logcondens: Computations Related to Univariate Log-Concave Density Estimation

Lutz Dümbgen
University of Bern

Kaspar Rufibach
University of Zurich

Abstract

Maximum likelihood estimation of a log-concave density has attracted considerable attention over the last few years. Several algorithms have been proposed to estimate such a density. Two of those algorithms, an iterative convex minorant and an active set algorithm, are implemented in the R package **logcondens**. While these algorithms are discussed elsewhere, we describe in this paper the use of the **logcondens** package and discuss functions and datasets related to log-concave density estimation contained in the package. In particular, we provide functions to (1) compute the maximum likelihood estimate (MLE) as well as a smoothed log-concave density estimator derived from the MLE, (2) evaluate the estimated density, distribution and quantile functions at arbitrary points, (3) compute the characterizing functions of the MLE, (4) sample from the estimated distribution, and finally (5) perform a two-sample permutation test using a modified Kolmogorov-Smirnov test statistic. In addition, **logcondens** makes two datasets available that have been used to illustrate log-concave density estimation.

Keywords: log-concave, density estimation, Kolmogorov-Smirnov test, R.

1. Introduction

1.1. About this document

Although first uploaded to CRAN in 2006, a detailed description (beyond the package manual) of the functionality of the R package **logcondens** (Rufibach and Dümbgen 2010) had been lacking so far. This document is an introduction to **logcondens**, based on Dümbgen and Rufibach (2010), that not only discusses and illustrates the use of the implemented iterative convex minorant (**ICMA**) and the active set algorithm (**ASA**), but more functions that are useful in connection with univariate log-concave density estimation. The package also provides functions to evaluate quantities whose explicit computations are not immediate, such as distribution and quantile functions, the smoothed log-concave density estimator and the corresponding distribution function, sampling from the different estimators, and a two-sample permutation test. In addition, the data sets analyzed in Dümbgen and Rufibach (2009) and Koenker and Mizera (2010) are provided as part of **logcondens**. Using the former of these datasets we illustrate how **logcondens** can be used to explore data. The package is available from the Comprehensive R Archive Network at `http://CRAN.R-project.org/package=logcondens`.

This document was created using `Sweave` (Leisch 2002), LaTeX (Knuth 1984; Lamport 1994), and R (R Development Core Team 2010). This means that all of the code has been checked by R.

1.2. Log-concave density estimation

One way to nonparametrically estimate a density from univariate i.i.d. data is imposing a qualitative constraint such as monotonicity, convexity, or log-concavity. In contrast to smoothing methods such as kernel estimation or roughness penalization, methods relying on shape constraints are fully automatic, i.e. they do not necessitate any choice of tuning parameters such as a bandwidth or a penalty parameter. Choosing these tuning parameters is notoriously involved, since typically their optimal values depend on properties of the unknown density to be estimated. To fix notation, let f be a probability density on $\mathbb{R}$. We call f *log-concave* if it may be written as

$$f(x) \;=\; \exp\varphi(x)$$

for some concave function $\varphi : \mathbb{R} \to [-\infty, \infty)$. Based on a sample of i.i.d. random variables $X_1, \ldots, X_n \in \mathbb{R}$ from f we seek to estimate this density via maximizing the normalized log-likelihood function

$$\ell(\varphi) \;=\; n^{-1} \sum_{i=1}^{n} \log f(X_i) \;=\; n^{-1} \sum_{i=1}^{n} \varphi(X_i)$$

over all concave functions $\varphi : \mathbb{R} \to [-\infty, \infty)$ such that $\int \exp\varphi(x)\,\mathrm{d}x = 1$. The resulting maximum likelihood estimators (MLEs) of φ and f are denoted by $\widehat{\varphi}$ and $\widehat{f} = \exp\widehat{\varphi}$, respectively. Dümbgen and Rufibach (2009) show that the maximizer $\widehat{\varphi}$ of ℓ is unique, piecewise linear on the interval $[X_{(1)}, X_{(n)}]$ with knots only at (some of the) observations $X_{(i)}$, and $\widehat{\varphi} = -\infty$ elsewhere. Here $X_{(1)} \le X_{(2)} \le \cdots \le X_{(n)}$ are the ordered observations, and a "knot" of $\widehat{\varphi}$ is a location where this function changes slope. The MLEs $\widehat{\varphi}$, $\widehat{f}$ and $\widehat{F}$ are consistent with certain rates of convergence, see Dümbgen and Rufibach (2009) and Balabdaoui, Rufibach, and Wellner (2009).

The merits of using a log-concave density have been extensively described in Balabdaoui *et al.* (2009), Cule, Gramacy, and Samworth (2009), and Walther (2009). The most relevant properties, in our opinion, are:

- Many parametric models consist of log–concave densities, at least for large parts of the parameter space. Examples include: Normal, Uniform, Gamma(r, λ) for $r \ge 1$, Beta(a, b) for $a, b \ge 1$, generalized Pareto, Gumbel, Fréchet, logistic or Laplace, to mention only some of these models. Therefore, assuming log–concavity offers a flexible non–parametric alternative to purely parametric models. Note that a log–concave density need not be symmetric.

- Given that log-concavity seems to be a plausible assumption for many datasets one encounters in applications, we also advocate the use of this package in routine data analysis. Instead of looking at a histogram (via `hist()`) or a kernel density estimate when exploring data we propose to additionally display the log-concave estimate, or its smoothed version described below, to get an idea about the distribution of the data. Especially for small sample sizes, kernel estimates are prone to artifacts, but

assuming additional structure on the underlying distribution reduces the influence of single observations on the estimated density. To illustrate this point we added the estimate resulting from invoking `density()` to Figure 1 (black line).

- Every log–concave density is automatically unimodal.
- The nonparametric MLE of a continuous unimodal density does not exist (see e.g. Birgé 1997) but the nonparametric MLE of a log-concave density does. Thus the class of log-concave densities may be a useful and valuable surrogate for the larger class of unimodal densities.
- Using a standard EM algorithm, mixtures with log-concave component densities can be computed, see Chang and Walther (2007) and Cule, Samworth, and Stewart (2010).
- Log-concavity turns out to be a valuable assumption in dimensions higher than 1 as well, see Cule *et al.* (2009, 2010); Koenker and Mizera (2010); Schuhmacher, Hüsler, and Dümbgen (2009); Seregin and Wellner (2010).

Figure 1 provides an introductory example based on simulated data. Namely, we generated a random sample of size $n = 40$ from a standard normal distribution (the corresponding density function is in fact log-concave) and estimated the density of these observations under a log-concavity restriction. In the left plot of Figure 1 we provide plots of the true density the observations are sampled from, the estimated density, and the kernel density estimate as provided by `density()`. Given the small sample size of $n = 40$, both the log-concave and kernel estimate capture the shape of the truth quite accurately. The features of the log-concave MLE are visible: The support of $\widehat{f}$ is $[X_{(1)}, X_{(n)}]$ and the knots, marked by vertical lines at the bottom of the plot, are indeed at some of the observations. On the other hand, the notorious bumpy behavior of the kernel estimate appears in the tails, especially on the left hand side. Piecewise linearity of the estimated log-density is visualized in the right plot of Figure 1.

We will illustrate the main functions of **logcondens** on this simulated dataset and then discuss their merits in analyzing a real-world dataset. To generate the data, invoke:

```
R> library("logcondens")
R> set.seed(1)
R> n.sim <- 40
R> x.sim <-sort(rnorm(n.sim))
```

Then, we compute and display the log-concave density estimate, the density we sampled from, and a standard kernel estimate:

```
R> res <- logConDens(x.sim, smoothed = FALSE, print = FALSE)
R> xs <- seq(-5, 5, by = 0.01)
R> f.true <- dnorm(xs)
R> par(las = 1, oma = c(0, 0, 3, 0), mar = c(3, 3.5, 0.5, 0.5),
+      mfrow = c(1, 2))
R> plot(res, which = "density", add.title = FALSE, legend.pos = "none")
R> title(main = "Log-concave density estimation from i.i.d. data",
```

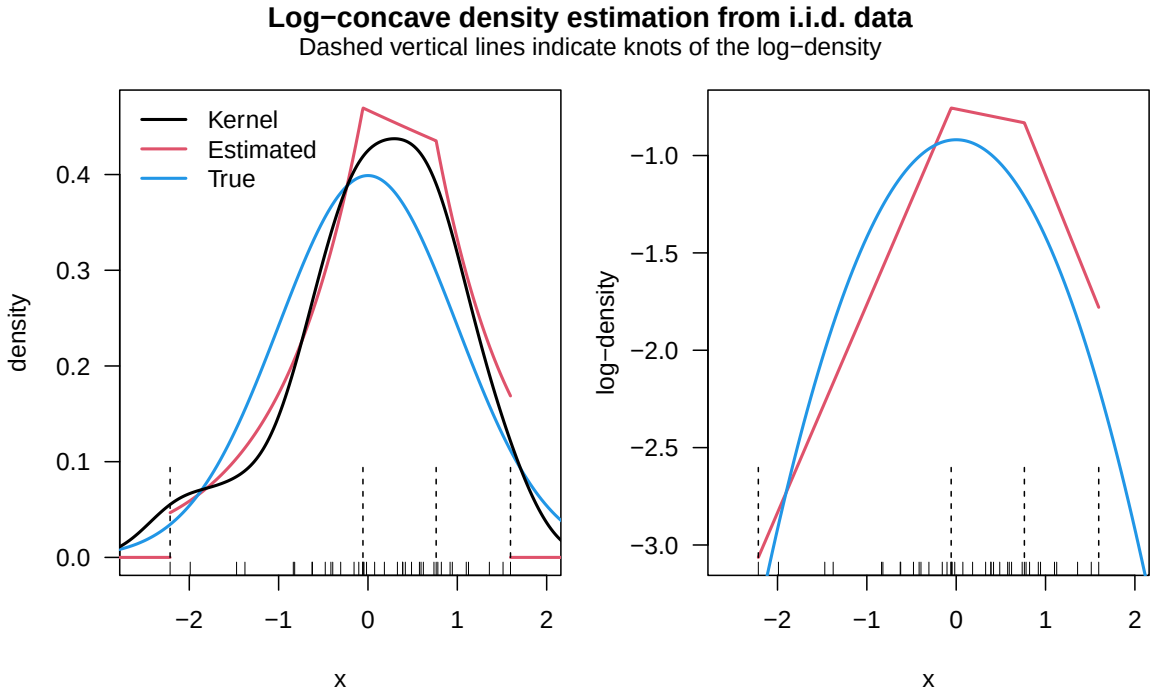

Figure 1: Introductory example based on simulated data.

```
+      outer = TRUE)
R> mtext("Dashed vertical lines indicate knots of the log-density", 3,
+      outer = TRUE)
R> lines(xs, f.true, col = 4, lwd = 2)
R> lines(density(x.sim), lwd = 2)
R> legend("topleft", c("Kernel", "Estimated", "True"), lty = 1, lwd = 2,
+      col = c(1, 2, 4), bty = "n")
R> plot(res, which = "log-density", add.title = FALSE, legend.pos = "none")
R> lines(xs, log(f.true), col = 4, lwd = 2)
```

In Sections 2 and 3 we briefly review some theory behind log-concave density estimation. The use of the smoothed log-concave density estimator introduced in Dümbgen and Rufibach (2009) is illustrated in Section 4. After explaining the implementation of the main functions of the package **logcondens** in Section 5, sampling from both estimators is discussed briefly in Section 6. In Section 7 we take up the simulated data from the introduction and use it to illustrate the package's main functions. How to apply log-concave densities for the analysis of a real world dataset is shown in Section 8. We conclude in Section 9 with a small simulation study illustrating the use of log-concave estimates to improve the power of a two-sample Kolmogorov-Smirnov test.

Explicit formulae for the distribution, integrated distribution, quantile function, the smoothed estimator as well as for the computation of the difference between two log-concave CDFs are postponed to the appendix.

2. Computing the log-concave estimator

The general log-likelihood. As discussed in the introduction, our goal is to maximize the functional $\ell(\varphi)$ over all log-concave functions $\varphi : \mathbb{R} \to [-\infty, \infty)$ such that $\exp(\varphi)$ is a probability density. Let us first modify ℓ somewhat: For given support points $x_1 < x_2 < \cdots < x_m$ and probability weights $w_1, w_2, \ldots, w_m > 0$ we consider

$$\ell(\varphi) \;=\; \sum_{j=1}^{m} w_j \varphi(x_j).$$

The standard setting. By default, $x_1 < x_2 < \ldots < x_m$ are the different elements of $\{X_1, X_2, \ldots, X_n\}$, and $w_j := n^{-1} \# \{i \le n : X_i = x_j\}$. In the idealized setting of i.i.d. observations X_i from a density f, the numbers m and n coincide, while $x_j = X_{(j)}$ and $w_j = n^{-1}$. However, there could be tied observations, e.g. due to rounding errors, resulting in $m < n$ support points.

An alternative setting. In case of very large sample sizes one may wish to reduce computation time and storage space by approximating the raw data X_i. Let $x_1 < x_2 < \cdots < x_m$ be given support points such that $[x_1, x_m] \supset [X_{(1)}, X_{(n)}]$. Then we define probability weights w_j, $1 \le j \le m$, as follows: If a raw observation X_i falls into $[x_j, x_{j+1}]$, we add $n^{-1}(x_{j+1} - X_i)/(x_{j+1} - x_j)$ to w_j and $n^{-1}(X_i - x_j)/(x_{j+1} - x_j)$ to w_{j+1}. That way we approximate the empirical distribution of the raw data X_i by a discrete distribution $\sum_{j=1}^{m} w_j \delta_{x_j}$ having the same mean and possibly larger variance. However one can show that the increase in variance is no larger than $\max_{j<m}(x_{j+1} - x_j)^2/4$. Finally, pairs (x_j, w_j) with $w_j = 0$ are removed to reduce the dimension as much as possible.

We implemented automatic computation of weights and the binning algorithm described above in the function `preProcess` in **logcondens**. This function is automatically invoked by those functions that take the raw data as argument, specifically `activeSetLogCon`, `icmaLogCon` and the functions that depend on them, most importantly `logConDens`. For a description of how to specify a user-defined grid we refer to the respective help files in `logcondens`.

The modified log-likelihood. To relax the constraint of f being a density and to get a criterion function to maximize over *all* concave functions, we employ the standard trick (cf. Silverman 1982, Theorem 3.1) of adding a Lagrange term to ℓ, leading to the modified functional

$$L(\varphi) \;=\; \sum_{j=1}^{m} w_j \varphi(x_j) - \int_{\mathbb{R}} \exp \varphi(t)\, \mathrm{d}t. \tag{1}$$

With the same arguments as Dümbgen and Rufibach (2009) one can show that there exists a unique concave function $\widehat{\varphi}$ maximizing this functional $L(\cdot)$. It satisfies the equation $\int \exp \widehat{\varphi}(t)\, \mathrm{d}t = 1$ and has the following additional properties: $\widehat{\varphi} = -\infty$ on $\mathbb{R} \setminus [x_1, x_m]$, and $\widehat{\varphi}$ is continuous and piecewise linear on $[x_1, x_m]$ with knots only in $\{x_1, x_2, \ldots, x_m\}$. Any such function φ is fully specified by the vector $\varphi = (\varphi(x_j))_{j=1}^{m}$. We therefore restrict attention to the set $\mathcal{P}(x_1, x_2, \ldots, x_m)$ of vectors $\varphi \in \mathbb{R}^m$ such that

$$\frac{\varphi_{j+1} - \varphi_j}{x_{j+1} - x_j} \;\ge\; \frac{\varphi_j - \varphi_{j-1}}{x_j - x_{j-1}} \quad \text{for } i = 2, \ldots, m-1.$$

This allows to rewrite the integral (1) as

$$L(\varphi) \;=\; \sum_{j=1}^{m} w_j \varphi_j - \sum_{j=1}^{m-1} (x_{j+1} - x_j) J(\varphi_j, \varphi_{j+1})$$

with the auxiliary function

$$J(r,s) \;=\; \begin{cases} (\exp(r) - \exp(s))/(r-s) & \text{if } r \neq s, \\ \exp(r) & \text{if } r = s. \end{cases}$$

The *constrained optimization* problem we are now aiming to solve reads

$$\widehat{\varphi} \;=\; \operatorname*{arg\,max}_{\varphi \in \mathcal{P}(x_1, x_2, \ldots, x_m)} L(\varphi),$$

where L is a strictly concave functional on $\mathbb{R}^m$, see Dümbgen, Hüsler, and Rufibach (2010, Section 2). For more details on the computations leading to the final form of L we refer to Dümbgen and Rufibach (2009). It is important to note that (iterative) maximization of the functional L yields the vector $(\widehat{\varphi}(x_j))_{j=1}^m$ and not the density estimate directly. However, due to the piecewise linearity of the function $\widehat{\varphi}$ the maximizing vector $(\widehat{\varphi}(x_j))_{j=1}^m$ can be identified with

$$\widehat{\varphi}(x) \;=\; \begin{cases} \widehat{\varphi}_j + (x - x_j)\widehat{s}_{j+1} & \text{for } x \in [x_j, x_{j+1}], 1 \leq j < m \\ -\infty & \text{for } x \in \mathbb{R} \setminus [x_1, x_m] \end{cases}$$

for $x \in \mathbb{R}$, where $\widehat{s}_{j+1} = \Delta\widehat{\varphi}_{j+1}/\Delta x_{j+1}$ and $\Delta v_{j+1} := v_{j+1} - v_j$, $1 \leq j < m$, for any vector $\mathbf{v} \in \mathbb{R}^m$. Finally, the density estimate at x is then simply $\widehat{f} = \exp\widehat{\varphi}$, i.e. $\widehat{f} = 0$ outside $[x_1, x_m]$. These two functions are implemented in `evaluateLogConDens`. Computation of additional functions at a given point x is discussed in Section 5.2.

Approximation. From the definitions of $L(\varphi)$ and $J(r,s)$ above it is clear that numerical inaccuracies may occur whenever two consecutive components φ_j, φ_{j+1} of the vector φ under consideration are getting very close. This means that the argument φ has "flat" stretches. To avoid these numerical inaccuracies in computations, we approximate J and its derivatives in the implementation in **logcondens** by Taylor polynomials of degree four if $|r-s|$ is small. Exact bounds and formulae for these polynomials are worked out in Dümbgen *et al.* (2010, Section 6). Similar approximations are also used to compute $\int \widehat{F}$ and $\widehat{F}^*$, see the appendix.

An iterative convex minorant algorithm. First attempts to compute $\widehat{\varphi}$ are described in Rufibach (2007): Four different algorithms were proposed that all reliably found the maximum of L. However, not all are equally efficient. As a clear winner in the contest arranged in that paper in terms of speed came off the ICMA. For this reason, this algorithm was chosen to be implemented in **logcondens**, as the function `icmaLogCon`. First proposed by Groeneboom and Wellner (1992) and further detailed by Jongbloed (1998), the ICMA is especially tailored to maximize a smooth objective function over particular convex cones by maximizing quadratic approximations to the objective function via the pool-adjacent-violaters algorithm (PAVA).

An active set algorithm. For a description of active set algorithms see e.g. Fletcher (1987, Section 10.3) or Nocedal and Wright (1999, Section 16.4). The application to log-concave density estimation is discussed in considerable generality by Dümbgen *et al.* (2010, Section 3) and therefore omitted here. A key feature of an ASA is that it solves a *finite* number of *unconstrained* optimization problems. The function `activeSetLogCon` estimates a log-concave density via an ASA.

Related work. Note also the implementations of the above two algorithms for *isotonic* estimation in the R package **isotone** and the description in de Leeuw, Hornik, and Mair (2009). In a similar context, an ASA was used to compute an estimate in the ordered factor regression problem, see Rufibach (2010) and the R package **OrdFacReg** (Rufibach 2009).

Range of applicability. The minimal sample size that allows estimation of a log-concave density is $n = 2$. Estimating a log-concave density for n in the millions causes no problems on a laptop computer and depending on the resources may take a few minutes. Alternatively, to speed up computation time for large n, one can approximate the empirical distribution of the raw data by a discrete distribution with $m << n$ support points as indicated before.

3. Characterization and properties of the estimator

In what follows let $\widehat{F}$ be the distribution function corresponding to the density $\widehat{f} = \exp\widehat{\varphi}$, and let $\mathbb{F}$ be the distribution function of the discrete measure $\sum_{j=1}^m w_j \delta_{x_j}$. Using suitable directional derivatives of the log-likelihood function, Dümbgen and Rufibach (2009) derive various useful facts about $\widehat{\varphi}$. Here are the two most relevant ones in the present context:

A characterization. Let $\tilde{\varphi}$ be a concave function which is linear on all intervals $[x_j, x_{j+1}]$, $1 \le j < m$, while $\tilde{\varphi} = -\infty$ on $\mathbb{R} \setminus [x_1, x_m]$. Defining $\tilde{F}(x) := \int_{-\infty}^x \exp\tilde{\varphi}(r)\,\mathrm{d}r$, we assume further that $\tilde{F}(x_m) = 1$. Then $\tilde{\varphi} = \widehat{\varphi}$ and $\tilde{F} = \widehat{F}$ if, and only if, for arbitrary $t \in [x_1, x_m]$,

$$\int_{x_1}^t \tilde{F}(r)\,\mathrm{d}r \;\le\; \int_{x_1}^t \mathbb{F}(r)\,\mathrm{d}r$$

with equality in case of t being a knot of $\tilde{\varphi}$, i.e.

$$t \;\in\; \mathcal{S}(\tilde{\varphi}) := \{x_1, x_m\} \cup \{t \in (x_1, x_m) : \tilde{\varphi}'(t-) > \tilde{\varphi}'(t+)\}.$$

This characterization entails that $\widehat{F}$ is rather close to $\mathbb{F}$ in the sense that $\mathbb{F}(t-) \le \widehat{F}(t) \le \mathbb{F}(t)$ for any knot t of $\widehat{\varphi}$. It is also relevant for our algorithms, because

$$H(t, \tilde{\varphi}) := \int_{x_1}^t (\tilde{F} - \mathbb{F})(r)\,\mathrm{d}r \;=\; \frac{d}{du}\Big|_{u=0} L(\tilde{\varphi} + u\Delta_t) \tag{2}$$

with $\Delta_t(x) := \min(x - t, 0)$. Suitable functions to compute $H = H(\cdot, \widehat{\varphi})$ are implemented in **logcondens**, see Section 5.

Further properties. Another remarkable property of $\widehat{F}$ and $\mathbb{F}$ is the inequality

$$\int h(x)\,\mathrm{d}\widehat{F}(x) \;\leq\; \int h(x)\,\mathrm{d}\mathbb{F}(x) \quad \text{for arbitrary convex } h : \mathbb{R} \to \mathbb{R}.$$

In particular, setting $h(x) = \pm x$ and $h(x) = x^2$ yields

$$\mathrm{Mean}(\widehat{F}) \;=\; \mathrm{Mean}(\mathbb{F}) \;=\; \sum_{j=1}^{m} w_j x_j = \bar{X} \tag{3}$$

and

$$\mathrm{Var}(\widehat{F}) \;\leq\; \mathrm{Var}(\mathbb{F}). \tag{4}$$

4. Smoothing the log-concave density estimator

From Dümbgen and Rufibach (2009, Theorem 2.1) we know that $\widehat{\varphi}$ is equal to 0 outside $[x_1, x_m]$, i.e. $\widehat{\varphi}$ has potentially sharp discontinuities at both ends of its support. In addition, the estimator $\widehat{\varphi}$ may have quite sharp kinks; compare the mode of $\widehat{f}$ in Figure 1. To overcome these minor inconveniences and for additional reasons elaborated in the analysis of the reliability data in Section 8, Dümbgen and Rufibach (2009) introduce a *smoothed log-concave density estimator* $\widehat{f}^*$, defined for some bandwidth $\gamma > 0$ as

$$\widehat{f}^*(x) \;=\; \int_{-\infty}^{\infty} \phi_\gamma(x-y)\widehat{f}(y)\,\mathrm{d}y,$$

i.e. the convolution of $\widehat{f}$ with a Gaussian kernel ϕ_γ with mean 0 and standard deviation γ. By virtue of a celebrated result of Prékopa (1971), it is known that the class of log-concave densities is closed under convolution, whence $\widehat{f}^*$ is log-concave, too. Due to the simple structure of $\widehat{\varphi}$ and our restriction to Gaussian kernels, one can deduce explicit formulae for $\widehat{f}^*(x)$ and its distribution function $\widehat{F}^*(x)$ at any x, see Section 5.2 and Appendix C.

Choice of bandwidth. Let $\widehat{F}^*$ be the distribution function of the smoothed density $\widehat{f}^*$. Note that $\widehat{F}^*$ is the distribution function of $X + Z$ with independent random variables $X \sim \widehat{F}$ and $Z \sim N(0, \gamma^2)$. Thus it follows from (3) that all distribution functions $\widehat{F}^*, \widehat{F}, \mathbb{F}$ have the same mean $\bar{X}$, whereas

$$\mathrm{Var}(\widehat{F}^*) \;=\; \mathrm{Var}(\widehat{F}) + \gamma^2 \;\leq\; \mathrm{Var}(\mathbb{F}) + \gamma^2,$$

according to (4). Now consider the estimator

$$\widehat{\sigma}^2 \;=\; n(n-1)^{-1}\,\mathrm{Var}(\mathbb{F}) \;=\; n(n-1)^{-1}\sum_{j=1}^{m} w_j (x_j - \bar{X})^2.$$

In the standard setting, this is an unbiased estimator of $\mathrm{Var}(X_i)$. Hence, as proposed by Dümbgen and Rufibach (2009, Section 3), the default value for γ is the square root of

$$\widehat{\gamma}^2 \;=\; \widehat{\sigma}^2 - \mathrm{Var}(\widehat{F}).$$

This yields an estimated distribution function $\widehat{F}^*$ with standard deviation $\widehat{\sigma}$. Using $\gamma = \widehat{\gamma}$ also makes $\widehat{f}^*$ a *fully automatic* estimator.

The variance of $\widehat{F}$ can be computed explicitly as follows:

$$\begin{aligned}\mathrm{Var}(\widehat{F}) &= \int_{x_1}^{x_m} (x - \bar{X})^2 \widehat{f}(x)\,\mathrm{d}x \\ &= \sum_{j=2}^{m} \Delta x_j \Big((x_{j-1} - \bar{X})^2 J_{10}(\widehat{\varphi}_{j-1}, \widehat{\varphi}_j) + (x_j - \bar{X})^2 J_{10}(\widehat{\varphi}_j, \widehat{\varphi}_{j-1}) \\ &\qquad - (\Delta x_j)^2 J_{11}(\widehat{\varphi}_{j-1}, \widehat{\varphi}_j) \Big).\end{aligned}$$

Here J_{10} equals the partial derivative of $J(r, s)$ with respect to r, while J_{11} is the partial derivative of J with respect to r and s, see Dümbgen *et al.* (2010, Section 2) for a derivation and computational details.

5. Implementation and main functions

5.1. Function to compute the estimate

Using the simulated data from the introduction we now describe the function to estimate a log-concave density and the corresponding `summary` and `plot` methods.

The primary function of the package is `logConDens` which returns an object of class `dlc`. A `dlc` object is a list consisting of

- `xn`: the vector $(X_i)_{i=1}^n$ of original observations,
- `x`: the vector $(x_j)_{j=1}^m$ of support points,
- `w`: the vector $(w_j)_{j=1}^m$ of weights,
- `phi`: the estimated vector $\widehat{\varphi} = (\widehat{\varphi}(x_j))_{j=1}^m$,
- `IsKnot`: the vector $(1\{x_j \text{ is a knot of } \widehat{\varphi}\})_{j=1}^m$,
- `L`: the value $L(\widehat{\varphi})$,
- `Fhat`: a vector $(\widehat{F}(x_j))_{j=1}^m$ with values of the c.d.f. of $\widehat{f}$,
- `H`: a vector $(H(x_j))_{j=1}^m$ of directional derivatives (cf. (2) in Section 3),

as generated by `activeSetLogCon`. If `smoothed = TRUE` in the call of `logConDens` then `dlc` additionally contains the entries

- `f.smoothed`: A vector that contains the values of $\widehat{f}^*$ either at an equidistant grid of 500 values ranging from $\underline{\mathbf{x}} - 0.1r$ to $\overline{\mathbf{x}} + 0.1r$ for $\underline{\mathbf{x}} = \min_{j=1,\ldots,m} x_j$, $\overline{\mathbf{x}} = \max_{j=1,\ldots,m} x_j$, and $r = \overline{\mathbf{x}} - \underline{\mathbf{x}}$ if `xs = NULL` or the value of $\widehat{f}^*$ at the values of `xs` if a vector is provided as the latter argument.
- `F.smoothed`: The values of $\widehat{F}_n^*$ on the grid.

- `gam`: The computed value of $\widehat{\gamma}$.
- `xs`: Either the vector of 500 grid points or the initial vector `xs` of points at which `f.smoothed` is to be computed.

Finally, the entry `smoothed` returns the value of the initial argument. Note that the knots collected in `IsKnot` are not derived from $\widehat{\varphi}$ but appear as a direct by-product of the active set algorithm. The quantities H_j can be used to verify the characterization of the estimator in terms of distribution function as elaborated in Dümbgen and Rufibach (2009). See Section 3 for an illustration.

A `summary` and `plot` method are available for the class `dlc`. The `summary` method provides some basic quantities of the estimates whereas the `plot` method is intended to provide a quick way of standard plotting $\widehat{\varphi}$, $\widehat{f}$, and $\widehat{F}$ and $\widehat{f}^*$, $\widehat{F}^*$ (optional). For more tailor-made plots we recommend to extract the necessary quantities from the `dlc` object and plot them with the desired modifications, as we do when generating Figure 3.

Note that to plot $\widehat{f}$ and $\widehat{F}$ we do not simply linearly interpolate the points $\exp\widehat{\varphi}(t)$ for $t \in \widehat{\mathcal{S}}_m(\widehat{\varphi})$ but we rather compute $\widehat{f}$ and $\widehat{F}$ on a sufficiently fine grid of points using the function `evaluateLogConDens`, to display the shape of the estimated density and distribution function between knot points correctly.

5.2. Evaluation of the fitted estimators

Thanks to the simple structure of $\widehat{\varphi}$, closed formulae can be derived for $\widehat{\varphi}, \widehat{f}, \widehat{F}, \widehat{F}^{-1}$ as well as $\widehat{f}^*, \widehat{F}^*$ on their respective domain, see Section 2 and the appendix for details. These formulae are implemented in **logcondens** via the two functions `evaluateLogConDens` and `quantilesLogConDens`.

To compute the process $H = H(\cdot, \widehat{\varphi})$ in (2) one needs to be able to compute the integral of the distribution functions $\widehat{F}$ and $\mathbb{F}$, at an arbitrary point t. Again, exploiting the structure of $\widehat{\varphi}$ a closed formula can be derived for $\int_{x_1}^t \widehat{F}(r)\,\mathrm{d}r$, see Appendix B. This formula is implemented in the function `intF` in the package **logcondens**, and we use this function together with its companion `intECDF` (ECDF: empirical cumulative distribution function) for the empirical distribution function to compute the process D.

6. Sampling from the different estimators

Cule *et al.* (2010) describe Monte Carlo estimation of functionals of $\widehat{f}$ (which may be one- or multidimensional) by sampling from a random vector (variable) $\widehat{\mathbf{X}}$ that has density $\widehat{f}$ and plugging in these samples into the functionals of interest. In order to sample from $\widehat{f}$, they use a rejection sampling procedure, see Cule *et al.* (2010, Section 7.1), implemented as the function `rlcd` in Cule *et al.* (2009). In **logcondens** we implemented sampling not only from $\widehat{f}$, but also from $\widehat{f}^*$ as the function `rlogcon`, using the quantile function $\widehat{F}^{-1}$ implemented in `quantilesLogConDens`. Note that by making use of the quantile function $\widehat{F}^{-1}$ in `rlogcon` we generate a sample of a given size from $\widehat{f}$ *directly*, without the need of rejecting (some of) the generated numbers. There is no need to implement the quantile function $\widehat{F}^*$ for the sake of simulating samples from $\widehat{F}^*$, thanks to the fact that $\widehat{f}^*$ is the density of the convolution of $\widehat{f}$ with a given normal kernel.

Sampling from the estimate may serve at least two purposes: First, as described above, generated samples can be used to approximate functionals of $\widehat{f}$ or $\widehat{f}^*$, as described in Cule *et al.* (2010). Second, in certain applications it may be the explicit goal to get simulated samples from an estimated density, see Section 8 below and Dümbgen and Rufibach (2009, Section 3).

7. Illustration of main functions on simulated example

To demonstrate the code in **logcondens** we take up the sample generated in Section 1.2 (denoted as the object `x.sim`). Note that the `plot` method for a `dlc` object is already illustrated in that introductory example. Here, we detail application of the functions discussed in Section 5. To get a summary of the estimated densities invoke

```
R> res <- logConDens(x.sim, smoothed = TRUE, print = FALSE)
R> summary(res)

Estimation of a log-concave density from i.i.d. data

Number of initial observations: n = 40

Number of unique observations (or grid points): m = 40

log-likelihood: -2.18

Maximum likelihood estimate:
Mode: x[16] = -0.056
Value of log-density at mode: -0.76
Value of density at mode: 0.47

Smoothed maximum likelihood estimate:
Mode: 0.27
Value of log-density at mode: -0.81
Value of density at mode: 0.45

Number of knots of the MLE: 4

Knots of the MLE:
x[1, 16, 31, 40] =
-2.215, -0.056,  0.763,  1.595
```

Compare the displayed numbers, such as mode, value at the mode, and knots of the log-density to Figure 1. To compute the value of all the estimated functions we invoke:

```
R> evaluateLogConDens(xs = -1, res, which = 1:5)

     xs log-density density    CDF smooth.density smooth.CDF
[1,] -1      -1.765  0.1712 0.1164         0.1793      0.124
```

```
R> quantilesLogConDens(ps = 0.5, res)

      ps quantile
[1,] 0.5   0.1688
```

To illustrate the characterization of the estimators provided in Section 3, we use the functions `intF` and `intECDF` to compute the process $H(t)$, see Figure 2. The upper plot in Figure 2 displays $\mathbb{F}$ and $\widehat{F}$ whereas in the lower plot H is provided. The graphs illustrate that (1) $\widehat{F}$ closely approximates $\mathbb{F}$, a fact that can be made precise, see Dümbgen and Rufibach (2009, Corollary 2.5 and Theorem 4.4), (2) the kinks of $\widehat{\varphi}$ occur where the process H is equal to zero as postulated in Section 3 (compare Figures 1 and 2).

```
R> n <- res$n
R> xn <- res$xn
R> ss <- sort(unique(c(x.sim, seq(min(x.sim), max(x.sim), length = 200))))
R> H1 <- intF(ss, res)
R> H2 <- intECDF(ss, xn)
R> Ht <- H1 - H2
R> ED <- 0 : (n + 1) / n
R> par(mfrow = c(2, 1), mar = c(0, 5, 2, 1), mex = 1, las = 1)
R> plot(x.sim, res$Fhat, type = 'n', xaxt = 'n', ylab = 'Distribution
+      functions')
R> rug(x.sim); lines(x.sim, res$Fhat, col = 2, lwd = 1.5)
R> lines(c(min(xn) - 10, xn, max(xn) + 10), ED, type = 's', lwd = 1.5)
R> abline(v = res$knots, lty = 3); par(mar = c(4.5, 5, 0, 1))
R> legend(-2.1, 1, c("empirical distribution function", expression("CDF based
+      on "*hat(f)[m])), lty = 1, col = 1:2, lwd = 2, bty = "n")
R> plot(ss, Ht, type = 'n', xlab = "generated random sample x",
+      ylab = "process H(t)", yaxt = "n")
R> lines(ss, Ht, col = 2, lwd = 1.5)
R> ax <- -c(0.02, 0.01, 0); axis(2, at = ax , labels = ax, cex.axis = 0.8)
R> rug(x.sim); abline(v = res$knots, lty = 3); abline(h = 0, lty = 1)
```

8. Exemplary analysis of reliability dataset

To further demonstrate the functions in **logcondens** we will apply them to analyze a real-life dataset that had previously been used to illustrate log-concave density estimation. The reliability data was collected as part of a consulting project at the Institute for Mathematical Statistics and Actuarial Science at the University of Bern, see also Dümbgen and Rufibach (2009, Section 3). Basic descriptive statistics of the dataset are provided in Table 1. Note that in this dataset we have $n = 786$ original observations. However, pooling tied observations and introducing weights as described in Section 2 we end up with only $m = 535$ unique observations.

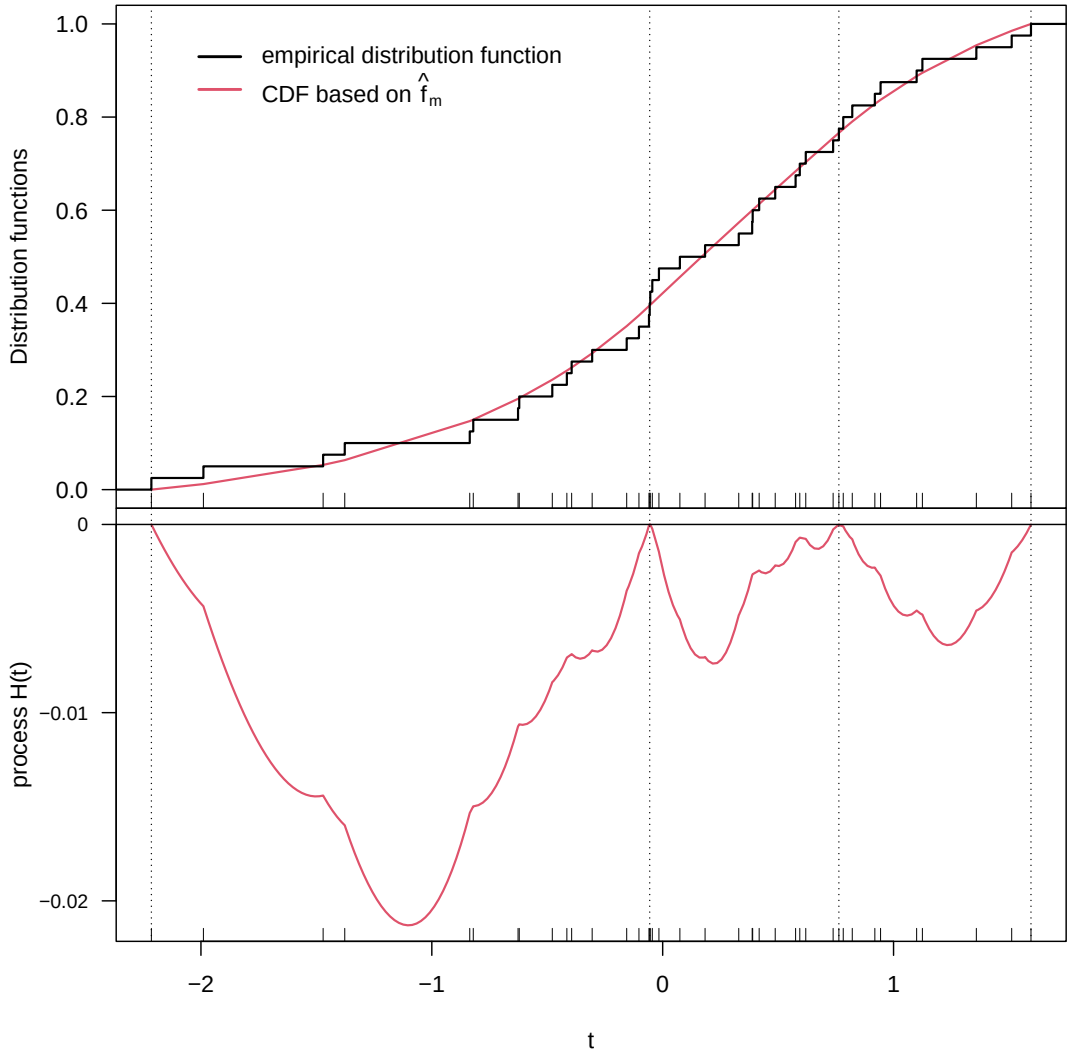

Figure 2: Distribution functions and the process $H(t)$ for the simulated data. Vertical bars at the bottom of the plots indicate observations.

A company asked for Monte Carlo experiments to predict the reliability of a certain device they produce, where the reliability depended in a certain deterministic way on five different and independent random input parameters. The dataset `reliability` in **logcondens** contains only data on the first of these input parameters. The goal was to fit a suitable distribution to this sample that can be used to simulate from. As elaborated in Dümbgen and Rufibach (2009, Section 3) we first considered two standard approaches to estimate the unknown density f: namely

(i) fitting a Gaussian density $\widehat{f}_{\mathrm{par}}$ with mean $\mu(\mathbb{F})$ and variance $\widehat{\sigma}^2 = n(n-1)^{-1}\,\mathrm{Var}(\mathbb{F})$,

(ii) the kernel density estimator

$$\widehat{f}_{\mathrm{ker}}(x) \;:=\; \int \phi_{\widehat{\sigma}/\sqrt{n}}(x-y)\,\mathrm{d}\mathbb{F}(y),$$

Variable	n	Minimum	1^{st} Quartile	Median	3^{rd} Quartile	Maximum	IQR
reliability	786	1404.2	1632.2	1688.2	1735.6	1853.6	103.4

Table 1: Descriptive statistics of reliability data.

where ϕ_σ denotes the Normal density with mean 0 and variance σ^2.

The very small bandwidth $\widehat{\sigma}/\sqrt{n}$ was chosen to obtain a density with variance $\widehat{\sigma}^2$ and to avoid putting too much weight into the tails, which was crucial in the engineers' application. We think that in general such undersmoothed density estimators are a simple way of depicting the empirical distribution of raw data.

Looking at the data, approach (i) is clearly inappropriate because the reliability sample of size $n = 786$ reveals a skewed and pronouncedly non-gaussian distribution. This can be seen in Figure 3, where the multimodal curve corresponds to $\widehat{f}_{\text{ker}}$, while the dashed line depicts $\widehat{f}_{\text{par}}$. Approach (ii) yielded Monte Carlo results agreeing well with measured reliabilities, but the engineers were not satisfied with the multimodality of $\widehat{f}_{\text{ker}}$. Choosing a kernel estimator with larger bandwidth would overestimate the variance and put too much weight into the tails. Thus we agreed on a third approach and estimated f by the smoothed log-concave estimator $\widehat{f}^*$ introduced in Section 4.

This density estimator is the skewed *unimodal* curve in Figure 3. Apart from the advantages described above – unimodality, variance equal to $\widehat{\sigma}^2$, and not too heavy tails – it yielded convincing results in the Monte Carlo simulations, too. In addition and as expected, the kinks and the discontinuities at x_1 and x_m of $\widehat{f}$ are smoothed out by $\widehat{f}^*$. Note that both estimators $\widehat{f}$ and $\widehat{f}^*$ are fully automatic.

Now, the primary goal of this data analysis was to provide simulated samples from the estimated densities $\widehat{f}$ and $\widehat{f}^*$. The function **rlogcon** that implements sampling from the distribution that corresponds to $\widehat{f}$ in the package **logcondens** is based on the quantile function $\widehat{F}^{-1}$, derived in Appendix A and implemented as `quantilesLogConDens`. Thanks to the fact that $\widehat{f}^*$ is the density of the convolution of $\widehat{f}$ with a Gaussian kernel with standard deviation γ, a random number X^* from $\widehat{f}^*$ can easily be obtained by computing $X^* = X + Y$ with independent random variables $X \sim \widehat{F}$ and $Y \sim N(0, \gamma^2)$.

To illustrate how the functions in **logcondens** can be used to explore different estimates, we provide below the code that is used to generate Figure 3. Note that this is the same Figure already displayed in Dümbgen and Rufibach (2009, Figure 2).

```
R> x.rel <- sort(reliability)
R> n <- length(x.rel)
R> mu <- mean(x.rel); sig <- sd(x.rel)
R> xs <- seq(1350, 1950, length.out = 500)
R> res <- logConDens(x.rel, smoothed = TRUE, print = FALSE, xs = xs)
R> f.smoothed <- res$f.smoothed
R> xs2 <- xs[(xs >= min(x.rel)) & (xs <= max(x.rel))]
R> f <- rep(NA, length(xs2))
R> for (i in 1:length(xs2)){f[i] <- evaluateLogConDens(xs2[i],
+      res)[, "density"]}
R> h <- sig / sqrt(n)
```

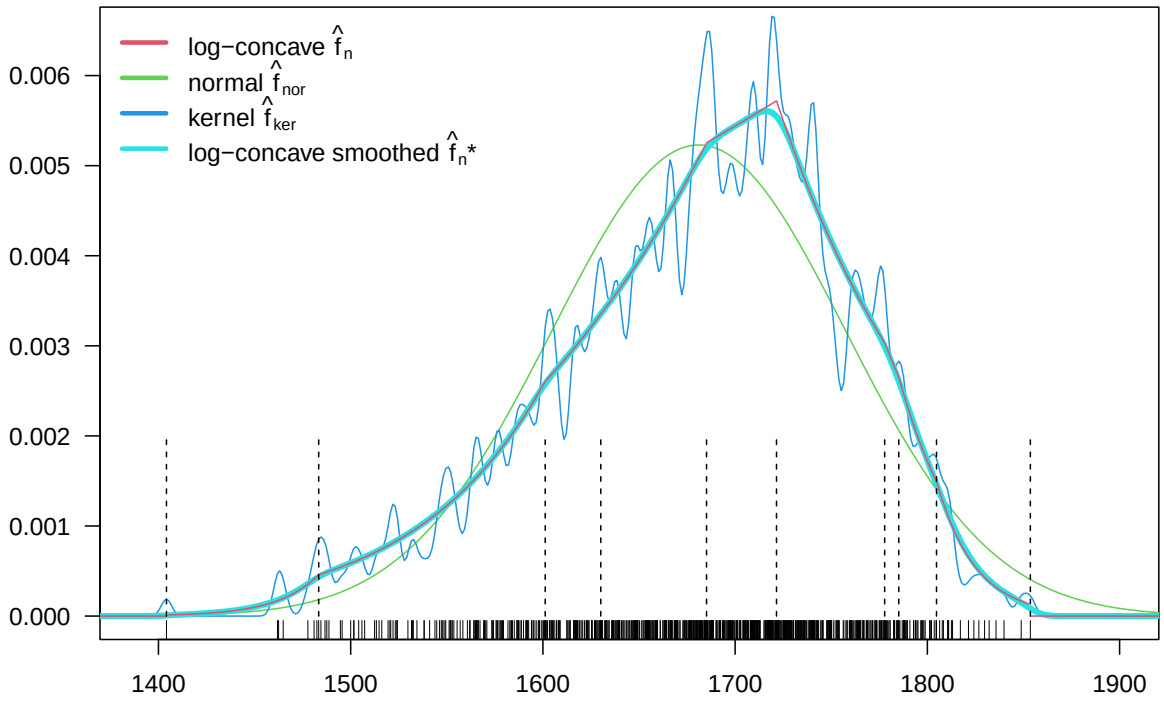

Figure 3: Different estimates of the density of the reliability data.

```
R> f.kernel <- rep(NA, length(xs))
R> for (i in 1:length(xs)){f.kernel[i] <- mean(dnorm(xs[i], mean =
+      x.rel, sd = h))}
R> f.normal <- dnorm(xs, mean = mu, sd = sig)
R> par(las = 1, mar = c(3, 3.5, 0.5, 0.5))
R> plot(0, 0, type = 'n', xlim = c(1390, 1900), ylim =
+      c(0, 6.5 * 10^-3), ylab = "")
R> rug(x.rel)
R> lines(xs, f.normal, col = 3)
R> lines(xs, f.kernel, col = 4)
R> lines(xs, f.smoothed, lwd = 4, col = 5)
R> lines(xs2, f, col = 2)
R> segments(c(-1300, max(x.rel)), c(0, 0), c(min(x.rel), 2000),
+      c(0, 0), col = 2)
R> legend("topleft", c(expression("log-concave "*hat(f)[n]),
+      expression("normal "*hat(f)[nor]), expression("kernel "*hat(f)[ker]),
+      expression("log-concave smoothed "*hat(f)[n]*"*")),
+      lty = 1, lwd = 3, col = 2:5, bty = "n")
R> segments(res$knots, 0, res$knots, 0.002, lty = 2)
```

Finally, we illustrate how we can efficiently generate samples from $\widehat{f}$ and $\widehat{f}^*$ and thus content the engineers that initiated this research. In addition, samples from the estimators could be used to approximate some functional of $\widehat{f}$ or $\widehat{f}^*$, as described in Cule *et al.* (2010).

```
R> set.seed(1977)
```

```
R> rel_samples <- rlogcon(n = 20, x0 = x.rel)
```

The sorted sample of size $n = 20$ from the log-concave density estimator can be extracted using

```
R> sort(rel_samples$X)
```

```
 [1] 1511.954 1594.168 1621.873 1645.044 1663.611 1673.327 1674.606
 [8] 1675.477 1689.173 1689.405 1696.573 1697.265 1710.816 1712.501
[15] 1716.976 1724.642 1730.252 1783.473 1792.702 1807.486
```

and one from the smoothed log-concave density estimator via

```
R> sort(rel_samples$X_star)
```

```
 [1] 1514.480 1599.645 1615.953 1648.505 1662.295 1663.369 1677.193
 [8] 1677.583 1685.057 1686.471 1689.579 1705.722 1707.009 1712.771
[15] 1718.095 1727.095 1727.765 1780.637 1785.171 1813.476
```

9. Smooth two-sample permutation test

Dümbgen and Rufibach (2009, Theorem 4.4) have shown that the distribution function estimators $\mathbb{F}$ and $\widehat{F}$ are, under mild assumptions, asymptotically equivalent and $\widehat{F}$ acts in this sense as a smoother of $\mathbb{F}$. However, by imposing log-concavity efficiency gains in estimation seem plausible for small to moderate sample sizes. This leads us to the proposal of a *smooth test for distribution functions* which we now describe.

Suppose the researcher is given two samples $(X_i)_{i=1}^{n_1}$ and $(Y_i)_{i=1}^{n_2}$ of independent random variables $X_i \sim F_X$ and $Y_j \sim F_Y$ with unknown distribution functions F_X, F_Y.

To test whether $H_0 : F_X = F_Y$ versus $H_1 : F_X \neq F_Y$, a commonly used two-sample test statistic is the Kolmogorov-Smirnov statistic, comparing the empirical distribution functions $\mathbb{F}_X$ and $\mathbb{F}_Y$ of the two samples:

$$\mathbb{K} = \mathbb{K}(\mathbb{F}_X, \mathbb{F}_Y) \ := \ \big(n_1 n_2/(n_1+n_2)\big)\big)^{1/2} \|\mathbb{F}_X - \mathbb{F}_Y\|_\infty$$

where $\|f\|_\infty := \sup_{x\in\mathbb{R}} |f(x)|$ for any function $f : \mathbb{R} \to \mathbb{R}$. The limiting distribution of $\mathbb{K}_n$ and the corresponding asymptotic test can be found in Durbin (1973).

If one imposes that F_X and F_Y both have log-concave density functions, we propose the following modified test statistic:

$$\widehat{K} = \widehat{K}(\widehat{F}_X, \widehat{F}_Y) \ = \ \big(n_1 n_2/(n_1+n_2)\big)\big)^{1/2} \|\widehat{F}_X - \widehat{F}_Y\|_\infty$$

where $\widehat{F}_X$ and $\widehat{F}_Y$ are the log-concave distribution function estimators of F_X and F_Y. Deriving the limiting distribution of this statistic is a difficult task, but if one assumes that under H_0 the pooled sample $(Z_1, Z_2, \ldots, Z_{n_1+n_2}) := (X_1, \ldots, X_{n_1}, Y_1, \ldots, Y_{n_2})$ has the same distribution as

$$(Z_{\Pi_1}, \ldots, Z_{\Pi_{n_1}}, Z_{\Pi_{n_1+1}}, \ldots, Z_{\Pi_{n_1+n_2}})$$

where Π is a random permutation of $\{1, \ldots, n_1 + n_2\}$ not depending on the data one can perform a Monte Carlo permutation test of H_0 as follows: Generate M independent copies of Π and calculate the corresponding values of the test statistic $\widehat{K}^{(1)}, \widehat{K}^{(2)}, \ldots, \widehat{K}^{(M)}$. Then a nonparametric p-value for H_0 is given by

$$\widehat{p} \;=\; \frac{1 + \#\{i \le M \,:\, \widehat{K}^{(i)} \ge \widehat{K}\}}{1 + M}.$$

The null hypothesis H_0 is rejected if p is not larger than the pre-specified significance level α. Details on the computation of $\widehat{K}(\widehat{F}_X, \widehat{F}_Y)$ as well as for $\widehat{K}^*_n(\widehat{F}_X, \widehat{F}_Y)$, the statistic based on the *smoothed* log-concave distribution function estimates, are provided in Appendix E.

Computation of differences between distribution functions is implemented in the function `maxDiffCDF` in the **logcondens** package. To generate two samples from a Gamma distribution and invoke the smooth two sample test that is based on the function `maxDiffCDF` use the code:

```
R> set.seed(1)
R> n1 <- 20
R> n2 <- 25
R> x <- sort(rgamma(n1, 2, 1))
R> y <- sort(rgamma(n2, 2, 1) + 0.5)
R> twosample <- logconTwoSample(x, y, M = 5, display = FALSE)
R> twosample$p.value

[1] 0.3333 0.3333

R> twosample$test.stat.orig

[1] 0.3716 0.3652

R> twosample$test.stats[1:5, ]

       [,1]    [,2]
[1,] 0.1453 0.07292
[2,] 0.1606 0.08579
[3,] 0.2418 0.23463
[4,] 0.3336 0.33189
[5,] 0.6606 0.64047
```

So in this example we get p-values of 0.33 and 0.33, respectively, quantifying the evidence against the null hypothesis of equal distribution functions. The first of these p-values results from a permutation test based on the log-concave MLE whereas the latter is computed via the smoothed log-concave density estimate. Note that we have chosen the (too) small number of permutations $M = 5$ only in this illustrative example. Clearly, in applications we recommend to set M to at least 1000, as we do in our simulations below.

To assess the performance of this nonparametric permutation test in terms of power compared to the Kolmogorov-Smirnov test we conducted a simulation study considering the following settings:

$$\begin{array}{lllll}
\text{Setting 1:} & n_1 = 20 & n_2 = 25 & X \sim \mathrm{N}(0,1) \text{ vs. } Y \sim \mathrm{N}(\mu,1) \\
\text{Setting 2:} & n_1 = 20 & n_2 = 25 & X \sim \mathrm{Gam}(2,\,1) \text{ vs. } Y \sim \mathrm{Gam}(2,1) + \mu \\
\text{Setting 3:} & n_1 = 20 & n_2 = 25 & X \sim \mathrm{Gam}(2,\,1) \text{ vs. } Y \sim \mathrm{Gam}(\tau,1)
\end{array}$$

for $\mu \in \{0, 0.5, 1, 1.5, 2\}$ and $\tau \in \{1, 1.5, 2, 2.5, 3\}$. We simulated each setting 1000 times and $M = 999$ was used as argument in `logconTwoSample`. The number of rejected null hypothesis when adopting a significance level of $\alpha = 0.05$ are displayed in Figure 4. We find, in accordance with Cule *et al.* (2010, Section 9, Remark (ii)), that the test based on the log-concave CDF uniformly outperforms, for the rather moderate sample sizes we consider, the Kolmogorov-Smirnov test. For very large sample sizes the difference in power between the two tests decreases (not shown). We therefore recommend to use the function `logconTwoSample` to assess the null hypothesis of two identical distribution functions, especially when only few or moderately many observations are available.

Note that this modified test is valid even if the assumption of log-concavity of the density functions is violated. Such a violation would only affect the power.

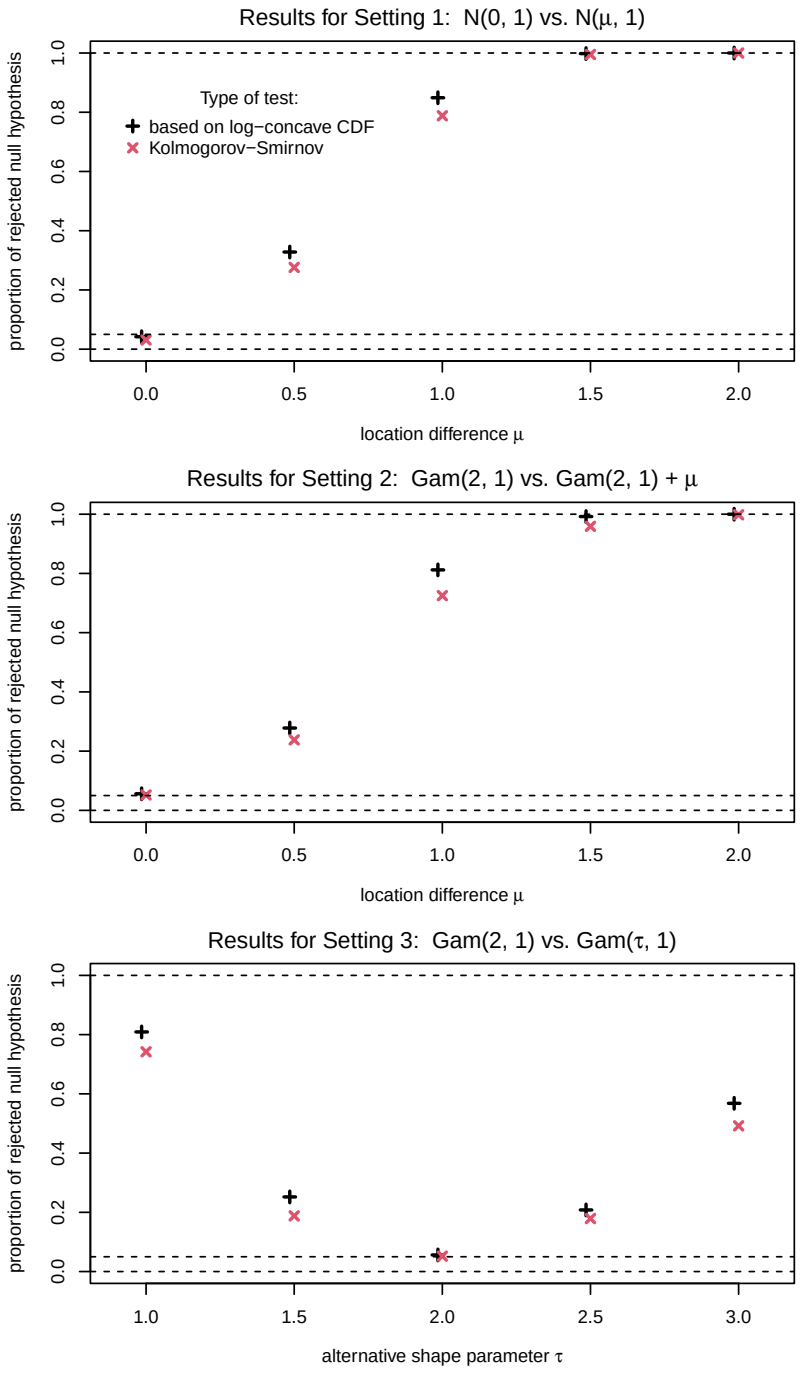

Figure 4: Proportion of rejected null hypothesis at $\alpha = 0.05$ for the null hypothesis of equal distribution functions for two samples of size $n_1 = 20$ and $n_2 = 25$. Horizontal offset of points only to increase legibility.

10. Final remarks

In Koenker and Mizera (2010, Figure 3) quasi-concave density estimation, a generalization of log-concave density estimation, was illustrated on radial and rotational velocities of the Bright Star Catalog, see Hoffleit and Warren (1991). For convenience, we also included this dataset in **logcondens** as a dataframe named `brightstar`.

A. Density, distribution and quantile function

Formulas to compute the log-density and density function at a given point $x_0 \in \mathbb{R}$ are provided in Section 2. The distribution function estimator $\widehat{F}$ and its integral $I(t) := \int_{x_1}^t \widehat{F}(r)\,\mathrm{d}r$ are implemented in the functions `evaluateLogConDens` and `intF`. For completeness, we provide here the corresponding formulas. Note that (5) is also provided in Dümbgen *et al.* (2010, Theorem 2.1). Recall from Section 2 the function J. Here, we use a slightly generalized version, defined as

$$\tilde{J}(r,s,v) \;:=\; \int_0^v \exp\bigl((1-t)r + ts\bigr)\,\mathrm{d}t$$

for arbitrary $r, s \in \mathbb{R}$ and $v \in [0,1]$. The relation $\tilde{J}(r,s,v) = \exp(r)\tilde{J}(0,s-r,v)$ holds, and

$$\tilde{J}(r,s,v) \;=\; \begin{cases} \bigl(\exp(r) - \exp(r+v(s-r))\bigr)/(r-s) & \text{if } r \neq s, \\ v\exp(r) & \text{if } r = s, \end{cases}$$

see also the manual of **logcondens**, especially the help file `Jfunctions`.

In what follows, we will use a simple generic parametric model. Namely, define for $\theta \in \mathbb{R}$ a probability density on $[0,1]$ as

$$g_\theta(x) \;:=\; J(0,\theta)^{-1}\exp(\theta x) \;=\; \begin{cases} \theta e^{\theta x}/(e^\theta - 1) & \text{if } \theta \neq 0 \\ 1 & \text{if } \theta = 0. \end{cases}$$

The distribution and quantile function corresponding to g_θ are given as

$$G_\theta(r) \;=\; \begin{cases} (e^{\theta r} - 1)/(e^\theta - 1) & \text{if } \theta \neq 0 \\ r & \text{if } \theta = 0 \end{cases}$$

and

$$G_\theta^{-1}(u) \;=\; \begin{cases} \log\bigl(1 + (e^\theta - 1)u\bigr)/\theta & \text{if } \theta \neq 0 \\ u & \text{if } \theta = 0 \end{cases}$$

for $r, u \in [0,1]$, respectively. Note that G_θ^{-1} is implemented as `qloglin` in **logcondens**. For $|\theta| \leq 10^{-6}$ a Taylor approximation is used, because

$$G_\theta^{-1}(u) \;=\; u + \theta u(1-u)/2 + O(\theta^2)$$

as $\theta \to 0$, uniformly in $u \in [0,1]$. Now, let $\varphi \in \mathbb{R}^m$ where this vector is identified with a function $\varphi : \mathbb{R} \to [-\infty, \infty)$ via

$$\varphi(x) \;:=\; \begin{cases} -\infty & \text{for } x \notin [x_1, x_m]\,, \\ \varphi_j + \dfrac{x - x_j}{\Delta x_{j+1}}\,\Delta\varphi_{j+1} & \text{for } x \in [x_j, x_{j+1}],\ 1 \leq j < m\,, \end{cases}$$

where $\Delta v_{j+1} := v_{j+1} - v_j$ for any $\mathbf{v} \in \mathbb{R}^m$. Suppose further that $f := \exp(\varphi)$ is a probability density on $\mathbb{R}$ and let F be the corresponding distribution function. Then $F(x_1) = F_1 := 0$, and

$$F(x_j) = F_j \;:=\; \sum_{i=1}^{j-1} \Delta x_{j+1} J(\varphi_i, \varphi_{i+1}) \quad \text{for } 2 \le j \le m$$

with $F_m = 1$. For a proof we refer to Dümbgen *et al.* (2010, Theorem 2.1). To compute the quantile function of F, note that for any $x \in [x_j, x_{j+1}]$ by again exploiting the special structure of φ,

$$\begin{aligned} F(x) - F(x_j) &= \int_{x_j}^{x} \exp(\varphi(t))\,\mathrm{d}t \\ &= \int_{x_j}^{x} \exp\Big(\varphi_j + \frac{x - x_j}{\Delta x_{j+1}}\,\Delta\varphi_{j+1}\Big)\,\mathrm{d}t \\ &= \Delta x_{j+1} \int_0^{(x-x_j)/\Delta x_{j+1}} \exp\Big((1-v)\varphi_j + v\varphi_{j+1}\Big)\,\mathrm{d}v \\ &= \Delta x_{j+1} \tilde{J}\Big(\varphi_j, \varphi_{j+1}, \frac{x - x_j}{\Delta x_{j+1}}\Big). \end{aligned} \tag{5}$$

Some tedious computations lead to the corresponding quantile function

$$F^{-1}(u) \;=\; x_j + \Delta x_{j+1} G^{-1}_{\Delta x_{j+1}\Delta\varphi_{j+1}}\Big((u - F_j)/\Delta F_{j+1}\Big) \quad \text{for } u \in [F_j, F_{j+1}], 1 \le j < m\,.$$

This function is the basis of `quantilesLogConDens` in **logcondens**.

B. The integral of F at an arbitrary x_0

Recall the definition $s_j = \Delta\varphi_j/\Delta x_j$ from Section 2. In addition, we define $f_j = f(x_j)$ for $j = 1, \ldots, m$. To be able to compute the process $H = H(\cdot, \widehat{\varphi})$ introduced in (2) the aim is to derive an explicit formula for

$$I_j(x) \;=\; \int_{x_j}^{x} F(r)\,\mathrm{d}r$$

for any $j = 1, \ldots, m-1$ and $x \in [x_j, x_{j+1}]$. Using (5) we can write

$$\begin{aligned} I_j(x) &= \int_{x_j}^{x} \Big(F_j + \Delta x_{j+1} \tilde{J}\Big(\varphi_j, \varphi_{j+1}, \frac{r - x_j}{\Delta x_{j+1}}\Big)\Big)\,\mathrm{d}r \\ &= (x - x_j) F_j + \Delta x_{j+1} \int_{x_j}^{x} \tilde{J}\Big(\varphi_j, \varphi_{j+1}, \frac{r - x_j}{\Delta x_{j+1}}\Big)\,\mathrm{d}r. \end{aligned}$$

But,

$$\begin{aligned} \int_{x_j}^{x} \tilde{J}\Big(\varphi_j, \varphi_{j+1}, \frac{r - x_j}{\Delta x_{j+1}}\Big)\,\mathrm{d}r &= \Delta x_{j+1} \int_0^{(x-x_j)/\Delta x_{j+1}} \tilde{J}(\varphi_j, \varphi_{j+1}, y)\,\mathrm{d}y \\ &= -s_{j+1}^{-1} \int_0^{(x-x_j)/\Delta x_{j+1}} (\exp(\varphi_j) - \exp(\varphi_j + y\Delta\varphi_{j+1}))\,\mathrm{d}y \\ &= -\frac{x - x_j}{\Delta\varphi_{j+1}} f_j + s_{j+1}^{-1} \tilde{J}\Big(\varphi_j, \varphi_{j+1}, \frac{x - x_j}{\Delta x_{j+1}}\Big). \end{aligned} \tag{6}$$

Putting the pieces together we receive

$$I_j(x) \;=\; (x-x_j)F_j + \Delta x_{j+1}\Big(s_{j+1}^{-1}\tilde{J}\Big(\varphi_j,\varphi_{j+1},\frac{x-x_j}{\Delta x_{j+1}}\Big) - \frac{x-x_j}{\Delta\varphi_{j+1}}f_j\Big).$$

Specifically, for $x = x_{j+1}$,

$$I_j(x_{j+1}) \;=\; \Delta x_{j+1}\Big[F_j + s_{j+1}^{-1}\Big(\tilde{J}(\varphi_j,\varphi_{j+1},1) - f_j\Big)\Big].$$

We finally get

$$I(t) \;=\; \int_{x_1}^{t} F(r)\,\mathrm{d}r \;=\; \Big(\sum_{i=1}^{i_0} I_i(x_{i+1})\Big) + I_{i_0}(t)$$

where $i_0 = \min\{m-1\,,\ \max\{i\ :\ x_i \le t\}\}$.

Approximation for $s_{j+1} \to 0$. Since we divide by s_{j+1} in the compuation of $I_j(x)$ this expression becomes numerically unstable or even undefined once the slope s_{j+1} of φ approaches or even equals 0. To avoid these problems, we compute the limit for $a := s_{j+1} \to 0$ in the integral in (6):

$$\begin{aligned}
\lim_{a\to 0}\int_{x_j}^{x}\tilde{J}\Big(\varphi_j,\varphi_{j+1},\frac{r-x_j}{\Delta x_{j+1}}\Big)\,\mathrm{d}r &= \lim_{a\to 0}\int_0^{(x-x_j)/\Delta x_{j+1}}\frac{\exp(\varphi_j + y\Delta\varphi_{j+1}) - \exp\varphi_j}{a}\,\mathrm{d}y\\
&= f_j\lim_{a\to 0}\int_0^{(x-x_j)/\Delta x_{j+1}}\frac{\exp(ya\Delta x_{j+1})-1}{a}\,\mathrm{d}y\\
&= f_j\int_0^{(x-x_j)/\Delta x_{j+1}} y\Delta x_{j+1}\,\mathrm{d}y\\
&= f_j\frac{(x-x_j)^2}{2\Delta x_{j+1}}.
\end{aligned}$$

This approximation is used in `intF` once $s_{j+1} \le 10^{-6}$.

C. The smoothed log-concave density estimator

For $\gamma > 0$ and $x \in \mathbb{R}$ recall the density function ϕ_γ of a centered normal density as

$$\phi_\gamma(x) \;=\; \frac{1}{\sqrt{2\pi}\gamma}\exp(-x^2/(2\gamma^2)).$$

Its distribution function is denoted by Φ_γ. Elementary calculations reveal that for arbitrary numbers a, x and $\gamma > 0$,

$$e^{ay}\phi_\gamma(x-y) \;=\; \exp(ax + a^2\gamma^2/2)\phi_\gamma(y - x - a\gamma^2),$$

so that for real boundaries $u < v$,

$$\begin{aligned}
q_\gamma(x,a,u,v) &:= \int_u^v e^{a(y-u)}\phi_\gamma(x-y)\,\mathrm{d}y\\
&= \exp(a(x-u)+a^2\gamma^2/2)(\Phi_\gamma(v-x-a\gamma^2) - \Phi_\gamma(u-x-a\gamma^2))\\
&= \exp(a(x-u)+a^2\gamma^2/2)(\Phi_\gamma(x-u+a\gamma^2) - \Phi_\gamma(x-v+a\gamma^2)). \qquad (7)
\end{aligned}$$

The smoothed log-concave density estimator then amounts to

$$\begin{aligned}
\widehat{f}^*(x) &= \int_{-\infty}^{\infty} \phi_\gamma(x-y)\widehat{f}(y)\,\mathrm{d}y \\
&= \sum_{j=2}^{m} \int_{x_{j-1}}^{x_j} \phi_\gamma(x-y)\widehat{f}(y)\,\mathrm{d}y \\
&= \sum_{j=2}^{m} \widehat{f}_{j-1} \int_{x_{j-1}}^{x_j} \exp(\widehat{s}_j(y-x_{j-1}))\phi_\gamma(x-y)\,\mathrm{d}y \\
&= \sum_{j=2}^{m} \widehat{f}_{j-1}\, q_\gamma(x, \widehat{s}_j, x_{j-1}, x_j).
\end{aligned}$$

The function `evaluateLogConDens` implements $\widehat{f}^*$ in **logcondens**. However, note that it is most convenient to compute $\widehat{f}^*$ via specifying `smoothed = TRUE` in `logConDens`.

In extreme situations, e.g. data sets containing extreme spacings, numerical problems may occur in (7). For it may happen that the exponent is rather large while the difference of Gaussian CDFs is very small. To moderate these problems, we are using the following bounds in the function `Q00` implementing q_γ in **logcondens**:

$$\exp(-m^2/2)\big(\Phi(\delta)-\Phi(-\delta)\big) \;\le\; \Phi(b)-\Phi(a) \;\le\; \exp(-m^2/2)\cosh(m\delta)\big(\Phi(\delta)-\Phi(-\delta)\big)$$

for arbitrary numbers $a < b$ and $m := (a+b)/2$, $\delta := (b-a)/2$.

D. The smoothed log-concave CDF estimator

Using partial integration, we get for arbitrary numbers x, $a \neq 0$ and real boundaries $u < v$:

$$\begin{aligned}
Q_\gamma(x,a,u,v) &:= \int_u^v e^{a(y-u)}\Phi_\gamma(x-y)\,\mathrm{d}y \\
&= \Big[a^{-1}e^{a(y-u)}\Phi_\gamma(x-y)\Big]_{y=u}^{v} + a^{-1}\int_u^v e^{a(y-u)}\phi_\gamma(x-y)\,\mathrm{d}y \\
&= +a^{-1}\big(e^{a(v-u)}\Phi_\gamma(x-v) - \Phi_\gamma(x-u)\big) + a^{-1}q_\gamma(x,a,u,v)
\end{aligned}$$

with $q_\gamma(x,a,u,v)$ as in (7). As $a \to 0$, this converges to

$$\begin{aligned}
Q_\gamma(x,0,u,v) &:= \int_u^v \Phi_\gamma(x-y)\,\mathrm{d}y \\
&= \Big[(y-x)\Phi_\gamma(x-y)\Big]_{y=u}^{v} + \int_u^v (y-x)\phi_\gamma(x-y)\,\mathrm{d}y \\
&= \Big[(y-x)\Phi_\gamma(x-y) - \gamma^2\phi_\gamma(x-y)\Big]_{y=u}^{v} \\
&= (x-u)\Phi_\gamma(x-u) - (x-v)\Phi_\gamma(x-v) + \gamma^2\big(\phi_\gamma(x-u) - \phi_\gamma(x-v)\big).
\end{aligned}$$

This leads to the formula

$$\begin{aligned}\widehat{F}^*(x) &= \int_{-\infty}^{\infty} \Phi_\gamma(x-y)\widehat{f}(y)\,\mathrm{d}y \\ &= \sum_{j=2}^{m} \int_{x_{j-1}}^{x_j} \Phi_\gamma(x-y)\widehat{f}(y)\,\mathrm{d}y \\ &= \sum_{j=2}^{m} \widehat{f}_{j-1} \int_{x_{j-1}}^{x_j} \exp(\widehat{s}_j(y-x_{j-1}))\Phi_\gamma(x-y)\,\mathrm{d}y \\ &= \sum_{j=2}^{m} \widehat{f}_{j-1}\, Q_\gamma(x, \widehat{s}_j, x_{j-1}, x_j).\end{aligned}$$

This representation is implemented in `evaluateLogConDens` and `logConDens`, where for numerical reasons, $Q_\gamma(x, \widehat{s}_j, x_{j-1}, x_j)$ is replaced with $Q_\gamma(x, 0, x_{j-1}, x_j)$ in case of $|\widehat{s}_j| \leq 10^{-6}$.

E. Computation of the two-sample test statistic

To calculate $\mathbb{K}$, the built-in R function `ks.test` was used. This function computes an exact p-value in a two-sample problem if the product of the respective sample size is smaller than 10^4, as fulfilled in our simulation examples. As to the computation of $\widehat{K}_n$, define $D := \widehat{F}_X - \widehat{F}_Y$, and let $z_1 < \ldots < z_N$ denote the sorted pooled observations $X_1, \ldots, X_{n_1}, Y_1, \ldots, Y_{n_2}$, i.e. $N = n_1 + n_2$. The goal is to find

$$x_* := \operatorname*{arg\,max}_{x\in\mathbb{R}} |D(x)|,$$

enabling us to compute $\widehat{K} = |D(x_*)|$. Note that in general D is not unimodal, let alone concave. However, again thanks to the special structure of the log-density estimate computation of x_X is possible according to the following scheme. With the order statistics $X_{(i)}$ and $Y_{(j)}$ of the two samples define

$$L := \max\{X_{(1)}, Y_{(1)}\} \quad \text{and} \quad R := \min\{X_{(n)}, Y_{(n)}\}.$$

First, we sort out a simple pathological case: If $R \leq L$, then any point x between R and L yields $|D(x)| = 1$, yielding $\widehat{K} = (n_1 n_2/(n_1+n_2)))^{1/2}$. Otherwise, due to the monotonicity of distribution functions one has

$$\operatorname*{arg\,max}_{x\in\mathbb{R}} |D(x)| = \operatorname*{arg\,max}_{x\in[L,R]} |D(x)|.$$

A necessary condition for a $x \in (L, R)$ to be a maximum of $|D(\cdot)|$ is $D'(x) = 0$, which is equivalent to $\widehat{\varphi}_X(x) = \widehat{\varphi}_Y(x)$. As for the computation of x, proceed as follows:

1. Identify all $k \in \{1, \ldots, N-1\}$ such that there exists a $y_k \in [z_k, z_{k+1}] := \mathcal{I}_k$ with $\widehat{\varphi}_X(y_k) = \widehat{\varphi}_Y(y_k)$. Let $\mathcal{K}$ denote the set of these k's.

2. Since both $\widehat{\varphi}_X$ and $\widehat{\varphi}_Y$ are linear on every interval $\mathcal{I}_k$, we can find for each k the point y_k where these two functions intersect via the equality

$$\begin{aligned}&\Big(1 - \frac{y_k - z_k}{z_{k+1} - z_k}\Big)\widehat{\varphi}_X(z_k) + \frac{y_k - z_k}{z_{k+1} - z_k}\widehat{\varphi}_X(z_{k+1}) = \\ &\quad \Big(1 - \frac{y_k - z_k}{z_{k+1} - z_k}\Big)\widehat{\varphi}_Y(z_k) + \frac{y_k - z_k}{z_{k+1} - z_k}\widehat{\varphi}_Y(z_{k+1})\end{aligned}$$

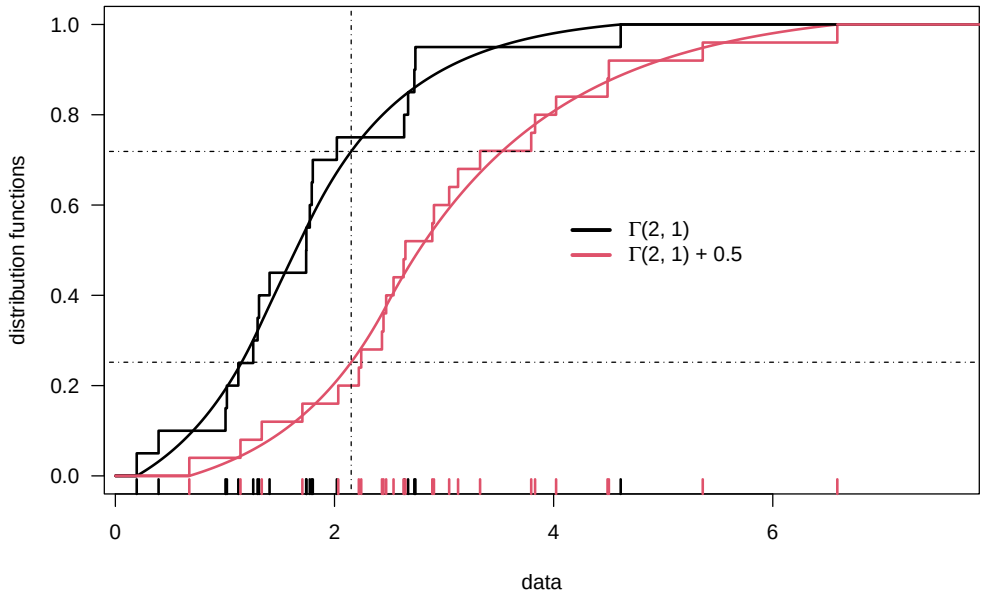

Figure 5: Estimated empirical and log-concave distribution functions for Gam(2, 1) and Gam(2, 1) + 0.5. Horizontal and vertical lines indicate location of largest difference between CDFs.

yielding

$$y_k \;=\; z_k - (\widehat{\varphi}_Y(z_k) - \widehat{\varphi}_X(z_k))\Big(\frac{\widehat{\varphi}_X(z_{k+1}) - \widehat{\varphi}_X(z_k)}{z_{k+1} - z_k} - \frac{\widehat{\varphi}_Y(z_{k+1}) - \widehat{\varphi}_Y(z_k)}{z_{k+1} - z_k}\Big)^{-1}.$$

Note that if the denominator, the difference of slopes of $\widehat{\varphi}_X$ and $\widehat{\varphi}_Y$ on $[z_k, z_{k+1}]$, is zero then the two functions must match on that interval (otherwise they would not intersect). This in turn implies that the difference D is constant on that interval so that we can consider $y_k = z_k$ a possible point where the maximum of D occurs. Consequently, $|D(\cdot)|$ is maximal at one (or more) points in the set

$$\mathcal{R} = \{L, R\} \cup \{y_k \;:\; k \in \mathcal{K}\}$$

and

$$\widehat{K}_n \;=\; \Big(n_1 n_2/(n_1 + n_2)\Big)^{1/2} \max\{D(r) \;:\; r \in \mathcal{R}\}.$$

Figure 5 gives an example.

The computation of $\widehat{K}_n^*(\widehat{F}_X, \widehat{F}_Y)$ appears to be less straightforward than that of $\widehat{K}_n(\widehat{F}_X, \widehat{F}_Y)$. In `logconTwoSample` we first approximate the function $(D^*)'_x = (\partial/\partial x)(D^*)(u, x)$ for $u = z_1 - 0.1 \cdot (z_N - z_1)$ on a sufficiently dense equidistant grid (the number of elements in that grid can be provided to `logconTwoSample` using the argument `n.grid`). Then, similar to the computation of $\widehat{K}_n(\widehat{F}_X, \widehat{F}_Y)$, we identify those grid intervals where $(D^*)'_x$ changes sign. To finally find the maximum of the latter function we then invoke `uniroot` to find the precise location of the zeros on these intervals, compute the value of $(D^*)'_x$ at each of these zeros and identify the largest of these values. In general, the computation of $\widehat{K}_n^*(\widehat{F}_X, \widehat{F}_Y)$ is more

time-consuming than that of $\widehat{K}_n(\widehat{F}_X, \widehat{F}_Y)$. However, since on the level of CDFs the MLE and its smoothed version are very similar, differences with respect to the smooth two-sample test in terms of power are very small. For these reasons we omitted the smooth version of the test in our small simulation study reported on in Section 9.

Acknowledgments

We thank Dominic Schuhmacher, an associate editor and two referees for constructive comments. The work of Lutz Dümbgen was supported by the Swiss National Science Foundation.

References

Balabdaoui F, Rufibach K, Wellner JA (2009). "Limit Distribution Theory for Maximum Likelihood Estimation of a Log-Concave Density." *The Annals of Statistics*, **37**(3), 1299–1331.

Birgé L (1997). "Estimation of Unimodal Densities Without Smoothness Assumptions." *The Annals of Statistics*, **25**(3), 970–981. ISSN 0090-5364.

Chang G, Walther G (2007). "Clustering with Mixtures of Log-Concave Distributions." *Computational Statistics & Data Analysis*, **51**, 6242–6251.

Cule M, Gramacy R, Samworth R (2009). "**LogConcDEAD**: An R Package for Maximum Likelihood Estimation of a Multivariate Log-Concave Density." *Journal of Statistical Software*, **29**(2). URL http://www.jstatsoft.org/v29/i02/.

Cule M, Samworth R, Stewart M (2010). "Maximum Likelihood Estimation of a Multidimensional Log-Concave Density." *Journal of the Royal Statistical Society B*, **72**(5), 545–607.

de Leeuw J, Hornik K, Mair P (2009). "Isotone Optimization in R: Pool-Adjacent-Violators Algorithm (PAVA) and Active Set Methods." *Journal of Statistical Software*, **32**(5), 1–24. ISSN 1548-7660. URL http://www.jstatsoft.org/v32/i05.

Dümbgen L, Hüsler A, Rufibach K (2010). "Active Set and EM Algorithms for Log-Concave Densities based on Complete and Censored Data." *Technical report*, University of Bern. Available at arXiv:0707.4643.

Dümbgen L, Rufibach K (2009). "Maximum Likelihood Estimation of a Log-Concave Density and its Distribution Function." *Bernoulli*, **15**, 40–68. ISSN 1350-7265.

Dümbgen L, Rufibach K (2010). "**logcondens**: Computations Related to Univariate Log-Concave Density Estimation." *Journal of Statistical Software*, **to appear**.

Durbin J (1973). *Distribution Theory for Tests based on the Sample Distribution Function*. Society for Industrial and Applied Mathematics, Philadelphia, Pa. Conference Board of the Mathematical Sciences Regional Conference Series in Applied Mathematics, No. 9.

Fletcher R (1987). *Practical Optimization*. Wiley, Chichester.

Groeneboom P, Wellner JA (1992). *Information Bounds and Nonparametric Maximum Likelihood Estimation*, volume 19 of *DMV Seminar*. Birkhäuser Verlag, Basel. ISBN 3-7643-2794-4.

Hoffleit D, Warren W (1991). *The Bright Star Catalog*. 5th edition. Yale University Observatory, New Heaven.

Jongbloed G (1998). "The Iterative Convex Minorant Algorithm for Nonparametric Estimation." *Journal of Computational and Graphical Statistics*, **7**(3), 310–321. ISSN 1061-8600.

Knuth DE (1984). *The TEXbook*, volume A of *Computers and Typesetting*. Addison-Wesley, Reading, Massachusetts.

Koenker R, Mizera I (2010). "Quasi-concave Density Estimation." *The Annals of Statistics*, **38**(5), 2998–3027.

Lamport L (1994). *LATEX: A Document Preparation System*. 2nd edition. Addison-Wesley, Reading, Massachusetts.

Leisch F (2002). "Dynamic Generation of Statistical Reports Using Literate Data Analysis." In W Härdle, B Rönz (eds.), *COMPSTAT 2002 – Proceedings in Computational Statistics*, pp. 575–580. Physica Verlag, Heidelberg.

Nocedal J, Wright SJ (1999). *Numerical optimization*. Springer, New York.

Prékopa A (1971). "Logarithmic Concave Measures with Application to Stochastic Programming." *Acta Universitatis Szegediensis. Acta Scientiarum Mathematicarum*, **32**, 301–316. ISSN 0001-6969.

R Development Core Team (2010). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

Rufibach K (2007). "Computing Maximum Likelihood Estimators of a Log-Concave Density Function." *Journal of Statistical Computation and Simulation*, **77**, 561–574.

Rufibach K (2009). ***OrdFacReg**: Least Squares, Logistic, and Cox-regression with Ordered Predictors*. R package version 1.0.1, URL http://CRAN.R-project.org/package=OrdFacReg.

Rufibach K (2010). "An Active Set Algorithm to Estimate Parameters in Generalized Linear Models with Ordered Predictors." *Computational Statistics & Data Analysis*, **54**(6), 1442 – 1456. doi:DOI:10.1016/j.csda.2010.01.014.

Rufibach K, Dümbgen L (2010). ***logcondens**: Estimate a Log-Concave Probability Density from iid Observations*. R package version 2.0.1, URL http://www.biostat.uzh.ch/aboutus/people/rufibach.html,http://www.staff.unibe.ch/duembgen.

Schuhmacher D, Hüsler A, Dümbgen L (2009). "Multivariate Log-Concave Distributions as a Nearly Parametric Model." *Technical report*, University of Bern. URL http://www.citebase.org/abstract?id=oai:arXiv.org:0907.0250.

Seregin A, Wellner JA (2010). "Nonparametric Estimation of Multivariate Convex-Transformed Densities." *The Annals of Statistics*. To appear.

Silverman BW (1982). "On the Estimation of a Probability Density Function by the Maximum Penalized Likelihood Method." *The Annals of Statistics*, **10**(3), 795–810. ISSN 0090-5364.

Walther G (2009). "Inference and Modeling with Log-concave Distributions." *Statistical Science*, **24**(3), 319–327.

Affiliation:

Lutz Dümbgen
Institute of Mathematical Statistics and Actuarial Science
University of Bern
3012 Bern, Switzerland
Telephone: +41/0/31631-8802
Fax: +41/0/31631-3805
E-mail: duembgen@stat.unibe.ch
URL: http://www.staff.unibe.ch/duembgen

Kaspar Rufibach
Biostatistics Unit
Institute for Social and Preventive Medicine
University of Zurich
8001 Zurich, Switzerland
Telephone: +41/0/44634-4643
Fax: +41/0/44634-4386
E-mail: kaspar.rufibach@ifspm.uzh.ch
URL: http://www.biostat.uzh.ch/aboutus/people/rufibach.html