

Reproducing results for SEND

29.05.2022

An artificial dataset of 50 samples is simulated from a bivariate standard normal distribution.

```
library(SEND)

N.sim <- 50
mu.sim <- rep(0, 2)
Sigma.sim <- diag(1, nrow = 2, ncol = 2)
simdata.norm <- MASS::mvrnorm(N.sim, mu = mu.sim, Sigma = Sigma.sim)

library(plotrix)

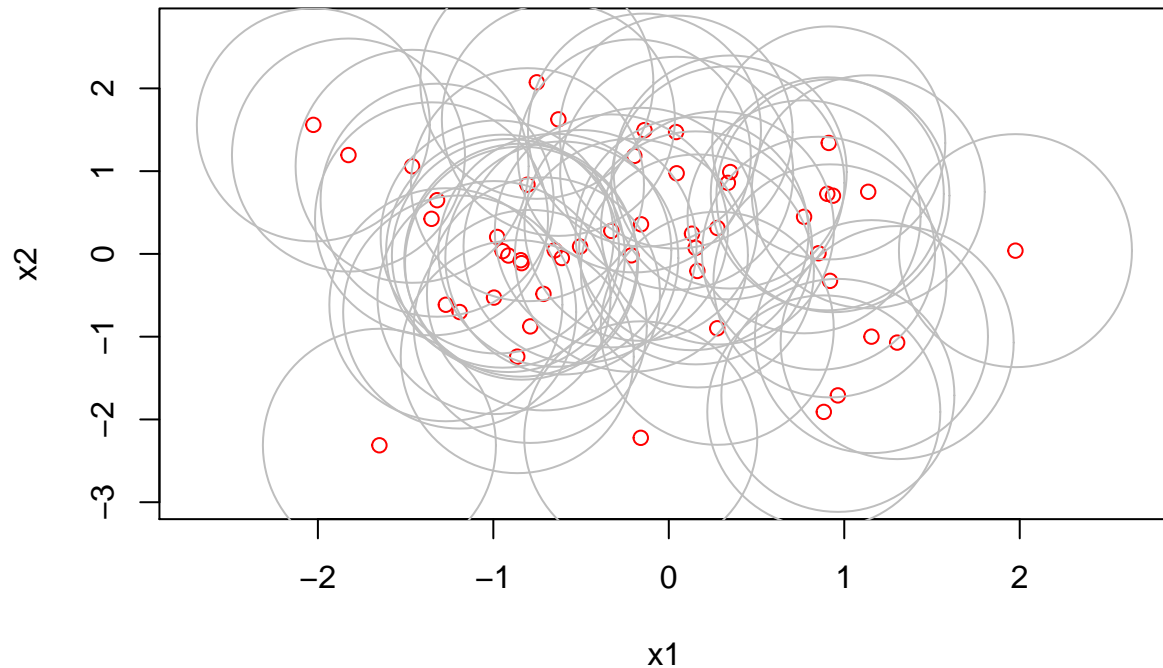
## Warning: package 'plotrix' was built under R version 4.0.5

# Estimate eps from the simulated dataset
fit.eps <- eps_estim(simdata.norm)

# adjust box scale
x.min <- min(simdata.norm[, 1]) - fit.eps$eps
x.max <- max(simdata.norm[, 1]) + fit.eps$eps
y.min <- min(simdata.norm[, 2]) - fit.eps$eps
y.max <- max(simdata.norm[, 2]) + fit.eps$eps

# Plot dataset
plot(simdata.norm, col = 'red',
      xlim = c(x.min, x.max),
      ylim = c(y.min, y.max),
      xlab = 'x1', ylab = 'x2')

# Plot neighborhood
for (i in 1:N.sim) {
  draw.circle(simdata.norm[i, 1], simdata.norm[i, 2], fit.eps$eps,
              border = 'gray')
}
```



The true value for c_α is estimated by Monte Carlo (MC) simulation with 10000 MC samples.

```
# Monte Carlo - True value
N.mc <- 10000; N.sim <- 50

# Bivariate standard normal
mu.sim <- rep(0, 2)
Sigma.sim <- diag(1, nrow = 2, ncol = 2)

# Bivariate spherical uniform
d.sim <- 2
r.sim <- 1

alarm.sim <- rep(0, N.mc)

for (i in 1:N.mc) {
  # simulation -- Bivariate standard normal
  # sample.mc <- MASS::mvrnorm(1, mu = mu.sim, Sigma = Sigma.sim)
  # simdata <- MASS::mvrnorm(N.sim, mu = mu.sim, Sigma = Sigma.sim)

  # simulation -- Bivariate spherical uniform
  sample.mc <- uniformly::runif_in_sphere(1, d = d.sim, r = r.sim)
  simdata <- uniformly::runif_in_sphere(N.sim, d = d.sim, r = r.sim)

  # estimation
  fit.eps <- eps_estim(simdata)
  alarm.sim[i] <- alarm_estim(sample.mc, simdata, fit.eps$eps)
```

```

# console logging
svMisc::progress(i, N.mc)
if (i == N.mc) message('Monte Carlo simulation -- completed')
}

```

Save output.

```

sink('simulation-MC-n50-spuniform.txt')
# ===== Begin =====
cat('Sim sample: ', N.sim, '|| MC sample: ', N.mc, '|| Dist: uniform\n')
# cat('mu: ', mu.sim, '|| Sigma:', Sigma.sim, '\n\n')
cat('dim', d.sim, '|| radius:', r.sim, '\n\n')

alpha <- c(0.10, 0.05, 0.01)
quantile(alarm.sim, probs = 1-alpha)
# ===== End =====
sink(file = NULL)
closeAllConnections()

```

Smoothed bootstrap method to approximate c_α given 300 simulated dataset.

```

# Average over 300 runs
N.avg <- 300; N.boot <- 10000; N.sim <- 50
c <- matrix(rep(0, N.avg*3), nrow = N.avg)
alpha <- c(0.10, 0.05, 0.01)

# Bivariate standard normal
mu.sim <- rep(0, 2)
Sigma.sim <- diag(1, nrow = 2, ncol = 2)

for (i in 1:N.avg) {
  # BOOTSTRAP
  simdata <- MASS::mvrnorm(N.sim, mu = mu.sim, Sigma = Sigma.sim)
  fit.eps <- eps_estim(simdata)
  alarm.boot <- sm.boot(simdata, fit.eps$eps, N.boot, N.sim)
  c[i, ] <- quantile(alarm.boot, probs = 1-alpha)

  # console logging
  svMisc::progress(i, N.avg)
  if (i == N.avg) message('Smoothed Bootstrap algorithm -- completed')
}

sink('simulation-SB-n50-stdnormal.txt')
# ===== Begin =====
cat('Sim sample: ', N.sim, '|| Bootstrap sample: ', N.boot, '|| Repeat: ', N.avg, '|| Dist: normal\n')
cat('mu: ', mu.sim, '|| Sigma:', Sigma.sim, '\n\n')
# cat('dim', d.sim, '|| radius:', r.sim, '\n\n')

alpha <- c(0.10, 0.05, 0.01)
1-alpha
colMeans(c)
# ===== End =====
sink(file = NULL)
closeAllConnections()

```

As an alternative to smoothed bootstrap method, cross-validation smoothing is used to estimate the best $\bar{\epsilon}_n$ with a prespecified grid from 0.01 to 3 with 10 points.

```
# Average over 300 runs
N.avg <- 300; N.sim <- 50
c <- matrix(rep(0, N.avg*3), nrow = N.avg)
alpha <- c(0.10, 0.05, 0.01)

# Bivariate standard normal
mu.sim <- rep(0, 2)
Sigma.sim <- diag(1, nrow = 2, ncol = 2)

for (i in 1:N.avg) {
  # CV SMOOTH
  simdata <- MASS::mvrnorm(N.sim, mu = mu.sim, Sigma = Sigma.sim)
  eps.data <- eps_estim(simdata)$eps
  c[i, ] <- cv.smooth(simdata, 0.01, 3, 10, alpha, eps.data)$alarm.cv

  # console logging
  svMisc::progress(i, N.avg)
  if (i == N.avg) message('Cross-validation Smoothing algorithm -- completed')
}

sink('simulation-CVS-n50-stdnormal.txt')
# ===== Begin =====
cat('Sim sample: ', N.sim, '|| Repeat: ', N.avg, '|| Dist: normal\n')
cat('mu: ', mu.sim, '|| Sigma:', Sigma.sim, '\n\n')
# cat('dim', d.sim, '|| radius:', r.sim, '\n\n')

alpha <- c(0.10, 0.05, 0.01)
1-alpha
colMeans(c)
# ===== End =====
sink(file = NULL)
closeAllConnections()
```