given that a proper definition of the HDR $L(f_\tau)$ was not available, no work on this area had been carried out until the recent contribution by Saavedra-Nieves and Crujeiras (2021b).

The contents of this paper, describing the contributions in **HDiR**, mainly focus on computation and plug-in estimation of highest density regions (HDRs) and density level sets in the circle and the sphere. Although general level sets can be also analysed using **HDiR**, we will not formally detail aspects on their computation and on their plug-in reconstruction given that they can be seen as a direct generalisation of those introduced for density level sets by replacing the density by the general function under study. Therefore, with the objective of showing the capabilities of the **HDiR** package for exact computation of directional HDRs and density level sets when f is known and for plug-in estimation otherwise, this paper is organized as follows. First, a basic overview on nonparametric plug-in estimation methods is given. Initially, the classical directional kernel density estimator is briefly introduced, as it is the key tool for plug-in reconstruction and exploratory methods. Then, the problems of threshold estimation (with known and unknown density) and specific bandwidth selection for HDRs are considered. Circular confidence regions for HDRs are also established. Next, the reader will find a guided tour across **HDiR** package, illustrating its use with simulated examples first and with two real data examples later. Following the perspective in Cuevas et al. (2006), **HDiR** also allows the computation and plug-in estimation of general level sets. A reconstruction example of

a (circular) regression level set is detailed. Moreover, distances between sets and circular/spherical scatterplots are also described as exploratory tools. Finally, some discussion is provided, considering on the possible extensions of the package.

2 Plug-in estimation methods

This section provides a brief background on the design of plug-in tools included in **HDiR** for directional (circular and spherical) HDR and density level set estimation. Following Cuevas et al. (2006), if a nonparametric estimator is available for a general function, this methodology may be directly extended for reconstructing the corresponding level sets.

Plug-in estimation methods for HDRs and level sets

Although there are other nonparametric alternative routes for level set estimation, the plug-in approach has received considerable attention in the Euclidean literature (see Tsybakov, 1997, Baíllo, 2003, Mason and Polonik, 2009, Rigollet et al., 2009, Mammen and Polonik, 2013 or Chen et al., 2017). This is with no doubt a natural methodology, which can be generalized to the directional setting as in Saavedra-Nieves and Crujeiras (2021b). Given that level set estimation is a simpler problem than HDR reconstruction, we will restrict to this last setting in what follows. Given a random sample $\mathcal{X}_n \in S^{d-1}$ of the unknown directional density f, plug-in methods reconstruct the $100(1-\tau)\%$ HDR namely $L(f_\tau)$ in (2) as

$$\hat{L}(\hat{f}_\tau) = \{x \in S^{d-1} : f_n(x) \geq \hat{f}_\tau\}, \tag{3}$$

where $\hat{f}_\tau$ is an estimator of the threshold f_τ and f_n denotes a nonparametric directional density estimator. Package **HDiR** implements the kernel density estimator provided in Bai et al. (1989) ($d > 2$). From $\mathcal{X}_n$, it is defined at a point $x \in S^{d-1}$ as

$$f_n(x) = \frac{1}{n}\sum_{i=1}^{n} K_{vM}(x; X_i; 1/h^2), \tag{4}$$

where K_{vM} denotes the von Mises-Fisher kernel density and $1/h^2 > 0$ is the concentration parameter.

Following Bai et al. (1989), package **HDiR** also enables to use any kernel function (not necessarily the von Mises-Fisher density implemented by default). An example where an uniform kernel is considered will be presented later. Even more generally, **HDiR** would allow the user to define different density estimators that the one introduced in (4). See, for instance, Pelletier (2005).

As for the concentration parameter $1/h^2$, it plays an analogous role to the bandwidth in the Euclidean case. For small values of $1/h^2$, the density estimator is oversmoothed. The opposite effect is obtained as $1/h^2$ increases. Hence, the choice of h is a crucial issue. For simplicity, in what follows, we refer to h as bandwidth parameter. Many approaches for selecting h in practice, in circular and even directional settings, have been proposed in the literature (see Taylor, 2008, Oliveira et al., 2012, Hall et al., 1987, Di Marzio et al., 2011 or García-Portugués, 2013). All these existing proposals designed for density estimation are implemented in the package **NPCirc** and their aim is to minimize some error criterion on the target density. However, such a bandwidth selector may not be adequate for HDRs or level set estimation. As far as we know, such a tool was not available in the directional setting until the selector by Saavedra-Nieves and Crujeiras (2021b). It is also available in package **HDiR**. Different plug-in estimators for HDRs emerge from the consideration of all these bandwith selectors.

For the circular and the spherical densities shown in Figure 1, now Figure 2 contains the HDR plug-in estimators (bluish colours) for $\tau = 0.5$ computed using cross-validation bandwidths and samples of sizes $n = 100$ and $n = 500$, respectively. Although the theoretical circular HDR is composed by three connected components (see Figure 1), the plug-in estimator is able to detect only the two biggest clusters when $n = 100$. In order to assess the agreement of a given estimate with the theoretical target, distances between sets are the usual tools to measure the discrepancies between the theoretical sets and the corresponding empirical reconstructions. One of the most common distances in the Euclidean setting is the Hausdorff distance between the boundaries of both sets.

If the target is the reconstruction of a HDR or a density level set, the Hausdorff metric is a suitable error criterion in the directional setting (see Cuevas et al., 2006 and Cholaquidis et al., 2022). If A and B are non-empty compact sets in the $d-$dimensional Euclidean space, the Hausdorff distance between A and B is defined as follows

$$d_H(A,B) = \max\left\{\sup_{x\in A} d_E\left(\{x\},B\right), \sup_{y\in B} d_E\left(\{y\},A\right)\right\},$$

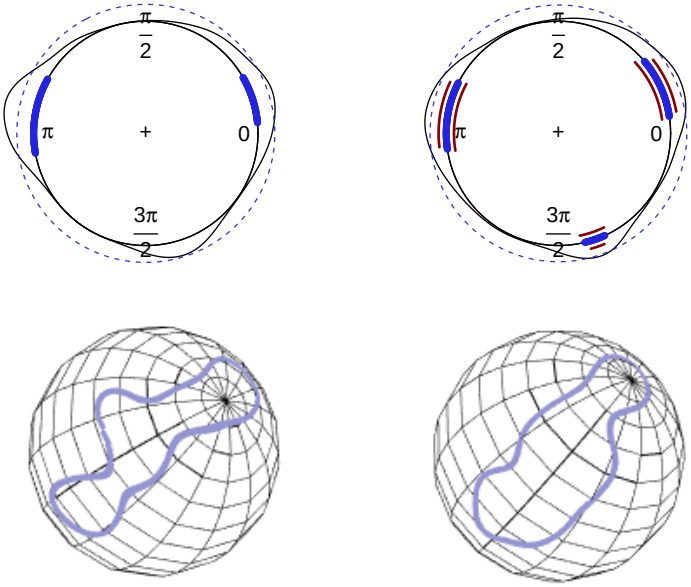

Figure 2: For the circular density shown in Figure 1 (first row), $\hat{L}(\hat{f}_\tau)$ (bluish colour) for $\tau = \tau_2 = 0.5$ computed from $\mathcal{X}_{100}$ (first column) and $\mathcal{X}_{500}$ (second column). Additionally, confidence regions are represented (dark red colour) for the second estimation. For the spherical density shown in Figure 1 (second row), $\hat{L}(\hat{f}_\tau)$ (bluish colour) for $\tau = \tau_2 = 0.5$ computed from $\mathcal{X}_{100}$ (first column) and $\mathcal{X}_{500}$ (second column).

where $d_E(\{x\}, B) = \inf_{y \in B}\{d_E(x, y)\}$ being $d_E(x, y)$ the Euclidean distance between two points. However, this metric d_H is not completely successful in detecting shape-related differences. For instance, two sets can be very close in Hausdorff distance and still show quite different shapes. This typically happens where the boundaries ∂A and ∂B are far apart, no matter the proximity of A and B. So, a natural way to reinforce the notion of visual proximity between two sets provided by Hausdorff distance is to account also for the proximity of their respective boundaries. In particular, Hausdorff distance between the boundaries of the theoretical HDR and its plug-in reconstruction is a measure of the estimation error. **HDiR** allows to compute Euclidean and Hausdorff distances between the frontiers of two arbitrary sets on the circle and on the sphere.

Threshold estimation and confidence regions for HDRs

For a given $\tau \in (0, 1)$, determining the set $L(f_\tau)$ in (2) and its plug-in estimator $\hat{L}(\hat{f}_\tau)$ in (3) involve the exact computation and the estimation of the threshold f_τ, respectively. As in the Euclidean setting, both tasks require the use of numerical integration methods. Specifically, **HDiR** uses the classical trapezoidal rule in the circular setting. However, for the spherical case, the computational cost becomes a major issue due to the complexity of the numerical integration algorithms considered on high dimensional spaces. It should be noted that package **SphericalCubature** includes some functions for solving numerical integration over spheres. However, it does not provide sufficiently accurate solutions for our problem.

An alternative approach is implemented in the internal function `sphere.integration` of **HDiR**. Specifically, the proposed numerical integration procedure on the sphere requires the definition of a triangular mesh, such as the ones depicted in Figure 3, obtained from the projection over the sphere of triangular meshes on an embedded icosaedrum. This type of mesh guarantees that there is not a prevailing direction. For computing the corresponding spherical integral, the Cartesian coordinates of the mesh vertices are transformed into spherical coordinates and standard quadrature formulae are applied in each triangle over the plane formed by the azimuthal and polar angles (see Strang and Fix, 1973).

Package **HDiR** additionally includes a computationally feasible approach for estimating f_τ in the circular and spherical context. As before, let X be a random vector with directional density f and let $Y = f(X)$ be the random vector obtained by transforming X by its own density function. Since $\mathbb{P}(f(X) \geq f_\tau) = 1 - \tau$, f_τ is exactly the $\tau-$ quantile of Y. Saavedra-Nieves and Crujeiras (2021b) establish that f_τ can be estimated as a sample quantile from a set of independent and identically distributed random vectors with the same distribution as Y. In particular, if $\mathcal{X}_n = \{X_1, \cdots, X_n\}$

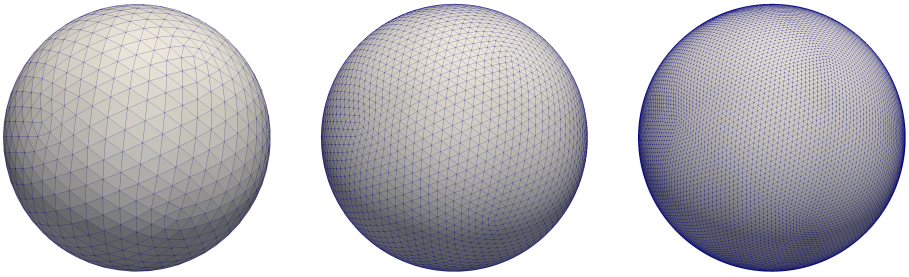

Figure 3: 3D Cartesian meshes for numerical integration on the unit sphere S^2 composed by a total of 2000 (left), 8000 (center) and 32000 (right) triangular cells.

denotes a set of independent observations in S^{d-1} from a density f, $\{f(X_1), \cdots, f(X_n)\}$ is a set of independent observations from the distribution of Y. Let $f_{(j)}$ be the $j-$th largest value of $\{f(X_i)\}_{i=1}^n$ so that $f_{(j)}$ is the (j/n) sample quantile of Y. We shall use $f_{(j)}$ as an estimate of f_τ. Specifically, we choose $\hat{f}_\tau = f_{(j)}$ where $j = \lfloor \tau n \rfloor$. Threshold values in Figure 2 were estimated following this approach.

This estimation method presents a lower computational complexity than numerical integration algorithms. Furthermore, it involves a statistical approximation. Therefore, it is possible to establish confidence intervals in order to quantify uncertainty in estimates of f_τ and, as direct consequence, to establish confidence regions for HDR's. Following Hyndman (1996), the simplest case where X is a circular random variable is considered by Saavedra-Nieves and Crujeiras (2021b). Standard asymptotic results for a sample in Cox and Hinkley (1979) ensure that $\hat{f}_\tau$ is asymptotically normally distributed with mean f_τ and variance $\tau(1-\tau)/(n[g(f_\tau)]^2)$ where

$$g(y) = y \sum_{i=1}^{n(y)} |f^{'}(z_i)|^{-1},$$

and $\{z_i\}$ denote those points in the sample space of X such that $f(z_i) = y, i = 1, 2, \cdots, n(y)$. Figure 2 (first row, right) depicts the confidence regions obtained with package **HDiR** (in dark red colour) for the circular model presented in Figure 1.

Suitable bandwidth selection for HDRs estimation

The plug-in reconstruction of the directional HDRs in (3) also involves the calculation of the kernel density estimator in (4) that is known to be heavily dependent on the selection of h. Package **HDiR** implements the proposal in Saavedra-Nieves and Crujeiras (2021b) where the first selector of h specifically designed for HDRs reconstruction is presented. The idea is to use an error criterion that quantifies the differences between the theoretical region and its plug-in reconstruction. In the real-valued setting, Samworth and Wand (2010) use a similar idea in order to propose one of the first bandwidth selectors for HDRs estimation.

The closed expression of the Hausdorff distance between the boundaries of the HDR and its plug-in reconstruction, $d_H(\partial L(f_\tau), \partial \hat{L}(\hat{f}_\tau))$, is not known in the directional case. However, such a distance could be approximated through a bootstrap procedure. With this view in mind, Saavedra-Nieves and Crujeiras (2021b) consider a new bandwidth selector as follows:

$$h_* = \arg\min_{h>0} \mathbb{E}_B \left[d_H(\partial L^*(\hat{f}_\tau^*), \partial \hat{L}(\hat{f}_\tau)) \right], \tag{5}$$

where $\mathbb{E}_B$ denotes the bootstrap expectation with respect to random samples of points $\mathcal{X}_n = \{X_1^*, \cdots, X_n^*\}$ generated from the directional kernel f_n that, of course, requires a pilot bandwidth chosen for computing $\hat{L}(\hat{f}_\tau)$.

3 Using HDiR

This section presents an overview of the structure of the package. **HDiR** (Saavedra-Nieves and Crujeiras, 2021a) is an easy-to-use toolbox that R practitioners can use for computation or plug-in estimation of directional highest density regions and general level sets defined on the circle and sphere. The methods included in the package facilitate both data exploration and nonparametric estimation of

the target regions. Functions in this library automatize the required operations for the computation of this kind of sets. First, we will describe the real data sets included in the package. Then, the functions available in **HDiR** are detailed. Of course, there exist several libraries in the CRAN repository of R dealing with plug-in estimation of Euclidean level sets and HDRs. In particular, the library **pdfCluster** (Azzalini et al., 2014) provides a routine to estimate the probability density function by kernel methods from a set of linear data with arbitrary dimension. The main focus is on cluster analysis via kernel density estimation according to the approach by Hartigan (1975). For modal clustering, package **LPCM** (Einbeck and Evers, 2019) implements the mean-shift algorithm and **Modalclust** (Cheng and Ray, 2014) performs the method for mode seeking introduced in Li et al. (2007). There are also other packages that do not solve the task of estimate HDRs directly, but they usually allow to compute the linear kernel density estimator and, therefore, address HDRs graphical representation (not necessarily with an appropriate estimate). A brief summary of the capabilities of these libraries are provided below.

- **denpro** (Klemelä, 2005, Klemelä, 2006, Klemelä, 2008, Klemelä, 2009, Holmström et al., 2017 and Klemelä, 2015): This library allows to visualize multivariate densities and density estimates with level set trees and also to represent level sets with shape trees in moderate dimensional cases. Furthermore, the kernel estimator implemented by default could be replaced by other density estimates.
- **hdrcde** (Hyndman et al., 2018): This package computes Euclidean HDRs in one and two dimensions. The specific HDR bandwidth selector proposed in Samworth and Wand (2010) is also implemented. Confidence regions for one-dimensional HDRs and bivariate HDRs scatterplots (colouring sample points according to the region in which they fall) are also available.
- **lsbs** (Doss and Weng, 2018): This package implements the bandwidth selector for two-dimensional Euclidean level sets and HDRs proposed in Doss and Weng (2018). A plug-in strategy to estimate the asymptotic risk function and minimize to get the optimal bandwidth matrix is applied.

Other packages such as **sm** (Bowman and Azzalini, 2018) and **ks** (Duong, 2007) also include tools for kernel density estimation allowing for graphical displays of density contours in the two- and three-dimensional Euclidean spaces. Moreover, there are many libraries in the CRAN repository for directional data analysis but, as far as we know, none of them solves the problem of level set or HDR reconstruction. In this section, we would like to highlight those packages including tools for kernel density estimation, both for circular and directional data:

- **circular** (Agostinelli and Lund, 2013): It is an extension of the **CircStats** package. It provides functions for the statistical analysis (descriptive statistics, circular models, hypothesis tests), graphical representation and some classical circular datasets.
- **Directional** (Tsagris et al., 2017): A collection of functions for directional data analysis are implemented in this library. Apart from hypothesis testing, discriminant and regression analysis, it allows to compute the kernel density estimation for hyper-spherical data using a von Mises-Fisher kernel.
- **DirStats** (García-Portugués, 2021): This library also allows to compute a kernel density estimator and, additionally, it implements the cross-validation and plug-in bandwidth selectors in Hall et al. (1987) and García-Portugués (2013), respectively.
- **NPCirc** (Oliveira et al., 2014): Nonparametric density and regression estimation methods for circular data are included in this package. Specifically, a circular kernel density estimation procedure is provided, jointly with different alternatives for choosing the smoothing parameter. Based on the kernel density estimator, a SiZer technique (CircSiZer) is developed for circular data. The package also includes functions for nonparametric circular regression.

Note also that there are other packages including tools for circular/directional data analysis. For instance, **CircStats** (Lund and Agostinelli, 2012) is a companion to Jammalamadaka and Sengupta (2001), although functions implemented in this package are also available in **circular**. **CircNNTSR** (Fernández-Durán and Gregorio-Domínguez, 2013) provides an alternative estimation method for circular distributions based on nonnegative trigonometric sums. **isocir** (Barragán et al., 2013) implements some routines for analyzing angular data subjected to order constraints on a unit circle. Finally, **movMF** (Hornik and Grün, 2014) is focused on mixtures of von Mises distributions, allowing to draw random samples from these models and to proceed with parameter estimation, by using an expectation-maximization algorithm.

Specifically, the goal of **HDiR** package is to provide tools for directional (circular and spherical) general level sets and HDRs exact computation also including their plug-in estimation. This library implements the first specific bandwidth selector devised for directional HDRs proposed in Saavedra-Nieves and Crujeiras (2021b), but it also allows directly user-defined bandwidth selection and to use the

Dataset	Description
`earthquakes`	Geographical coordinates (latitude and longitude) of earthquakes of magnitude greater than or equal to 2.5 degrees between October 2004 and April 2020
`sandhoppers`	Orientation of two sandhoppers species, Talitrus saltator and Talorchestia brito under different natural conditions
Function	**Description**
`circ.boot.bw`	Circular bootstrap bandwidth for HDRs estimation
`circ.distances`	Euclidean and Hausdorff distances between two sets of points on the unit circle
`circ.hdr`	Computation of HDRs and general level sets for a given circular real-valued function
`circ.plugin.hdr`	Circular plug-in estimation of HDRs and level sets and confidence regions
`circ.scatterplot`	Circular scatterplot for plug-in HDRs
`dspheremix`	Density functions for mixtures of spherical von Mises-Fisher
`rspheremix`	Random generation functions for mixtures of spherical von Mises-Fisher
`sphere.boot.bw`	Spherical bootstrap bandwidth for HDRs estimation
`sphere.distances`	Euclidean and Hausdorff distances between two sets of points on the unit sphere
`sphere.hdr`	Computation of HDRs and general level sets for a given spherical real-valued density
`sphere.plugin.hdr`	Spherical plug-in estimation of HDRs and level sets
`sphere.scatterplot`	Spherical scatterplot for plug-in HDRs

Table 1: Summary of **HDiR** package contents.

existing directional bandwidth selection methods devised for kernel density estimation. Additionally, two alternative methods for estimating the threshold f_τ (based on the proposal in Hyndman, 1996 and numerical integration methods, respectively) are developed. Moreover, confidence regions for circular HDR are also available and can be depicted for illustration. Two exploratory tools are also implemented. The first one is a scatterplot computed from HDRs plug-in reconstructions. Sample points are coloured according to the directional HDRs in which they fall. Finally, Euclidean and Hausdorff distances between sets can be also computed. Their roles are crucial to measure the distances between directional clusters or, for instance, to quantify the estimation error between the theoretical HDRs and the corresponding plug-in estimators.

A complete description of the **HDiR** package capabilities is provided in this section. The complete list of functions, illustrative density models (density functions and random sample generation) and the two novel datasets available in **HDiR**, with a brief description, can be seen in Table 1.

Data description

The package **HDiR** includes a circular and a spherical datasets, used for the illustration of the different functions. The first dataset, `sandhoppers`, contains the orientation angles (in radians between 0 and 2π) of two species of sandhoppers, Talitrus saltator and Talorchestia brito. Orientation was measured under natural conditions on the exposed nontidal sand of Zouara beach located in the Tunisian northwestern coast. Additionally, other variables of interest for analyzing the behavioral plasticity of both species were also registered. For instance, information on the month, the time of the day, the temperature, the air relative humidity or the sex of each animal is also available. This dataset was already analyzed in Scapini et al. (2002) and Marchetti and Scapini (2003). Specifically, the behavior of these two species is compared through regresion procedures. Scapini et al. (2002) conclude that Talitrus saltator showed more differentiated orientations, depending on the time of day, period of the year and sex, with respect to Talorchestia brito. As an illustration, Saavedra-Nieves and Crujeiras (2021b) also study the behavior of these two species of sandhoppers under the HDR estimation approach.

The second dataset, `earthquakes`, contains the geographical coordinates (latitude and longitude) of earthquakes of magnitude greater than or equal to 2.5 degrees on the Richter scale registered on Earth between 1st October 2004 and 9th April 2020. It can be downloaded from the website of the

European-Mediterranean Seismological Centre (EMSC)[1]. The planar points included in the dataset correspond to spherical coordinates on Earth. Due to the important damages that earthquakes of a certain intensity may cause, cluster detection of HDRs could be also useful to identify, from a real dataset, where earthquakes are specially likely. This information is crucial for decision-making, for example, to update construction codes guaranteeing a better building seismic-resistance. Saavedra-Nieves and Crujeiras (2021b) also analyze the recent world earthquakes distribution through HDRs estimation from this dataset. Results shows that the greatest mode of sample distribution is identified in the Southeast of Europe. Countries such as Italy, Greece or Turkey (located within this cluster) are, as expected, the most affected areas in the analyzed period. The second dataset, `earthquakes`, contains the geographical coordinates (latitude and longitude) of earthquakes of magnitude greater than or equal to 2.5 degrees on the Richter scale registered on Earth between 1st October 2004 and 9th April 2020. It can be downloaded from the website of the European-Mediterranean Seismological Centre (EMSC)[2]. The planar points included in the dataset correspond to spherical coordinates on Earth. Due to the important damages that earthquakes of a certain intensity may cause, cluster detection of HDRs could be also useful to identify, from a real dataset, where earthquakes are specially likely. This information is crucial for decision-making, for example, to update construction codes guaranteeing a better building seismic-resistance. Saavedra-Nieves and Crujeiras (2021b) also analyze the recent world earthquakes distribution through HDRs estimation from this dataset. Results shows that the greatest mode of sample distribution is identified in the Southeast of Europe. Countries such as Italy, Greece or Turkey (located within this cluster) are, as expected, the most affected areas in the analyzed period.

Spherical density models

Functions `dspheremix` and `rspheremix` allow to compute density functions and to generate data from the spherical distributions introduced in Saavedra-Nieves and Crujeiras (2021b). These densities represent a variety of complex structures showing multimodality and/or asymetry. Any user of package **HDiR** could use them for simulations or even for illustration purposes.

Function `dspheremix` computes the density function of 9 different spherical distributions that can be written as finite mixtures of spherical von Mises-Fisher. Function `rspheremix` is designed for random data generation from these 9 spherical models. Both functions have an argument called `model` which allows to specify a model (a number between 1 and 9) among the ones considered in Saavedra-Nieves and Crujeiras (2021b). The other inputs of `dspheremix` and `rspheremix` are `x` and `n`, respectively. `x` represents a matrix whose rows collect to points on the unit sphere (in Cartesian coordinates) and `n` denotes the number of observations to be randomly generated.

Specifically, model number 9 corresponds to the spherical density shown in Figure 1. For instance, the evaluation of this density on the north pole $(0, 0, 1)$ and the south pole $(0, 0, -1)$ can be easily obtained by:

```
> data <- rbind(c(1, 0, 0), c(0, 0, 1))
> dspheremix(x = data, model = 9)
[1] 0.0009079986 7.0233299246
```

Output of this example with `dspheremix` is a numeric vector containing the density values on both poles. Additionally, 100 random deviates from the same model can be obtained, fixing `set.seed(1)` as in the rest of examples throughout this work, by:

```
> rspheremix(n = 100, model = 9)
            [,1]         [,2]         [,3]
  [1,]  0.254793394 -0.186993591  0.948743233
  [2,]  0.227755936  0.896600223  0.379783194
  [3,] -0.227024808  0.516581111  0.825592934
  [4,]  0.125075316  0.960536966 -0.248444967
```

Output of function `rspheremix` is a matrix of dimension $n \times 3$ where each row corresponds to the Cartesian coordinates of a point generated on the unit sphere. For this example, the output is partially shown (only four of one hundred sample points are printed).

Computation of HDRs and general level sets with HDiR

Functions `circ.hdr` and `sphere.hdr` must be considered when the objective is to compute theoretical density level sets or HDRs from a fully known circular and spherical density f, respectively. However,

[1] European-Mediterranean Seismological Centre: www.emsc-csem.org.
[2] European-Mediterranean Seismological Centre: www.emsc-csem.org.

they could be also used for exact computation or plug-in estimation of general level sets when f is any (circular or spherical) real-valued function. In particular, level sets of a theoretical regression curve could be determined.

The basic arguments of function `circ.hdr` that the user must provide are the circular (not necessarily a density) function `f` and, depending on the set to be computed (a level set or a HDR), `level` or `tau` must be indicated. It is worth to mention that `level` represents the value of t in (1) and `1-tau`, the probability coverage required for HDR computation in (2). Note that `tau` must be specified only when `f` is a density. Otherwise, fixing the probability content of the level set makes no sense. Additionally, a graphical display is generated with different plot arguments (`col`, `lty`, `shrink`, $\cdots$). If no graphical representation is required, it is enough to consider `plot.hdr=FALSE` (by default `plot.hdr=TRUE`).

If `level` is specified, the output is a list with two components: `levelset`, a matrix where each row contains the boundaries (in radians) of each connected component of the level set and `level`, the input `level` or a character indicating if the level set is equal to the empty set or the support distribution. If `tau` is provided, the output is also a list with the next components: `hdr`, a matrix where each row contains the boundaries (in radians) of each connected component of the HDR; `prob.content`, probability coverage `1-tau` and `level`, threshold of the HDR computed by numerical integration methods.

An example with the code lines in order to computing a level set (second code line) and a HDR (third code line) for the circular density represented in Figure 1 is given below. This circular density is the model 13 implemented in the package **NPCirc**. Therefore, it is necessary to install this library before executing the following code.

```
> f <- function(x){return(dcircmix(x, 13))}
> circ.hdr(f, level = 0.35)
$levelset
          [,1]      [,2]
[1,] 0.3301974 0.6698291
[2,] 2.8271189 3.1730400
[3,] 4.9089351 5.0913298
$level
[1] 0.35
> circ.hdr(f, tau = 0.5)
$hdr
          [,1]      [,2]
[1,] 0.2232764 0.7767501
[2,] 2.7201978 3.2799611
[3,] 4.8523298 5.1479351
$prob.content
[1] 0.5
$level
[1] 0.3024789
```

From the outputs obtained, some conclusions on the number of connected components can be extracted. HDR computed when $\tau = 0.5$ has exactly three connected components with boundaries fully detailed in the element `hdr` of the obtained list. Density level set with threshold 0.35 is slightly different but the information in `levelset` also shows the existence of three connected components.

As for function `sphere.hdr`, argument `f` is now a spherical real-valued function. Again, `f` may not be a density. The other basic arguments `level`, `tau` and `plot.hdr` coincide with the usage description for function `circ.hdr`. Additionally, the user can specify two parameters related to the estimated boundary or to the numerical integration possibilities on the unit sphere to calculate the HDRs threshold. In particular, `nborder` indicates the maximum number of boundary points to be represented and `tol`, the tolerance parameter used to determinate the boundary. Two extra parameters control the numerical integration procedure, when required. Argument `mesh` indicates the number of vertices on each edge of the embedded icosaedrum (reproducing the meshes in Figure 3). Possible values of this argument are 10, 20 and 40, corresponding with 2000, 8000 and 32000 triangular cells on the sphere, respectively. Quadrature formulae on the triangles are possible with different degrees, controlled by `deg`, with values ranging from 0 up to 6.

An example with the code lines in order to compute a level set (second line) and a HDR (third line) for the spherical density represented in Figure 1 is presented in what follows:

```
> f <- function(x){return(dspheremix(x, model = 9))}
> sphere.hdr(f, level = 0.1, mesh = 10, deg = 3)
> sphere.hdr(f, tau = 0.5, mesh = 10, deg = 3)
```

Outputs are similar to those presented for function `circ.hdr`. Again, `levelset` and `hdr` are matrices of

rows of points (in Cartesian coordinates) on the level set and HDR boundaries, respectively. Moreover, it is worth to mention that execution time of `sphere.hdr` is considerably higher when `tau` is set instead `level` because, in this case, threshold estimation via numerical integration methods is required.

Plug-in estimation of HDRs and general level sets with HDiR

The **HDiR** package contains the implementation of density plug-in methods in order to estimate HDRs. Furthermore, it also enables plug-in estimation of general level sets.

Basic plug-in estimation of HDRs and density level sets

Function `circ.plugin.hdr` allows to reconstruct density level sets or HDRs from the kernel estimator described in (4). The arguments `tau`, `level` and `plot.hdr` have basically the same description for function `circ.hdr`. The argument `sample` denotes a numeric vector of angles (in radians) corresponding to the sample of points $\mathcal{X}_n$. The smoothing parameter to be used for kernel density estimation is denoted through `bw`. Its value could be directly established by the user. Following Oliveira et al. (2014), it could be also chosen by using the classical functions `bw.rt`, `bw.CV`, `bw.pi` or `bw.boot` in **NPCirc** (by default `bw=bw.CV(circular(sample))` providing a cross-validation bandwidth). The previous options are designed for density estimation. An appropriate bandwidth for HDR estimation can be obtained using `circ.boot.bw`. The argument `tau.method` is a character value selecting the rule to estimate the HDRs threshold. This must be one of `"quantile"` or `"trapezoidal"`. The default option estimates the threshold using the quantile method proposed in Hyndman (1996); the second one, using the trapezoidal rule for numerical integration. The confidence for limits on HDR are established from `conf` that is a numeric probability that takes the value `conf=0.95` by default. Finally, `plot.hdrconf` is a logical string. If `plot.hdr=TRUE` and `plot.hdrconf=TRUE` (default options), the confidence region for the estimated HDR is added to the estimation graphical representation. The argument `boot` is a logical string. If `TRUE`, confidence regions are not computed. Its name is due to this option is only used by function `circ.boot.bw` for reducing the execution time. Default `boot=FALSE`.

If `level` is specified, the output is a list with four components: `levelset`, a matrix where each row contains the boundary (in radians) of a connected component of the level set or a character indicating if the HDR is equal to the empty set or the support distribution; `prob.content`, the empirical probability coverage of the set; `level` indicates the level of the level set and `bw`, the value of the smoothing parameter. If `tau` is provided, the output is a list with the next components: `hdr`, a matrix where each row contains the boundary (in radians) of a connected component of the level set; `prob.content`, the probability coverage `1-tau`; `level`, the estimated threshold; `bw`, the numeric value of the smoothing parameter used; `hdr.lo` and `hdr.hi`, HDRs corresponding to lower and upper confidence limits, respectively; `threshold.lo` and `threshold.hi` the corresponding thresholds.

For example, the circular confidence regions in Figure 2 can be obtained from the next code lines:

```
> sample <- rcircmix(500, 13)
> circ.plugin.hdr(sample, tau = 0.5, plot.hdrconf = TRUE, k = 2, col = "blue")
$hdr
          [,1]      [,2]
[1,] 0.1478027 0.6761185
[2,] 2.6761715 3.2736716
[3,] 4.9403824 5.1542246
$prob.content
[1] 0.5
$level
      50%
0.2952482
$bw
[1] 64.62809
$hdr.lo
          [,1]      [,2]
[1,] 0.1226448 0.7327238
[2,] 2.6447241 3.3114085
[3,] 4.9089351 5.1793825
$level.lo
      50%
0.2762859
$hdr.hi
         [,1]      [,2]
```

```
[1,] 0.179250 0.6320922
[2,] 2.713908 3.2422243
```

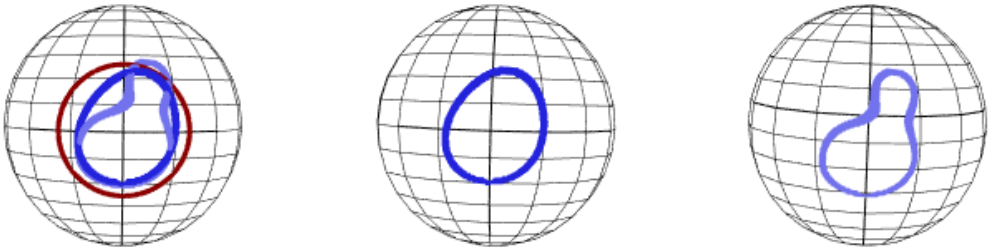

Figure 4: Theoretical (dark red colour) and estimated HDRs (left, bluish colours) from a random sample of size 500 when $\tau = 0.8$ using a cross-validation bandwidth and the specific bandwidth for spherical HDR reconstruction (left). Estimated HDRs from a random sample of size 500 using the specific bandwidth for spherical HDR reconstruction (center) and a cross-validation bandwidth (right) when $\tau = 0.8$.

```
[3,] 4.984409 5.1164877
$level.hi
      50%
0.3142105
```

Specifically, `hdr.lo` and `hdr.hi` in the output list contain the matrices whose rows correspond to the boundaries (in radians) of the connected components of lower and upper confidence regions, respectively. For this example, both regions have three connected components. Additionally, `level.lo` and `level.hi` contain the thresholds of both confidence sets.

The specific bandwidth for circular HDRs estimation described in Saavedra-Nieves and Crujeiras (2021b) can be computed from function `circ.boot.bw`. As in the previous circular functions described, the argument `sample` is a numeric vector of angles (in radians) representing $\mathcal{X}_n$ and `tau` corresponds to the probability coverage `1-tau` of the HDR to be reconstructed. The pilot smoothing parameter used is `bw`. Default `bw=bw.CV(circular(sample),upper = 100)`. As before, its value could be chosen by using the classical functions `bw.rt`, `bw.CV`, `bw.pi` or `bw.boot` in **NPCirc**. The number of bootstrap resamples is denoted by `B` (by default `B=50`) and `upper` is the numerical upper value for bounding the optimization procedure (by default `1.5bw`). The output of this function is a single numeric value corresponding to the selected smoothing parameter.

The following code lines show how to determine both bandwidths for the circular sample previously generated. Output shows that cross-validation selector takes a larger value than the proposal in Saavedra-Nieves and Crujeiras (2021b).

```
> bw.CV(sample, upper = 100); circ.boot.bw(sample, tau = 0.8, B = 2)
[1] 64.62809
[1] 37.06194
```

Function `sphere.plugin.hdr` is designed to estimate spherical HDRs or density level sets from the kernel estimator described in (4). The arguments `tau`, `level`, `plot.hdr`, `nborder`, `tol`, `mesh` and `deg` have the same description as for function `sphere.hdr`. The pilot smoothing parameter used is `bw` that, by default, is `bw="none"` selecting a cross-validation bandwidth. Although other options are possible. For instance, `bw` can be a numeric value o also `bw="rot"` allows to consider the rule of thumb suggested by García-Portugués (2013). The value of `bw` could be also selected directly by the user. The argument `ngrid` sets the resolution of the density calculation (by default `ngrid=500`).

If `level` is provided, the output is also a list with four components: `levelset`, a matrix of rows of points (on the HDR boundary; `prob.content`, the empirical probability coverage of the set `1-tau`; `level`, the level of the HDR and `bw`, the value of the smoothing parameter. If `tau` is an input, the output of `sphere.plugin.hdr` is a list with the following components: `hdr`, a matrix of rows of points on the HDR boundary; `prob.content`, probability coverage `1-tau` and `level`, threshold or level associated to the probability content `1-tau`. The threshold f_τ is computed through the algorithm proposed in Hyndman (1996). Numerical integration is not considered here in order to reduce the computation time.

The spherical HDRs estimators in Figure 2 can be reproduced through the next code lines:

```
> sample <- rspheremix(500, model = 9)
> sphere.plugin.hdr(sample, tau = 0.5, nborder = 2000)
```

The first specific bandwidth for spherical HDRs estimation described in Saavedra-Nieves and

Crujeiras (2021b) can be computed from function `sphere.boot.bw`. As in the previous spherical functions described, the argument `sample` is a matrix whose rows represent points on the unit sphere (in Cartesian coordinates) and `tau` corresponds to the probability coverage `1-tau` of the HDR to be reconstructed. The pilot smoothing parameter `bw` (default `bw="none"`) is chosen using cross-validation, although it may be set to a numeric value or `bw="rot"`, allowing to select the rule of thumb suggested by García-Portugués (2013). The argument `B` denotes again the number of bootstrap resamples that (default `B=50`) and `upper` is the numerical upper value for bounding the optimization procedure (default `1.5bw`). The output of this function is a single numeric value corresponding to the selected smoothing parameter.

The following code lines contain a simulated example where the cross-validation bandwidth and the proposal in Saavedra-Nieves and Crujeiras (2021b) provide HDR estimations which look quite different for the spherical model 8 in **HDiR**. Figure 4 shows the graphical representations of the theoretical HDR to be estimated when $\tau = 0.8$ (dark red colour) and the corresponding reconstructions (bluish colours) obtained from a random sample of size 500. In this case, the specific bandwidth for spherical HDRs reconstruction takes the value 0.28 while the cross-validation bandwidth is equal to 0.20.

```
> sample <- rspheremix(500, model = 8)
> bw.boot <- sphere.boot.bw(sample, bw = "rot", tau = 0.8, B = 2)
> sphere.plugin.hdr(sample, bw = bw.boot, tau = 0.8)
> sphere.plugin.hdr(sample, bw = "none", tau = 0.8)
```

Finally, it is important to note that function `sphere.plugin.hdr` for reconstructing spherical HDR's calls `vmf.kerncontour` in package **Directional** to compute the density on a grid on the sphere. Most of the computational work in this function is in estimating the density using `vmf.kerncontour`. Hence, the speed of this function depends largely on the speed of `vmf.kerncontour`. A similar situation occurs for function `sphere.boot.bw` where function `sphere.plugin.hdr` is called $(B+1)$ times where B indicates the number of bootstrap resamples.

Plug-in estimation of HDRs and density level sets from an arbitrary density estimator

Density estimators different from the one introduced in (4) could be naturally considered for plug-in estimation of HDRs or level sets. Functions `circ.hdr` and `sphere.hdr` in package **HDiR** allow to consider this option in the circular and spherical settings, respectively.

Next, an example with the code lines in order to determine a spherical HDR plug-in reconstruction (sixth line) from the kernel density estimator in Bai et al. (1989) with uniform kernel is shown.

```
> f <- function(x){
    sample <- rspheremix(500, model = 3)
    return(kde_dir(x, data = sample, h = 0.4,
     L = function(x) dunif(x)))
    }
> sphere.hdr(f, level = 0.3)
$levelset
               [,1]         [,2]      [,3]
  [1,]  0.3587511132 -0.159961736 0.9196249
  [2,] -0.4523490796  0.077542650 0.8884635
  [3,] -0.4588831000  0.060463844 0.8864369
  [4,]  0.2455354599 -0.291602658 0.9244892
$level
[1] 0.3
```

An spherical density estimator with uniform kernel is available in package **DirStats**. Before level set plug-in estimation, it is necessary to install this library in order to define the kernel estimator, in this example, from a sample of size 500 of model 3 in **HDiR** (lines from 1 to 5). The output contains a matrix of points on the boundary of the plug-in estimator in `levelset`. Note that only the first four points are printed in the example. The value of the threshold 0.3 considered for reconstruction is also shown in `level`.

Furthermore, if the considered density estimator for plug-in estimation is also a density function, argument `tau` in `circ.hdr` and `sphere.hdr` could be used.

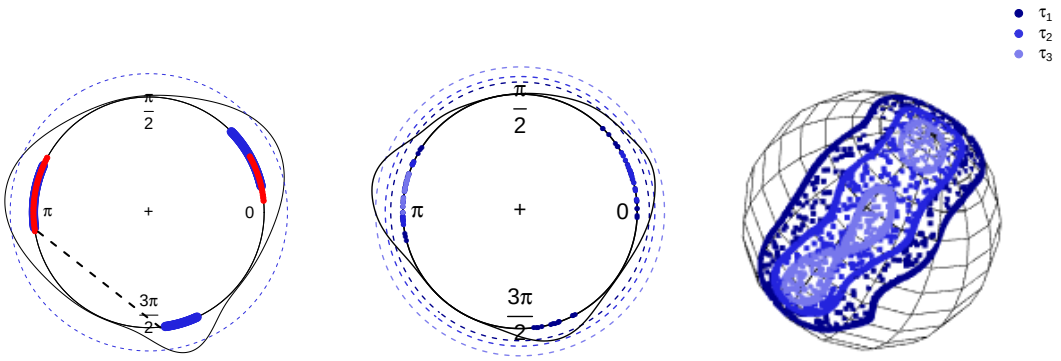

Figure 5: In the first column, the black dotted line represents the Hausdorff distance between $\partial L(f_\tau)$ (blue colour) and $\partial \hat{L}(\hat{f}_\tau)$ (red colour) for the circular density shown in Figure 1 when $\tau = 0.5$. In the second and third columns, scatterplots showing $\hat{L}(\hat{f}_{\tau_i})$ $(i = 1, 2, 3)$ for the circular and spherical densities contained in Figure 1 when $\tau_1 = 0.2$, $\tau_2 = 0.5$ and $\tau_3 = 0.8$. The circular scatterplot was computed from $\mathcal{X}_{100}$ and the spherical scatterplot from $\mathcal{X}_{1000}$.

Plug-in estimation of general level sets

A generalisation of the approach in Cuevas et al. (2006), for general level sets, to the directional setting can be performed in practice with **HDiR**. Again, functions `circ.hdr` and `sphere.hdr` address this problem for circular and sperical data, respectively.

Next, an example with the code lines in order to obtain the plug-in estimator of a regression curve (eighth line) with circular explanatory (`x`) variable and linear response (`y`). In this case, the regression curve is estimated through the Nadaraya-Watson estimator implemented in **NPCirc**. Here, it is computed from a sample of size 100 of variables `x` and `y` (lines from 1 to 7).

```
> f <- function(t){
    n <- 100
    x <- runif(n, 0, 2*pi)
    y <- sin(x)+0.5*rnorm(n)
    return(kern.reg.circ.lin(circular(x), y, t, bw = 10, method = "NW")$y)
    }
> circ.hdr(f, level = 0.5, plot.hdr = FALSE)
$levelset
          [,1]     [,2]
[1,] 0.4748553 2.757935
$level
[1] 0.5
```

Output in `levelset` contains the boundary (in radians) of the only connected component for the reconstructed regression level set.

Exploring data with HDiR

This section introduces a brief background on the design of two exploratory tools included in **HDiR**: distances between sets and circular/spherical scatterplots.

Distances between sets are a useful tool when the target is the reconstruction of a set. In particular, the Hausdorff distance can be seen as a suitable error criterion also in the directional setting. Additionally, it could be also used for measuring the distances between modes or clusters of two different populations. Figure 5 (first column) represents, through a black dashed line, the Hausdorff distance between $\partial L(f_\tau)$ (blue colour) and $\partial \hat{L}(\hat{f}_\tau)$ (red colour) for the circular density shown in Figure 1 when $\tau = 0.5$. Note that the maximum value of this error criterion is 2, the diameter of the unit circle. In this example, the Hausdorff estimation error that is equal to 1.38 is remarkably high.

Function `circ.distances` computes the Euclidean and Hausdorff distances between two sets of points in S^1. Its inputs are `x` and `y`, two numeric vectors of angles (in radians) determining both sets of points. The output is a list with two components: `dE`, a numeric value corresponding to the Euclidean distance, and `dH`, another numeric value corresponding to the Hausdorff distance.

Specifically, if x and y correspond to two HDRs boundaries, this function returns the distances between the circular HDRs frontiers. In particular, for the example in Figure 5 (left), the distances between $\partial L(\tau)$ and $\partial \hat{L}(\hat{f}_\tau)$ can be computed from the next code lines:

```
> sample <- rcircmix(100, 13)
> f <- function(x){return(dcircmix(x, 13))}
> circ.distances(as.numeric(circ.hdr(f, tau = 0.5)$hdr),
    + as.numeric(circ.plugin.hdr(sample, tau = 0.5)$hdr))
$dE
[1] 0.04402277
$dH
[1] 1.37933
```

The results obtained show that the Euclidean distance is considerably smaller than the Hausdorff distance that, as we mention before, takes the value 1.38.

Function `sphere.distances` also determines the Euclidean and Hausdorff distances but, in this case, between two sets of points on S^2. Now, the inputs x and y are two matrices whose rows represent points on the unit sphere (in Cartesian coordinates). The output of this function has the same organization as the output of `circ.distances` and it also allows to compute distances between spherical HDRs frontiers.

Distances between $\partial \hat{L}(\hat{f}_{\tau_2})$ and $\partial \hat{L}(\hat{f}_{\tau_3})$ represented in Figure 5 (right) can be computed from the next code lines:

```
> sample = rspheremix(1000, model = 9)
> x <- sphere.plugin.hdr(sample, tau = 0.8, plot.hdr = FALSE)$hdr
> y <- sphere.plugin.hdr(sample, tau = 0.5, plot.hdr = FALSE)$hdr
> sphere.distances(x, y)
$dE
[1] 0.08600028
$dH
[1] 0.258705
```

The performance of the specific bandwidth for HDR estimation introduced in Saavedra-Nieves and Crujeiras (2021b) can be also illustrated through the consideration of the Hausdorff distance in the example shown in Figure 4. Specifically, the value of the Hausdorff distance between the theoretical HDR and the reconstruction computed from the bandwidth proposed in Saavedra-Nieves and Crujeiras (2021b) is 0.20. However, the Hausdorff distance increases considerably, taking the value 0.36, when it measures the discrepancies between the theoretical HDR and the corresponding estimator obtained from a cross-validation approach.

Additionally, scatterplots are useful to identify the estimated directional HDRs in which sample points fall. This graphical tool is computed as follows. Given several values $\tau_1, \cdots, \tau_k \in (0,1)$ $(k \geq 1)$ and a random sample of points $\mathcal{X}_n$, the estimated HDRs $\hat{L}(\hat{f}_{\tau_1}), \cdots, \hat{L}(\hat{f}_{\tau_k})$ are represented using different colours jointly with the subset of sample points belonging to each of them. Figure 5 (second and third columns) displays the scatterplots for $\tau_1 = 0.2$, $\tau_2 = 0.5$ and $\tau_3 = 0.8$ for the circular and the spherical densities shown in Figure 1. They were calculated from random samples of sizes $n = 100$ and $n = 1000$, respectively.

Function `circ.scatterplot` produces a circular scatterplot with points coloured according to the HDRs in which they fall. Apart from the argument `tau` that represents a numeric vector of probabilities and `plot.density` that is a logical string indicating if the kernel density estimator is added to the scatterplot (default `plot.density=TRUE`), the other inputs (`sample`, `bw` and `tau.method`) have the same description for circular functions. The output is a scatterplot and also a list where the number of components is equal to the number of estimated HDR or, equivalently, to the length of `tau` vector. Each component contains the sample points in each HDR from the smallest value of `tau` to the largest one.

Next code lines allow to obtain a circular scatterplot computed from a circular sample of size 100 as the shown in Figure 5 (second column).

```
> sample<- rcircmix(100, model = 13)
> circ.scatterplot(sample, tau = c(0.2, 0.5, 0.8))
```

Spherical scatterplots can be represented from function `sphere.scatterplot`. Again, apart from `tau` that is a vector of probabilities, the description of the remaining parameters coincides with the rest of spherical functions. The output provides a scatterplot and, as in the circular case, a list where the number of components is equal to the number of estimated HDR containing the corresponding sample points from the smallest value of `tau` to the biggest one.

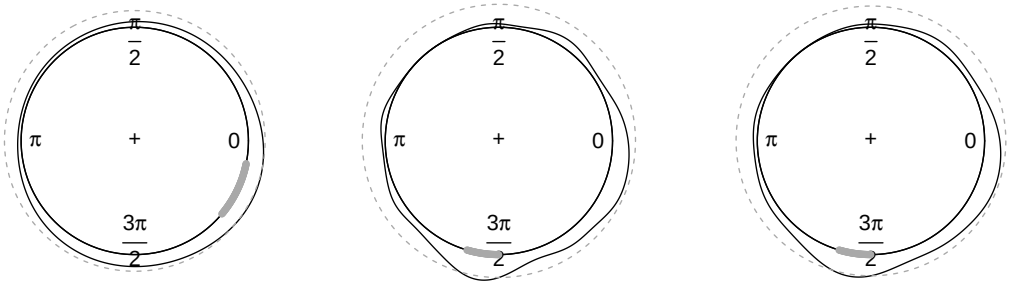

Figure 6: Plug-in estimations of HDRs (gray colour) with cross-validation bandwidth, when $\tau = 0.8$, for females (left) and males (center) of the species Talorchestia Brito when the orientation is registered in morning during October. Plug-in estimation of HDR (gray colour) with specific bandwidth h_*, when $\tau = 0.8$, for males (right) of the species Talorchestia Brito when the orientation is registered in morning during October.

As an illustration, the spherical scatterplot shown in Figure 5 (third column) could be computed from the next code lines:

```
> sample <- rspheremix(1000, model = 9)
> sphere.scatterplot(sample, tau = c(0.2, 0.5, 0.8))
```

Real data analysis with HDiR

Datasets `sandhoppers` and `earthquakes` included in **HDiR** are used next to illustrate briefly the usage of the set of functions previously described in the circular and spherical settings, respectively.

Figure 6 shows the estimated HDRs established in (3), when $\tau = 0.8$, for female (left) and male sandhoppers (right) of the species Talorchestia Brito when the orientation is registered in the morning during October. The largest modes of both distributions are located in completely different directions, indicating that variable sex is a factor with influence on the sandhoppers behavior. The code lines used are presented:

```
> data(sandhoppers)
> attach(sandhoppers)
> britoF <- angle[(species == "brito")&(time == "morning")&(sex == "F")
        + &(month == "October")]
> circ.plugin.hdr(sample = britoF, tau = 0.8, plot.hdrconf = FALSE)
> britoM <- angle[(species == "brito")&(time == "morning")&(sex == "M")
    + &(month == "October")]
> circ.plugin.hdr(sample = britoM, tau = 0.8, plot.hdrconf = FALSE)
```

According to Figure 6, no remarkable differences exist between the HDRs reconstructions for males using a cross-validation bandwidth (center) and the proposal h_* in Saavedra-Nieves and Crujeiras (2021b) (right). However, these smoothing parameters are quite different, taking values 33.86 and 19.47, respectively. For the subset of females, differences between smoothing parameters are smaller (5.78 and 3.39, respectively). Next, code lines show how to determine both bandwidths for the group of males (fist line) and females (second line).

```
> bw.CV(britoM); circ.boot.bw(britoM, tau = 0.8)
> bw.CV(britoF); circ.boot.bw(britoF, tau = 0.8)
```

As an example with the dataset `earthquakes` in Figure 7, we show the estimated HDR defined in (3) for $\tau = 0.8$. The largest mode of the earthquakes distribution is located in Southeast Europe. Note that it is necessary to install the packages **Directional**, **ggplot2**, **maps** and **mapproj** previously to obtain this figure.

```
> data(earthquakes)
> hdr <- as.data.frame(euclid.inv(sphere.plugin.hdr(euclid(earthquakes), tau = 0.8,
        + plot.hdr = FALSE)$hdr))
> world <- map_data("world")
> g.earthquakes <- ggplot()+
> geom_map(data = world, map = world, mapping = aes(map_id = region),
        +  color = "grey90", fill = "grey80")+
```

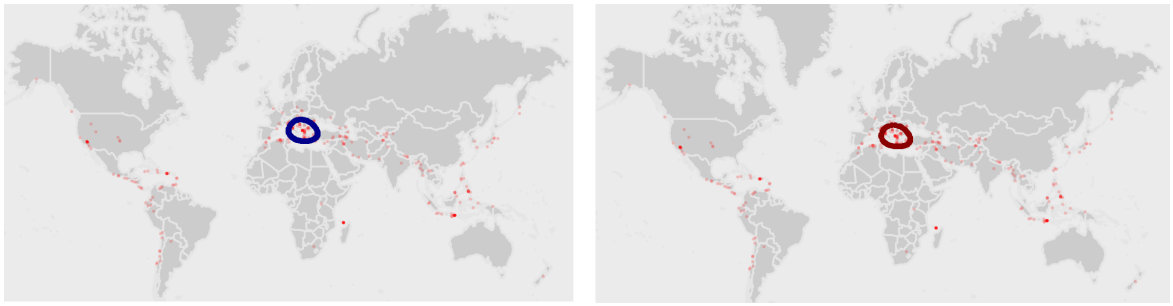

Figure 7: Contours of plug-in HDRs for $\tau = 0.8$ obtained from the sample of world earthquakes registered between October 2004 and April 2020 with cross-validation bandwidth (left) and with the specific bandwidth for spherical HDRs reconstruction (right).

```
> geom_point(data = earthquakes, mapping = aes(x = Longitude,
        +   y = Latitude), color = "red", alpha = 0.2, size = 0.75, stroke = 0)+
> geom_point(data = hdr, mapping = aes(x = Long, y = Lat),
        +          color = "darkblue", size = 1)+
> scale_y_continuous(breaks = NULL, limits = c(-90, 90))+
> scale_x_continuous(breaks = NULL, limits = c(-180, 180))+
> coord_map("mercator")
> g.earthquakes
```

The value of the bandwidth proposed in Saavedra-Nieves and Crujeiras (2021b) for `earthquakes` dataset with `tau=0.8` and `B=5` bootstrap resamples is 0.09 and it can be obtained from the next code line. In this particular case, Figure 7 shows that there is not a large differences between the HDRs reconstructed from cross-validation bandwidth (left) and the proposal in Saavedra-Nieves and Crujeiras (2021b) (right).

```
> sphere.boot.bw(euclid(earthquakes), tau = 0.8, B = 5)
```

Once the HDRs estimation has been performed for different values of τ, Euclidean and Hausdorff distances between the blue and red contours in Figure 7 are useful to analyse differences between them. For the previous example, distances can be computed from the following code lines. Note that the value of the bandwidth in Saavedra-Nieves and Crujeiras (2021b) has been directly inserted as an argument in the fourth line. Values obtained for Euclidean and Hausdorff distances are 0 and 0.02, respectively.

```
> hdr1 <- sphere.plugin.hdr(euclid(earthquakes), tau = 0.8, plot.hdr = FALSE)$hdr
> hdr2 <- sphere.plugin.hdr(euclid(earthquakes), bw = 0.09, tau = 0.8,
        + plot.hdr = FALSE)$hdr
> sphere.distances(hdr1, hdr2)
```

Apart from distances between HDRs, scatterplots are another powerful exploratory tool implemented in **HDiR**. For the `sandhoppers` dataset, Figure 8 shows the circular scatterplots for $\tau = 0.2, 0.5$ and 0.8 for females (left) and males (center) of the species Talorchestia Brito when the orientation is registered in the morning during October when $\tau = 0.2, 0.5$ and 0.8. They can be obtained from the following code:

```
> circ.scatterplot(britoF, tau = c(0.2, 0.5, 0.8))
> circ.scatterplot(britoM, tau = c(0.2, 0.5, 0.8))
```

Spherical scatterplots for `earthquakes` dataset when $\tau = 0.2$, $\tau = 0.5$ and $\tau = 0.8$ can be computed from the following code line. The function `euclid` allows to transforms the data to geographical coordinates (longitude and latitude) on Cartesian coordinates. Remark that the smoothing parameter is selected by using the rule of thumb proposed in García-Portugués (2013).

```
>  sphere.scatterplot(euclid(earthquakes), tau = c(0.2, 0.5, 0.8), bw = "rot",
    + nborder = 1500)
```

4 Discussion

HDiR has been mainly developed for facilitating the reconstruction of directional (circular and spherical) HDRs and density level sets, following a nonparametric plug-in approach. However, it

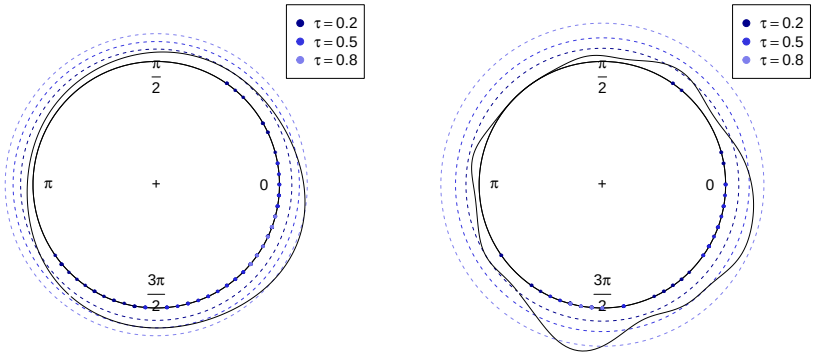

Figure 8: Circular scatterplots computed for $\tau = 0.2$, 0.5 and 0.8 from samples of females (left) and males (center) of the species Talorchestia Brito when the orientation is registered in morning during October.

also allows to solve the computation and the plug-in estimation of level sets for general real-valued functions, such as a regression curve. As consequence, plug-in reconstruction of HDRs could be performed by considering a different density estimator than the one implemented by default in **HDiR**.

The implemented tools are accessible for the scientific community, enabling their usage in practical problems such as the exploration of modes or the approximation of the distribution *effective support*. As previously noted, level set computation is also useful for determining distribution clusters, a task that can be accomplished by the identification of the connected components from a plug-in level set estimator.

Up to the authors' knowledge, **HDiR** is the only statistical package that allows to estimate (circular and spherical) HDRs and general level sets. For HDRs reconstruction, **HDiR** also implements the first specific selector for HDRs estimation in this context, proposed in Saavedra-Nieves and Crujeiras (2021b). Additionally, it offers graphical exploratory tools such as HDRs scatterplots that allow to visualize HDRs of a distribution taking into account different probability contents. Similarities or discrepancies between them could be measured through the Hausdorff distance also implemented in **HDiR**.

Future extensions of the **HDiR** package could include the estimation of level sets and HDRs in other supports, involving a circular or a spherical component, such as the torus or the cylinder. In addition, new specific bandwidths for HDR estimation could be implemented. A variety of bandwidths selectors emerge from the consideration of different distances in (5). Finally, cluster definition in Hartigan (1975) deserves to be exploited in the directional setting, for instance, by implementing cluster trees for hyperspherical data.

5 Acknowledgments

P. Saavedra-Nieves and R.M. Crujeiras acknowledge the financial support of Ministerio de Ciencia e Innovación of the Spanish government under grants PID2020-118101GB-I00 and PID2020-116587GB-I00 and ERDF. Authors also thank Prof. Felicita Scapini for providing the sandhoppers data (collected under the support of the European Project ERB ICI8-CT98-0270), Prof. Andrés Prieto for his help with spherical numerical integration. The authors also acknowlege the constructive comments of the AE and the reviewer, which have improved the contents of the paper and the package.

Bibliography

C. Agostinelli and U. Lund. R package 'circular': Circular statistics. 2013. URL `https://r-forge.r-project.org/projects/circular`. R package version 0.4-7. [p126]

A. Azzalini and N. Torelli. Clustering via nonparametric density estimation. *Statistics and Computing*, 17(1):71–80, 2007. [p121, 122]

A. Azzalini, G. Menardi, and T. Rosolin. Package pdfcluster: Cluster analysis via nonparametric density estimation. *J. Stat. Softw*, 11:1–26, 2014. [p126]

Z. Bai, C. R. Rao, and L. Zhao. Kernel estimators of density function of directional data. In *Multivariate Statistics and Probability*, pages 24–39. Elsevier, 1989. [p123, 133]

A. Baíllo. Total error in a plug-in estimator of level sets. *Statistics & probability letters*, 65(4):411–417, 2003. [p123]

A. Baíllo and A. Cuevas. Parametric versus nonparametric tolerance regions in detection problems. *Computational Statistics*, 21(3):523–536, 2006. [p122]

S. Barragán, M. A. Fernández, C. Rueda, and S. D. Peddada. isocir: An R package for constrained inference using isotonic regression for circular data, with an application to cell biology. *Journal of Statistical Software*, 54(4), 2013. [p126]

A. Bowman and A. Azzalini. Package 'sm'. 2018. URL https://CRAN.R-project.org/package=sm. R package version 2.2-5.6. [p126]

G. Box and G. Tiao. Bayesian inference in statistical analysis, reading. *Mass.: Addison-Wesley*, 1973. [p121]

A. Casa, J. E. Chacón, and G. Menardi. Modal clustering asymptotics with applications to bandwidth selection. *Electronic Journal of Statistics*, 14(1):835–856, 2020. [p122]

S.-J. Chang-Chien, M.-S. Yang, and W.-L. Hung. Mean shift-based clustering for directional data. In *Third International Workshop on Advanced Computational Intelligence*, pages 367–372. IEEE, 2010. [p121]

Y.-C. Chen, C. R. Genovese, and L. Wasserman. Density level sets: Asymptotics, inference, and visualization. *Journal of the American Statistical Association*, 112(520):1684–1696, 2017. [p123]

Y. Cheng and S. Ray. Parallel and hierarchical mode association clustering with an R package modalclust. *Open Journal of Statistics*, 4(10):826–836, 2014. [p126]

A. Cholaquidis, R. Fraiman, and L. Moreno. Level set and density estimation on manifolds. *Journal of Multivariate Analysis*, 189:104925, 2022. [p121, 123]

D. R. Cox and D. V. Hinkley. *Theoretical statistics*. CRC Press, 1979. [p125]

A. Cuevas, W. González-Manteiga, and A. Rodríguez-Casal. Plug-in estimation of general level sets. *Australian & New Zealand Journal of Statistics*, 48(1):7–19, 2006. [p121, 122, 123, 134]

M. Di Marzio, A. Panzera, and C. C. Taylor. Kernel density estimation on the torus. *Journal of Statistical Planning and Inference*, 141(6):2156–2173, 2011. [p123]

C. R. Doss and G. Weng. Bandwidth selection for kernel density estimators of multivariate level sets and highest density regions. *Electronic Journal of Statistics*, 12(2):4313–4376, 2018. [p126]

T. Duong. ks: Kernel density estimation and kernel discriminant analysis for multivariate data in R. *Journal of Statistical Software*, 21(7):1–16, 2007. [p126]

J. Einbeck and L. Evers. Lpcm: Local principal curve methods. 2019. URL https://CRAN.R-project.org/package=LPCM. R package version 0.46-3. [p126]

J. Fernández-Durán and M. Gregorio-Domínguez. Circnntsr: An R package for the statistical analysis of circular data using nonnegative trigonometric sums (nnts) models. 2013. URL https://CRAN.R-project.org/package=CircNNTSR. R package version 2.2-1. [p126]

E. García-Portugués. Exact risk improvement of bandwidth selectors for kernel density estimation with directional data. *Electronic Journal of Statistics*, 7:1655–1685, 2013. [p123, 126, 132, 133, 137]

E. García-Portugués. DirStats: Nonparametric methods for directional data. 2021. URL https://CRAN.R-project.org/package=DirStats. R package version 0.1.7. [p126]

P. Hall, G. Watson, and J. Cabrera. Kernel density estimation with spherical data. *Biometrika*, 74(4): 751–762, 1987. [p123, 126]

J. A. Hartigan. *Clustering algorithms*. John Wiley & Sons, Inc., 1975. [p121, 126, 138]

L. Holmström, K. Karttunen, and J. Klemelä. Estimation of level set trees using adaptive partitions. *Computational Statistics*, 32(3):1139–1163, 2017. [p126]

K. Hornik and B. Grün. movmf: an R package for fitting mixtures of von mises-fisher distributions. *Journal of Statistical Software*, 58(10):1–31, 2014. [p126]

R. J. Hyndman. Computing and graphing highest density regions. *The American Statistician*, 50(2): 120–126, 1996. [p122, 125, 127, 130, 132]

R. J. Hyndman, J. Einbeck, and M. Wand. hdrcde. 2018. URL https://CRAN.R-project.org/package=hdrcde. R package version 3.4. [p126]

S. R. Jammalamadaka and A. Sengupta. *Topics in circular statistics*, volume 5. world scientific, 2001. [p126]

J. Klemelä. Algorithms for manipulation of level sets of nonparametric density estimates. *Computational Statistics*, 20(2):349–368, 2005. [p126]

J. Klemelä. Visualization of multivariate density estimates with shape trees. *Journal of Computational and Graphical Statistics*, 15(2):372–397, 2006. [p126]

J. Klemelä. Mode trees for multivariate data. *Journal of Computational and Graphical Statistics*, 17(4): 860–869, 2008. [p126]

J. Klemelä. denpro. 2015. URL https://CRAN.R-project.org/package=denpro. R package version 0.9.2. [p126]

J. S. Klemelä. *Smoothing of multivariate data: density estimation and visualization*. John Wiley & Sons, 2009. [p126]

J. Li, S. Ray, and B. G. Lindsay. A nonparametric statistical approach to clustering via mode identification. *Journal of Machine Learning Research*, 8(8), 2007. [p126]

U. Lund and C. Agostinelli. Circstats. 2012. URL https://CRAN.R-project.org/package=CircStats. R package version 0.2-6. [p126]

E. Mammen and W. Polonik. Confidence regions for level sets. *Journal of Multivariate Analysis*, 122: 202–214, 2013. [p123]

G. M. Marchetti and F. Scapini. Use of multiple regression models in the study of sandhopper orientation under natural conditions. *Estuarine, Coastal and Shelf Science*, 58:207–215, 2003. [p127]

D. M. Mason and W. Polonik. Asymptotic normality of plug-in level set estimates. *The Annals of Applied Probability*, 19(3):1108–1142, 2009. [p123]

G. Menardi. A review on modal clustering. *International Statistical Review*, 84(3):413–433, 2016. [p121]

M. Oliveira, R. M. Crujeiras, and A. Rodríguez-Casal. A plug-in rule for bandwidth selection in circular density estimation. *Computational Statistics & Data Analysis*, 56(12):3898–3908, 2012. [p123]

M. Oliveira, R. M. Crujeiras, and A. Rodríguez-Casal. Npcirc: An r package for nonparametric circular methods. *Journal of Statistical Software, Articles*, 61(9):1–26, 2014. [p126, 130]

B. Pelletier. Kernel density estimation on riemannian manifolds. *Statistics & probability letters*, 73(3): 297–304, 2005. [p123]

P. Rigollet, R. Vert, et al. Optimal rates for plug-in estimators of density level sets. *Bernoulli*, 15(4): 1154–1178, 2009. [p123]

P. Saavedra-Nieves and R. M. Crujeiras. HDiR. 2021a. URL https://CRAN.R-project.org/package=HDiR. R package version 1.1.2. [p125]

P. Saavedra-Nieves and R. M. Crujeiras. Nonparametric estimation of directional highest density regions. *Advances in Data Analysis and Classification*, pages 1–36, 2021b. [p122, 123, 124, 125, 126, 127, 128, 132, 133, 135, 136, 137, 138]

R. Samworth and M. Wand. Asymptotics and optimal bandwidth selection for highest density region estimation. *The Annals of Statistics*, 38(3):1767–1792, 2010. [p122, 125, 126]

F. Scapini, A. Aloia, M. F. Bouslama, L. Chelazzi, I. Colombini, M. ElGtari, M. Fallaci, and G. M. Marchetti. Multiple regression analysis of the sources of variation in orientation of two sympatric sandhoppers, talitrus saltator and talorchestia brito, from an exposed mediterranean beach. *Behavioral Ecology and Sociobiology*, 51(5):403–414, 2002. [p127]

G. Strang and G. J. Fix. *An analysis of the finite element method*. Prentice-hall, 1973. [p124]

C. C. Taylor. Automatic bandwidth selection for circular density estimation. *Computational Statistics & Data Analysis*, 52(7):3493–3500, 2008. [p123]

M. Tsagris, G. Athineou, and A. Sajib. Directional. 2017. URL https://CRAN.R-project.org/package=Directional. R package version 5.6. [p126]

A. B. Tsybakov. On nonparametric estimation of density level sets. *The Annals of Statistics*, 25(3): 948–969, 1997. [p123]

M.-S. Yang, S.-J. Chang-Chien, and H.-C. Kuo. On mean shift clustering for directional data on a hypersphere. In *International Conference on Artificial Intelligence and Soft Computing*, pages 809–818. Springer, 2014. [p121]

Paula Saavedra-Nieves
CITMAga, Galician Centre for Mathematical Research and Technology
Universidade de Santiago de Compostela
Facultade de Matemáticas
Lope Gómez de Marzoa, s/n
Campus sur, 15782
Santiago de Compostela, Spain
paula.saavedra@usc.es

Rosa M. Crujeiras
CITMAga, Galician Centre for Mathematical Research and Technology
Universidade de Santiago de Compostela
Facultade de Matemáticas
Lope Gómez de Marzoa, s/n
Campus sur, 15782
Santiago de Compostela, Spain
rosa.crujeiras@usc.es