Robust Estimation for Circular Data using R

Claudio Agostinelli
claudio@unive.it

Dipartimento di Statistica
Università Ca' Foscari di Venezia
San Giobbe, Cannaregio 873, Venezia
Tel. 041 2347446, Fax. 041 2347444
http://www.dst.unive.it/~claudio

16 June 2006

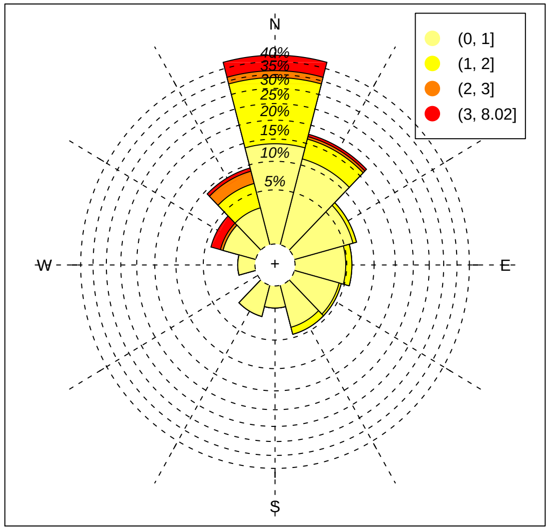
Outline

Circular Data

Circular data arises in many diverse scientific fields like in Natural, Physical, Medical and Social Sciences where some or all the measurements are directions. The two main ways correspond to the two principal circular measuring instruments:

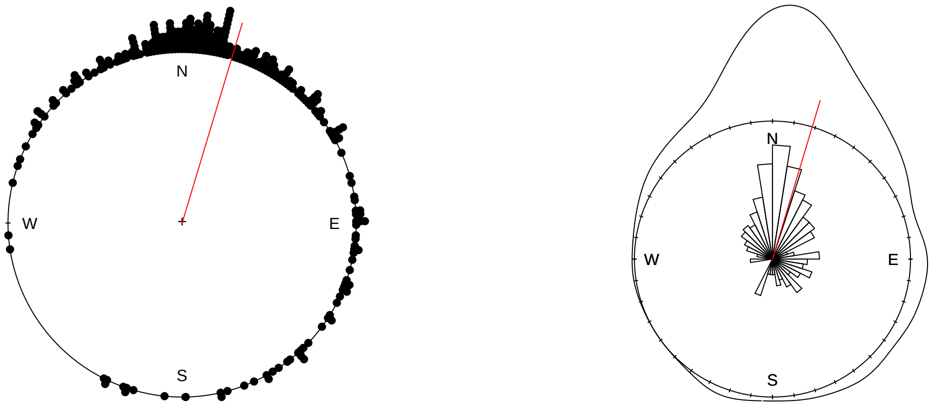
- the *compass*
  - e.g. wind directions and directions of migrating animals.
- the *clock*.
  - e.g. arrival times (on a 24-hour clock) of subjects.

Because of the nature of the circular observations, the analysis (descriptive and/or inferential) cannot be carried out with standard methods for observations on Euclidean space.
Some references: Mardia and Jupp [2000], Jammalamadaka and SenGupta [2001] and Jupp and Mardia [1989].



Wind rose of the dataset from Col de la Roa (Italy) meteorological station in March and April 2001 between 3.00 and 4.00 o'clock (n=310).

R code



Stack Circular Plot (Left) and estimated density (Right, together with a Rose Diagram) of the Wind direction dataset. ‣ R code

Models for Circular data

Some examples are:

- Von Mises (Normal Circular) distribution:

$$m(x;\mu,\kappa) = \frac{1}{2\pi I_0(x)} \exp(\kappa cos(x-\mu)), \qquad \kappa \geq 0$$

- Wrapped Normal distribution:

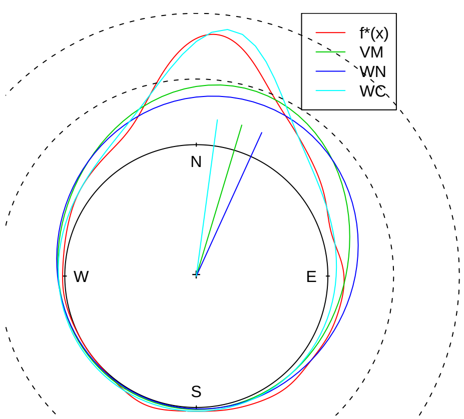
$$m(x;\mu,\rho) = \frac{1}{2\pi}\left(1 + 2\sum_{p=1}^{\infty} \rho^{p^2} \cos p(x-\mu)\right), \qquad \rho = \exp\left(\frac{-\sigma^2}{2}\right)$$

- Wrapped Cauchy distribution:

$$m(x;\mu,\rho) = \frac{1}{2\pi}\frac{1-\rho^2}{1+\rho^2-2\rho cos(x-\mu)}, \qquad \rho = \exp(-\sigma)$$

- Cardioid distribution:

$$m(x;\mu,\rho) = \frac{1+2\rho cos(x-\mu)}{2\pi}, \qquad -\frac{1}{2} < \rho < \frac{1}{2}$$



MLE estimators for the Wind direction dataset using

- Von Mises (VM, $\hat{\mu} = 16.74$, $\hat{\kappa} = 1.76$);
- Wrapped Normal (WN, $\hat{\mu} = 24.487$, $\hat{\rho} = 0.603$);
- Wrapped Cauchy (WC, $\hat{\mu} = 7.662$, $\hat{\rho} = 0.697$). ‣ R code.

Robust Statistics

Robust Statistics is a discipline which deals with the inference problem in the case of misspecification of the model with respect to the distribution of the data. A "robust" estimator is able to well describe the parameters, or some features, of the "majority" (or "bulk") of the data. And to quote:

- Hampel et al. [1986], pag. 56: "whereas in classical statistics the model has to fit all the data, in robust statistics it *may* be enough that it fit the majority of the data, the remainder being regarded as outliers";

Some references: Collett [1980], Lenth [1981], Upton [1993] and SenGupta and Laha [2001]. A survey is He [1992].

Outliers in Circular data

- In Euclidean setting the outliers are defined as:
  - Barnett et al. [1994], pag. 7: "... an outliers in a set of data to be an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data";
  - Hawkins [1980]: an outlier is "an observation that deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism";

  The definition is based on some **"geometric" distance** between observations.
- In Circular setting: the sample space is bounded and the parametric space is often bounded too (e.g. the parametric space of the mean direction in the Von Mises distribution is every interval of length 2π, in general, $[0, 2\pi)$).

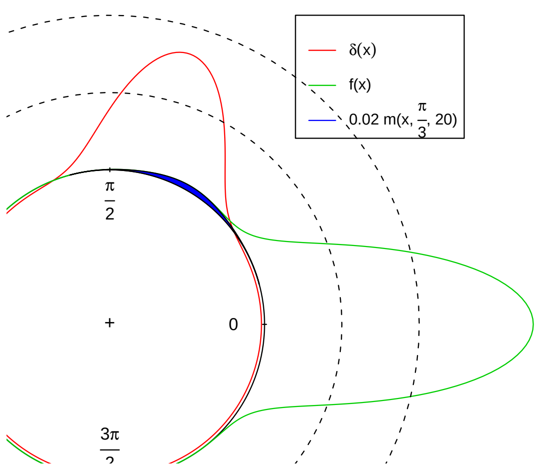
Outliers in Circular Data: Pearson Residuals

In our approach, outliers are observations that are highly unlikely to occur under the assumed model [see Markatou et al., 1995].

This definition is well adapted in the circle since it is based on a **"probabilistic" distance**. One way to measure this discrepancy is to use the Pearson Residuals [Lindsay, 1994] defined as follows

$$\delta(x, \theta, f^*) = \frac{f^*(x)}{m^*(x;\theta)} - 1$$

where $f^*(x)$ is a non parametric density estimator based on the data and $m^*(x;\theta)$ is a smoothed version of the density of the model. Note that $\delta(x, \theta, f^*) = \delta(x \mod (2\pi), \theta, f^*)$.



Pearson Residuals ($\delta(x; \theta = (0, 5), f(x))$) for $f(x) = 0.98m(x; 0, 5) + 0.02m(x; \pi/3, 20)$ where $m(x; \mu, \kappa)$ is the density of a Von Mises distribution. R code

Minimum Distance Estimators

For continuous models the Power Divergence Measure [Cressie and Read, 1988] between the densities $f^*(x)$, $m^*(x, \theta)$ is

$$\int_0^{2\pi} \frac{\left(\frac{f^*(x)}{m(x;\theta)}\right)^{\alpha+1}}{\alpha\ (\alpha+1)}\ m(x;\theta)\ dx$$

or using the Pearson Residual function (in the unsmoothed model) $\delta(x; \theta, f^*)$:

$$\int_0^{2\pi} G(\delta(x;\theta, f^*)) m(x;\theta)\ dx$$

where $G(\delta(x;\theta, f^*)) = \frac{(\delta(x;\theta,f^*)+1)^{\alpha+1}}{\alpha\ (\alpha+1)}$.
Examples are:

- $\alpha = -1/2$: Hellinger distance;
- $\alpha \to -1$: Kullback–Leibler divergence;
- $\alpha = -2$: Neyman's Chi–Square.

Numerical Calculation of Distances

Since $f^*(x)$ and $m^*(x;\theta)$ and their transformations are 2π periodic functions the Power Divergence Measure is computed easily and fast by Fast Fourier Transform and Parseval's formula. In fact, let $\phi_k(\cdot)$ the Fourier transform, we get

$$2\pi \int_0^{2\pi} f^*(x)^{\alpha+1}\, m^*(x;\theta)^{-\alpha}\,\mathrm{d}x = \sum_{k=-\infty}^{\infty} \phi_k\left(f^*(x)^{\alpha+1}\right)\phi_k\left(m^*(x;\theta)^{-\alpha}\right)$$

Weighted Likelihood Estimating Equations

The estimating equations of WLEE is a modified version of the MLE equations where at each score is associated a weight defined as follows

$$w(x;\theta,f^*) = \frac{A(\delta(x;\theta,f^*))+1}{\delta(x;\theta,f^*)+1}$$

where $A(\delta)$ is the Residual Adjustment Function, with the form related to the Power Divergence Measure given by

$$A(\delta) = \frac{(\delta+1)^{\alpha+1}-1}{\alpha+1}$$

Hence the WLEE estimator is the solution of

$$\sum_{i=1}^{n} w(x_i;\theta,f^*)u(x_i;\theta) = 0$$

where $u(x;\theta)$ is the score function for the model.

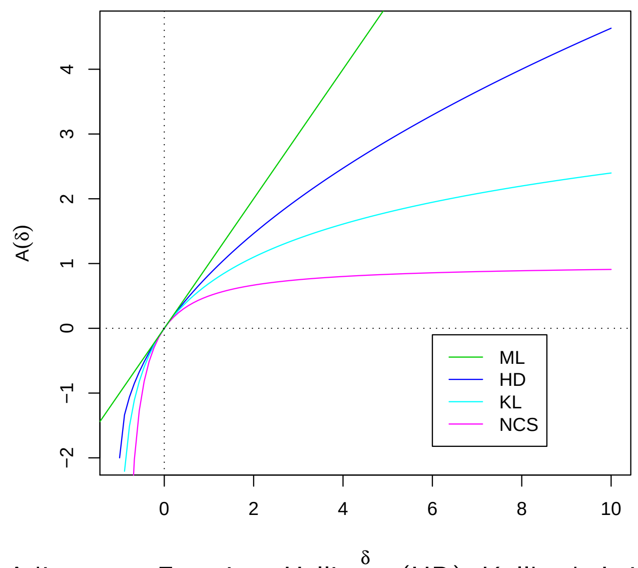
Example: Von Mises distribution

The WLEE for the Von Mises distribution is

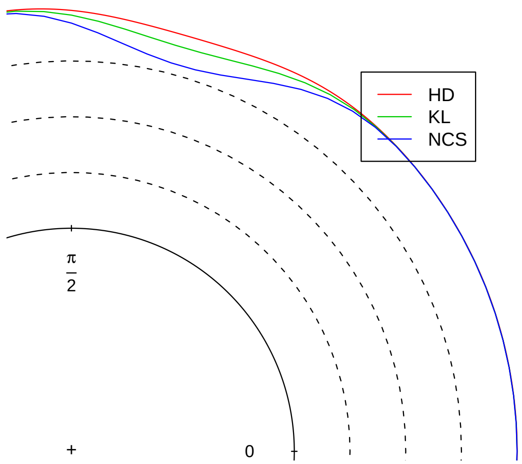
$$\begin{cases} \mu = \arctan^*\left(\frac{\sum_{i=1}^n w(x_i;\mu,\kappa)\ \sin(x_i)}{\sum_{i=1}^n w(x_i;\mu,\kappa)\ \cos(x_i)}\right) \\ \kappa = \mathrm{A}^{-1}\left(\frac{\sum_{i=1}^n w(x_i;\mu,\kappa)\ \cos(x_i-\mu)}{\sum_{i=1}^n w(x_i;\mu,\kappa)}\right) \end{cases}$$

where

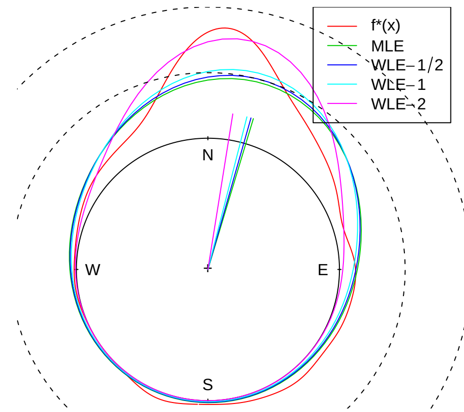
- $\mathrm{A}(\kappa) = \mathrm{I}_1(\kappa)/\mathrm{I}_0(\kappa)$ is the ratio of the two modified Bessel functions of the first kind with order zero and one;
- $\arctan^*$ is the "quadratic-specific" inverse of the tangent that provides the unique inverse of the tangent in $[0, 2\pi)$.



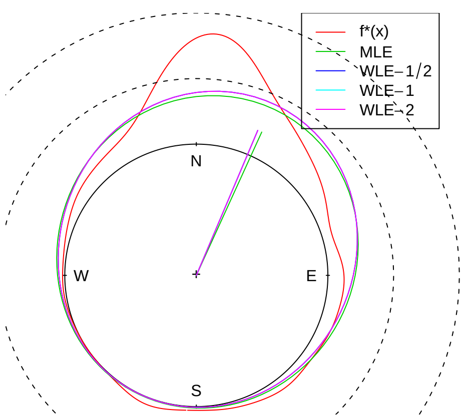
Residual Adjustment Function: Hellinger (HD), Kullback–Leibler (KL), Neyman's Chi–square (NCS) ▸ R code



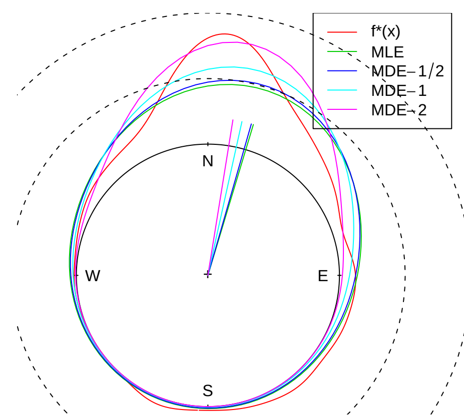
Weight function $(w(x; \theta = (0, 5), f(x)))$ for $f(x) = 0.98m(x; 0, 5) + 0.02m(x; \pi/3, 20)$ where $m(x; \mu, \kappa)$ is the density of a Von Mises distribution. For Residual Adjustment Function: Hellinger (HD), Kullback–Leibler (KL), Neyman's Chi–square (NCS) R code



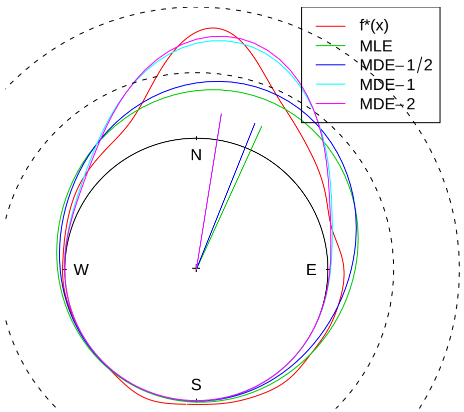
WLE estimators for the Wind direction dataset using the Von Mises model. R code



WLE estimators for the Wind direction dataset using the Wrapped Normal model. R code



MDE estimators for the Wind direction dataset using the Von Mises model. R code



MDE estimators for the Wind direction dataset using the Wrapped Normal model. ▸ R code

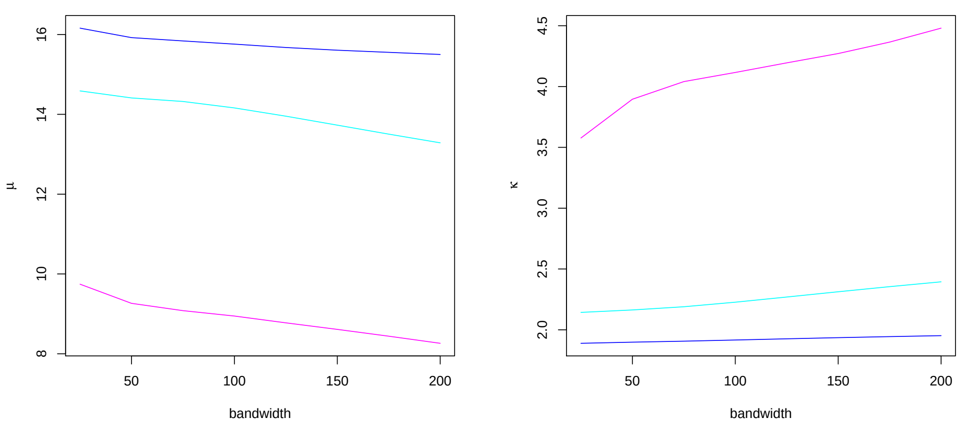
Results for the Wind dataset

Von Mises Model

	MLE	MDE -1/2	MDE -1	MDE -2	WLE -1/2	WLE -1	WLE -2
μ	16.740	15.966	12.456	9.124	15.840	14.322	9.081
κ	1.760	1.961	2.631	4.228	1.907	2.190	4.041

Wrapped Normal Model

	MLE	MDE -1/2	MDE -1	MDE -2	WLE -1/2	WLE -1	WLE -2
μ	24.487	21.787	9.156	9.084	22.414	22.414	22.414
σ	1.005	0.859	0.526	0.504	0.907	0.907	0.907



WLE estimators for the Wind direction dataset using the Von Mises model and different bandwidth. ▸ R code

Conclusions

- We introduced Robust estimators for circular data based on
  - Minimum Distance
  - Weighted Likelihood
- We used, mainly, packages `circular` and `wle`;
- The `Sweave` of the presentation and the dataset would be availables at: `www.dst.unive.it/~claudio/R/index.html`;
- useR! Focus Session on Robust Statistics is Friday, 3:00pm at HS 0.7!

```
> library(circular)
> dataset <- read.table(file = "./wind.dat",
+     sep = ";", header = TRUE)
> mag <- dataset$mag
> dir <- dataset$dir
> ok <- complete.cases(dir, mag)
> mag <- mag[ok]
> magt <- mag
> magt[magt > 3] <- 3.05
> dir <- circular(dir[ok], units = "degrees",
+     template = "geographics")
```



```
> breaks <- circular(0:11/6 * pi - pi/12)
> windrose(x = dir, y = magt, increment = 1,
+     breaks = breaks, fill.col = rev(heat.colors(4)),
+     label.freq = TRUE, main = "")
> legend(x = 0.7, y = 1.2, legend = c("(0, 1]",
+     "(1, 2]", "(2, 3]", "(3, 8.02]"), pch = 19,
+     col = rev(heat.colors(4)), pt.cex = 2)
```

Return


```
> plot(dir, stack = TRUE, ylim = c(-1, 1.22),
+     frame = TRUE)
> mc <- mean(dir)
> lines(c(mc, mc), c(-1, 0.2), col = 2)
> plot(density(dir, bw = 40), ticks = FALSE,
+     ylim = c(-1, 1.8), main = "", xlab = "",
+     ylab = "")
> rose.diag(dir, prop = 2, bins = 40, add = TRUE)
> lines(c(mc, mc), c(-1, 0.2), col = 2)
```

Return


```
> mvm <- mle.vonmises(dir)
> mwn <- mle.wrappednormal(dir)
> mwc <- mle.wrappedcauchy(dir)
> plot(density(dir, bw = 40), ticks = FALSE,
+     xlim = c(-0.8, 1.4), ylim = c(-1, 1.94),
+     main = "", xlab = "", ylab = "", col = 2)
> plot.function.circular(function(x) dvonmises(x,
+     mu = mvm$mu, kappa = mvm$kappa), join = TRUE,
+     add = TRUE, col = 3)
> plot.function.circular(function(x) dwrappednormal(x,
+     mu = mwn$mu, rho = mwn$rho), join = TRUE,
+     add = TRUE, col = 4)
> plot.function.circular(function(x) dwrappedcauchy(x,
+     mu = mwc$mu, rho = mwc$rho), join = TRUE,
+     add = TRUE, col = 5)
```

```
> lines(c(mvm$mu, mvm$mu), c(-1, 0.2), col = 3)
> lines(c(mwn$mu, mwn$mu), c(-1, 0.2), col = 4)
> lines(c(mwc$mu, mwc$mu), c(-1, 0.2), col = 5)
> lines(circular(seq(0, 2 * pi, length = 100)),
+     rep(0.5, 100), lty = 2, col = 1)
> lines(circular(seq(0, 2 * pi, length = 100)),
+     rep(1, 100), lty = 2, col = 1)
> legend(0.8, 2, legend = c("f*(x)", "VM", "WN",
+     "WC"), col = 2:5, lty = rep(1, 4))
```

Return


```
> delta <- function(x, eps = 0.02) {
+     res <- ((1 - eps) * dvonmises(x, circular(0),
+         5) + eps * dvonmises(x, circular(pi/3),
+         20))/dvonmises(x, circular(0), 5) -
+         1
+     return(res)
+ }
> curve.circular(delta, n = 501, join = TRUE,
+     xlim = c(-0.6, 2.7), ylim = c(-0.6, 1.8),
+     col = 2, tcl.text = 0.2, cex = 0.9)
> plot.function.circular(function(x) 0.98 * dvonmises(x,
+     circular(0), 20) + 0.02 * dvonmises(x,
+     circular(pi/3), 20), n = 501, join = TRUE,
+     add = TRUE, col = 3)
> lines(circular(seq(0, 2 * pi, length = 100)),
```



```
+     rep(0.5, 100), lty = 2, col = 1)
> lines(circular(seq(0, 2 * pi, length = 100)),
+     rep(1, 100), lty = 2, col = 1)
> legend(1.2, 2, legend = c(expression(delta(x)),
+     "f(x)", expression(paste("0.02 m(x, ",
+         frac(pi, 3), ", 20)", sep = ""))),
+     lty = rep(1, 3), col = c(2, 3, 4), cex = 0.8)
> cont <- function(x) {
+     den <- 0.02 * dvonmises(x, circular(pi/3),
+         20)
+     y <- c((1 + den) * sin(x), rev(sin(x)))
+     x <- c((1 + den) * cos(x), rev(cos(x)))
+     attr(x, "circularp") <- attr(y, "circularp") <- NULL
+     res <- list(x = x, y = y)
+     return(res)
+ }
```



```
> res <- cont(circular(seq(pi/3 - pi/4, pi/3 +
+     pi/4, 0.01)))
> polygon(x = res, col = 4)
```

Return

```
> A <- function(x, alpha) {
+     if (alpha == -1)
+         a <- log(x + 1)
+     else a <- ((x + 1)^(alpha + 1) - 1)/(alpha +
+         1)
+     return(a)
+ }
> plot(function(x) A(x, alpha = -1/2), from = -1,
+     to = 10, xlab = expression(delta), ylab = expression(
+     col = 4)
> plot(function(x) A(x, alpha = -1), from = -1,
+     to = 10, col = 5, add = TRUE)
> plot(function(x) A(x, alpha = -2), from = -1,
+     to = 10, col = 6, add = TRUE)
> abline(0, 1, col = 3)
```


Focus Session on Robust Statistics is Friday, 3:00pm at HS 0.7

```
> abline(h = 0, lty = 3)
> abline(v = 0, lty = 3)
> legend(6, -0.1, legend = c("ML", "HD", "KL",
+     "NCS"), col = 3:6, lty = rep(1, 4))
```

Return

Focus Session on Robust Statistics is Friday, 3:00pm at HS 0.7

```
> w <- function(x, alpha) {
+     w <- (A(x, alpha) + 1)/(x + 1)
+     w[w > 1] <- 1
+     w[w < 0] <- 0
+     return(w)
+ }
> weight <- function(x, alpha = -1/2) {
+     w(delta(x), alpha)
+ }
> curve.circular(weight, n = 501, join = TRUE,
+     xlim = c(-0.2, 2), ylim = c(0, 2), col = 2,
+     tcl.text = 0.2, cex = 0.9)
> plot.function.circular(function(x) weight(x,
+     alpha = -1), add = TRUE, col = 3)
> plot.function.circular(function(x) weight(x,
```


Focus Session on Robust Statistics is Friday, 3:00pm at HS 0.7

```
+     alpha = -2), add = TRUE, col = 4)
> lines(circular(seq(0, 2 * pi, length = 100)),
+     rep(0.25, 100), lty = 2, col = 1)
> lines(circular(seq(0, 2 * pi, length = 100)),
+     rep(0.5, 100), lty = 2, col = 1)
> lines(circular(seq(0, 2 * pi, length = 100)),
+     rep(0.75, 100), lty = 2, col = 1)
> legend(1.3, 1.7, legend = c("HD", "KL", "NCS"),
+     col = 2:4, lty = rep(1, 3))
```

Return

```
> library(wle)
> wvmhd <- wle.vonmises(dir, smooth = 75, group = 5,
+     alpha = -1/2)
> wvmkl <- wle.vonmises(dir, smooth = 75, group = 5,
+     alpha = -1)
> wvmncs <- wle.vonmises(dir, smooth = 75, group = 5,
+     alpha = -2)
> plot(density(dir, bw = 40), ticks = FALSE,
+     xlim = c(-0.8, 1.4), ylim = c(-1, 1.94),
+     main = "", xlab = "", ylab = "", col = 2)
> plot.function.circular(function(x) dvonmises(x,
+     mu = mvm$mu, kappa = mvm$kappa), join = TRUE,
+     add = TRUE, col = 3)
> lines(c(mvm$mu, mvm$mu), c(-1, 0.2), col = 3)
> plot.function.circular(function(x) dvonmises(x,
```



```
+     mu = wvmhd$mu, kappa = wvmhd$kappa), join = TRUE,
+     add = TRUE, col = 4)
> lines(c(wvmhd$mu, wvmhd$mu), c(-1, 0.2), col = 4)
> plot.function.circular(function(x) dvonmises(x,
+     mu = wvmkl$mu, kappa = wvmkl$kappa), join = TRUE,
+     add = TRUE, col = 5)
> lines(c(wvmkl$mu, wvmkl$mu), c(-1, 0.2), col = 5)
> plot.function.circular(function(x) dvonmises(x,
+     mu = wvmncs$mu, kappa = wvmncs$kappa),
+     join = TRUE, add = TRUE, col = 6)
> lines(c(wvmncs$mu, wvmncs$mu), c(-1, 0.2),
+     col = 6)
> lines(circular(seq(0, 2 * pi, length = 100)),
+     rep(0.5, 100), lty = 2, col = 1)
> lines(circular(seq(0, 2 * pi, length = 100)),
+     rep(1, 100), lty = 2, col = 1)
```



```
> legend(0.8, 2, legend = c("f*(x)", "MLE", expression(past
+     -1/2, sep = "")), expression(paste("WLE",
+     -1, sep = "")), expression(paste("WLE",
+     -2, sep = ""))), col = 2:6, lty = rep(1,
+     5))
```

Return


```
> library(wle)
> wwnhd <- wle.wrappednormal(dir, smooth = 0.006,
+     group = 5, alpha = -1/2)
> wwnkl <- wle.wrappednormal(dir, smooth = 0.006,
+     group = 5, alpha = -1)
> wwnncs <- wle.wrappednormal(dir, smooth = 0.006,
+     group = 5, alpha = -2)
> plot(density(dir, bw = 40), ticks = FALSE,
+     xlim = c(-0.8, 1.4), ylim = c(-1, 1.94),
+     main = "", xlab = "", ylab = "", col = 2)
> plot.function.circular(function(x) dwrappednormal(x,
+     mu = mwn$mu, rho = mwn$rho), join = TRUE,
+     add = TRUE, col = 3)
> lines(c(mwn$mu, mwn$mu), c(-1, 0.2), col = 3)
> plot.function.circular(function(x) dwrappednormal(x,
```

```
+     mu = wwnhd$mu, rho = wwnhd$rho), join = TRUE,
+     add = TRUE, col = 4)
> lines(c(wwnhd$mu, wwnhd$mu), c(-1, 0.2), col = 4)
> plot.function.circular(function(x) dwrappednormal(x,
+     mu = wwnkl$mu, rho = wwnkl$rho), join = TRUE,
+     add = TRUE, col = 5)
> lines(c(wwnkl$mu, wwnkl$mu), c(-1, 0.2), col = 5)
> plot.function.circular(function(x) dwrappednormal(x,
+     mu = wwnncs$mu, rho = wwnncs$rho), join = TRUE,
+     add = TRUE, col = 6)
> lines(c(wwnncs$mu, wwnncs$mu), c(-1, 0.2),
+     col = 6)
> lines(circular(seq(0, 2 * pi, length = 100)),
+     rep(0.5, 100), lty = 2, col = 1)
> lines(circular(seq(0, 2 * pi, length = 100)),
+     rep(1, 100), lty = 2, col = 1)
```



```
> legend(0.8, 2, legend = c("f*(x)", "MLE", expression(past
+     -1/2, sep = "")), expression(paste("WLE",
+     -1, sep = "")), expression(paste("WLE",
+     -2, sep = ""))), col = 2:6, lty = rep(1,
+     5))
```

Return


```
> library(wle)
> dvmhd <- mde.vonmises(dir, bw = 75, n = 1024,
+     alpha = -1/2)
> dvmkl <- mde.vonmises(dir, bw = 75, n = 1024,
+     alpha = -1)
> dvmncs <- mde.vonmises(dir, bw = 75, n = 1024,
+     alpha = -2)
> plot(density(dir, bw = 40), ticks = FALSE,
+     xlim = c(-0.8, 1.4), ylim = c(-1, 1.94),
+     main = "", xlab = "", ylab = "", col = 2)
> plot.function.circular(function(x) dvonmises(x,
+     mu = mvm$mu, kappa = mvm$kappa), join = TRUE,
+     add = TRUE, col = 3)
> lines(c(mvm$mu, mvm$mu), c(-1, 0.2), col = 3)
> plot.function.circular(function(x) dvonmises(x,
```



```
+     mu = dvmhd$mu, kappa = dvmhd$kappa), join = TRUE,
+     add = TRUE, col = 4)
> lines(c(dvmhd$mu, dvmhd$mu), c(-1, 0.2), col = 4)
> plot.function.circular(function(x) dvonmises(x,
+     mu = dvmkl$mu, kappa = dvmkl$kappa), join = TRUE,
+     add = TRUE, col = 5)
> lines(c(dvmkl$mu, dvmkl$mu), c(-1, 0.2), col = 5)
> plot.function.circular(function(x) dvonmises(x,
+     mu = dvmncs$mu, kappa = dvmncs$kappa),
+     join = TRUE, add = TRUE, col = 6)
> lines(c(dvmncs$mu, dvmncs$mu), c(-1, 0.2),
+     col = 6)
> lines(circular(seq(0, 2 * pi, length = 100)),
+     rep(0.5, 100), lty = 2, col = 1)
> lines(circular(seq(0, 2 * pi, length = 100)),
+     rep(1, 100), lty = 2, col = 1)
```

```
> legend(0.8, 2, legend = c("f*(x)", "MLE", expression(past
+     -1/2, sep = "")), expression(paste("MDE",
+     -1, sep = "")), expression(paste("MDE",
+     -2, sep = ""))), col = 2:6, lty = rep(1,
+     5))
```

Return


```
> library(wle)
> dwnhd <- mde.wrappednormal(dir, bw = 0.0375,
+     n = 1024, alpha = -1/2)
> dwnkl <- mde.wrappednormal(dir, bw = 0.0375,
+     n = 1024, alpha = -1)
> dwnncs <- mde.wrappednormal(dir, bw = 0.09,
+     n = 1024, alpha = -2)
> plot(density(dir, bw = 40), ticks = FALSE,
+     xlim = c(-0.8, 1.4), ylim = c(-1, 1.94),
+     main = "", xlab = "", ylab = "", col = 2)
> plot.function.circular(function(x) dwrappednormal(x,
+     mu = mwn$mu, rho = mwn$rho), join = TRUE,
+     add = TRUE, col = 3)
> lines(c(mwn$mu, mwn$mu), c(-1, 0.2), col = 3)
> plot.function.circular(function(x) dwrappednormal(x,
```



```
+     mu = dwnhd$mu, rho = dwnhd$rho), join = TRUE,
+     add = TRUE, col = 4)
> lines(c(dwnhd$mu, dwnhd$mu), c(-1, 0.2), col = 4)
> plot.function.circular(function(x) dwrappednormal(x,
+     mu = dwnkl$mu, rho = dwnkl$rho), join = TRUE,
+     add = TRUE, col = 5)
> lines(c(dwnkl$mu, dwnkl$mu), c(-1, 0.2), col = 5)
> plot.function.circular(function(x) dwrappednormal(x,
+     mu = dwnncs$mu, rho = dwnncs$rho), join = TRUE,
+     add = TRUE, col = 6)
> lines(c(dwnncs$mu, dwnncs$mu), c(-1, 0.2),
+     col = 6)
> lines(circular(seq(0, 2 * pi, length = 100)),
+     rep(0.5, 100), lty = 2, col = 1)
> lines(circular(seq(0, 2 * pi, length = 100)),
+     rep(1, 100), lty = 2, col = 1)
```



```
> legend(0.8, 2, legend = c("f*(x)", "MLE", expression(past
+     -1/2, sep = "")), expression(paste("MDE",
+     -1, sep = "")), expression(paste("MDE",
+     -2, sep = ""))), col = 2:6, lty = rep(1,
+     5))
```

Return

```
> bw <- seq(25, 200, 25)
> temp <- function(x, dir, alpha) wle.vonmises(x = dir,
+     group = 5, smooth = x, alpha = alpha)$mu
> muhdbw <- sapply(bw, temp, dir = dir, alpha = -1/2)
> muklbw <- sapply(bw, temp, dir = dir, alpha = -1)
> muncsbw <- sapply(bw, temp, dir = dir, alpha = -2)
> plot(bw, muhdbw, xlab = "bandwidth", ylab = expression(mu
+     main = "", type = "l", col = 4, ylim = range(muhdbw,
+         muklbw, muncsbw))
> lines(bw, muklbw, col = 5)
> lines(bw, muncsbw, col = 6)
> bw <- seq(25, 200, 25)
> temp <- function(x, dir, alpha) wle.vonmises(x = dir,
+     group = 5, smooth = x, alpha = alpha)$kappa
> kappahdbw <- sapply(bw, temp, dir = dir, alpha = -1/2)
```



```
> kappaklbw <- sapply(bw, temp, dir = dir, alpha = -1)
> kappancsbw <- sapply(bw, temp, dir = dir, alpha = -2)
> plot(bw, kappahdbw, xlab = "bandwidth", ylab = expression
+     main = "", type = "l", col = 4, ylim = range(kappahdb
+         kappaklbw, kappancsbw))
> lines(bw, kappaklbw, col = 5)
> lines(bw, kappancsbw, col = 6)
```

Return

C. Agostinelli. *Inferenza statistica robusta basata sulla funzione di verosimiglianza pesata: alcuni sviluppi*. PhD thesis, Dipartimento di Scienze Statistiche, Università di Padova, 1998.

C. Agostinelli. Robust estimation for circular data. Working Paper 2004.3, Dipartimento di Statistica, Università di Venezia, Venezia, 2004a.

Z.D. Bai, C.R. Rao, and L.C. Zhao. Kernel estimators of density function of directional data. *Journal of Multivariate Analysis*, 27: 24–39, 1988.

V. Barnett and T. Lewis. *Outliers in Statistical Data*. John Wiley, New York, 1994.

D. Collett. Outliers in circular data. *Applied Statistics*, 29(1): 50–57, 1980.

N. Cressie and T.R.C. Read. Multinomial goodness–of–fit tests. *Journal of the Royal Statistical Society, Series B*, 46:440–464, 1984.

N. Cressie and T.R.C. Read. *Cressie–Read Statistic*, pages 37–39. Wiley, 1988. In: Encyclopedia of Statistical Sciences, Supplementary Volume, edited by S. Kotz and N.L. Johnson.

D.L. Donoho and R.C. Liu. The "automatic" robustness of minimum distance estimators. *Annals of Statistics*, 16(2): 552–586, 1988.

D.E. Ferguson, H.F. Landreth, and J.P. McKeown. Sun compass orientation of the northern cricket frog, *Acris crepitans*. *Animal Behaviour*, 14:45–53, 1967.

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

F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw, and W.A. Stahel. *Robust Statistics: The Approach based on Influence Functions*. Wiley, 1986.

D. Hawkins. *Identification of Outliers*. Chapman & Hall, 1980.

X. He. Robust statistics of directional data: A survey. *Nonparametric Statistics and Related Topics*, pages 87–95, 1992.

X. He and D.G. Simpson. Robust direction estimation. *The Annals of Statistics*, 20(1):351–369, 1992.

H. Hendriks. Nonparametric estimation of a probability density on a riemnnian manifold using fourier expansions. *The Annals of Statistics*, 18(2):832–849, 1990.

S.R. Jammalamadaka and A SenGupta. *Topics in Circular Statistics*, volume 5 of *Multivariate Analysis*. World Scientific, 2001.

P.E. Jupp and K.V. Mardia. A unified view of the theory of directional statistics, 1975–1988. *International Statistical Review*, 57(3):261–294, 1989.

P.T. Kim. Deconvolution density estimation on so(n). *The Annals of Statistics*, 26(3):1083–1102, 1998.

J. Klemelä. Estimation of densities and derivatives of densities with directioinal data. *Journal of Multivariate Analysis*, 73: 18–40, 2000.

E.M. Knorr and R.T. Ng. A unified notion of outliers: Properties and computation. In *Proc. KDD*, pages 219–222, 1997. An extended version of this paper appears as: Knorr, E.M., Ng, R.T. (1997) A unified approach for mining outliers. In: Proc. CASCON, pag. 236–248.

D. Ko. Robust estimation of the concentration parameter of the von mises–fisher distribution. *The Annals of Statistics*, 20(2): 917–928, 1992.

D. Ko and T. Chang. Robust m-estimators on spheres. *Journal of Multivariate Analysis*, 45:104–136, 1993.

D. Ko and P. Guttorp. Robustness of estimators for directional data. *The Annals of Statistics*, 16(2):609–618, 1988.

R.V. Lenth. Robust measures of location for directional data. *Technometrics*, 23(2):77–81, 1981.

B.G. Lindsay. Efficiency versus robustness: The case for minimum hellinger distance and related methods. *Annals of Statistics*, 22: 1018–1114, 1994.

K.V. Mardia and P.E. Jupp. *Directional Statistics*. Wiley, 2000.

M. Markatou, A. Basu, and B.G. Lindsay. Weighted likelihood estimating equations: The continuous case. Technical report, Department of Statistics, Columbia University, New York, 1995.

M. Markatou, A. Basu, and B.G. Lindsay. Weighted likelihood estimating equations with a bootstrap root search. *Journal of the American Statistical Association*, 93:740–750, 1998.

C. Park, A. Basu, and B.G. Lindsay. The residual adjustment function and weighted likelihood: A graphical interpretation of robustness of minimum disparity estimators. *Computational Statistics and Data Analysis*, 39(1):21–33, 2002.

V.R. Prayag and A.P. Gore. Density estimation for randomly distributed circular objects. *Metrika*, 37:63–69, 1990.

E. Ronchetti. *Optimal Robust Estimators for the Concentration Parameter of a von Mises–Fisher Distribution*, chapter 6, pages 65–74. Wiley, 1992. In: The art of Statistical Science, edited by K.V. Mardia.

A. SenGupta and A.K. Laha. The slippage problem for the circular normal distribution. *Australian and New Zealand Journal of Statistics*, 43(4):461–471, 2001.

G. Upton. Outliers in circular data. *Journal of Applied Statistics*, 20:229–235, 1993.

These slides are prepared using LaTeX, beamer class and Sweave package in R . They are compiled with R ver. 2.2.0 running under OS darwin7.9.0 and packages `circular` ver. 0.3-5, `wle` ver. 0.9-2.