README2

Tobias Castelberg

July 27, 2022

Optimistic Circular Binary Segmentation for Change Point Detection

This package provides code for fast and robust change point detection using circular binary segmentation with reduced evaluations of the CUSUM statistic and its counterpart for the L1 loss function.

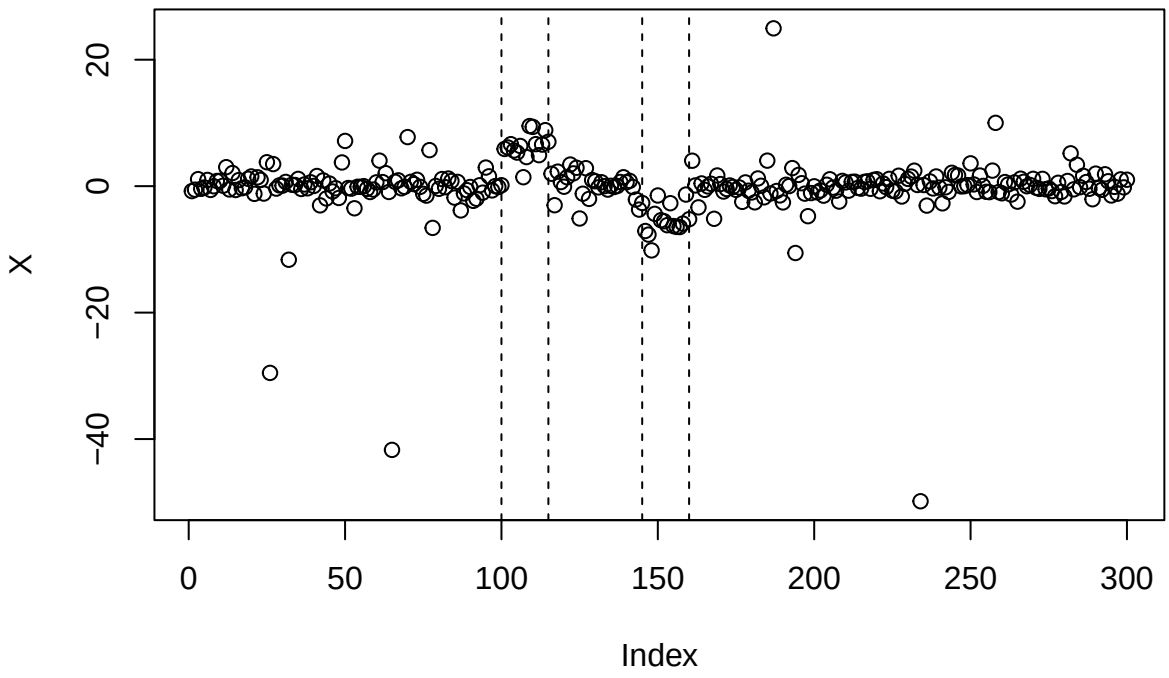
Installation from GitHub

Type `devtools::install_github("TobiasCastelberg/OCBS")` in the console to install the package and load it as usual with `library(OCBS)`.

Example on how to use the package

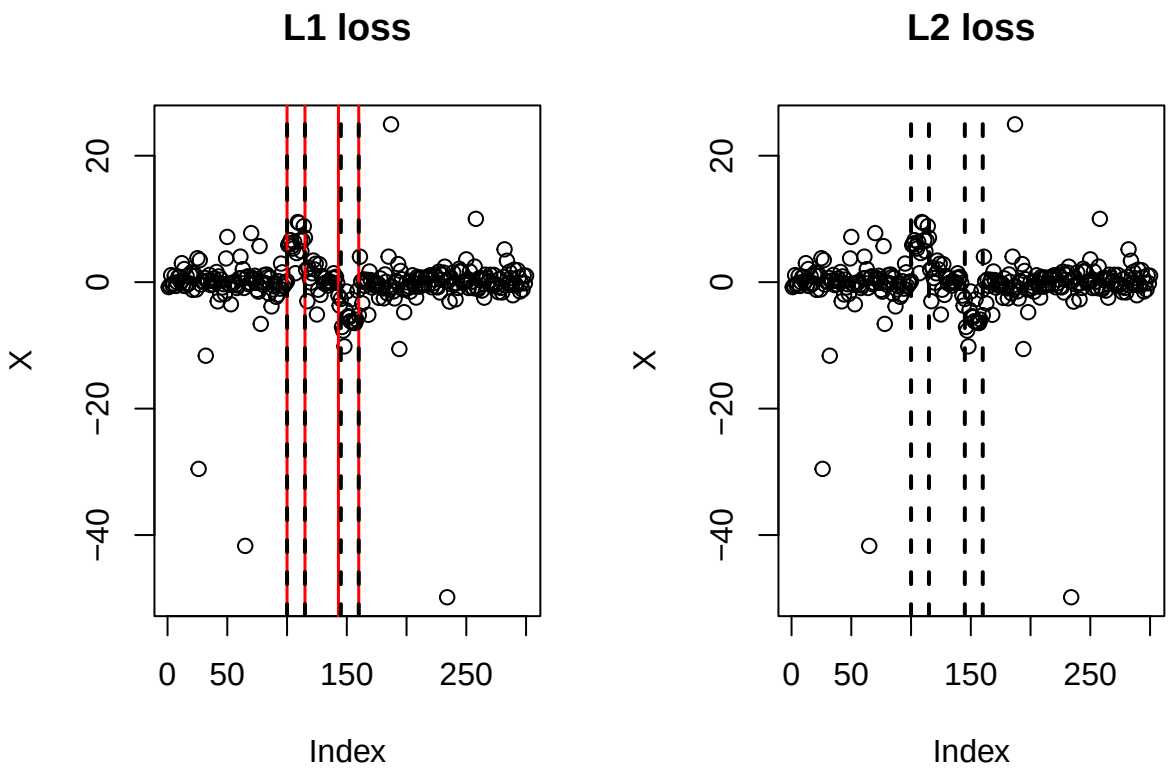
We provide an example on how to use the package with simulated data where we know the true change points.

```
set.seed(1)
X <- rep(c(0,6,0,-6,0),c(100,15,30,15,140)) + rt(300,2)
cps_true <- c(100,115,145,160)
plot(X)
abline(v=cps_true, lty=2)
```



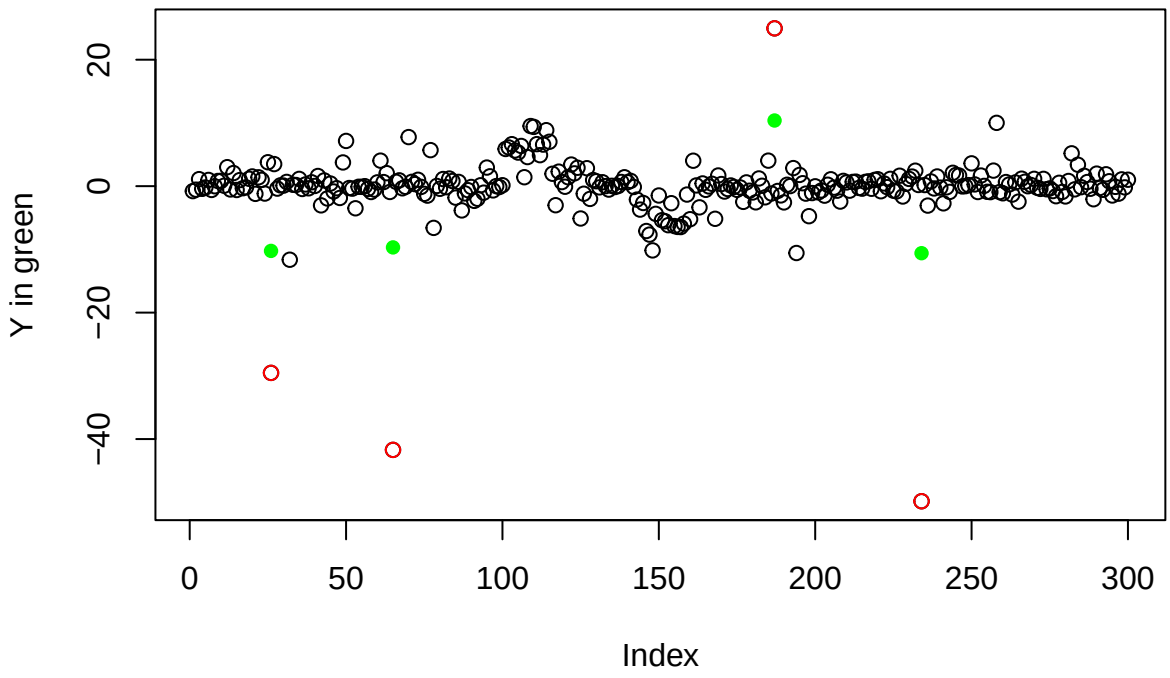
We now estimate the change points using L^1 and L^2 optimization.

```
cpsL1 <- OCBS::optimisticCBS(X, optimization = "L1", method = "advanced")
cpsL2 <- OCBS::optimisticCBS(X, optimization = "L2", method = "advanced")
par(mfrow=c(1,2))
plot(X); title("L1 loss")
abline(v=cpsL1, col="red", lty=1, lwd=1.5)
abline(v=cps_true, lty=2, lwd=2)
plot(X); title("L2 loss")
abline(v=cpsL2, col="blue", lty=1, lwd=1.5)
abline(v=cps_true, lty=2, lwd=2)
```



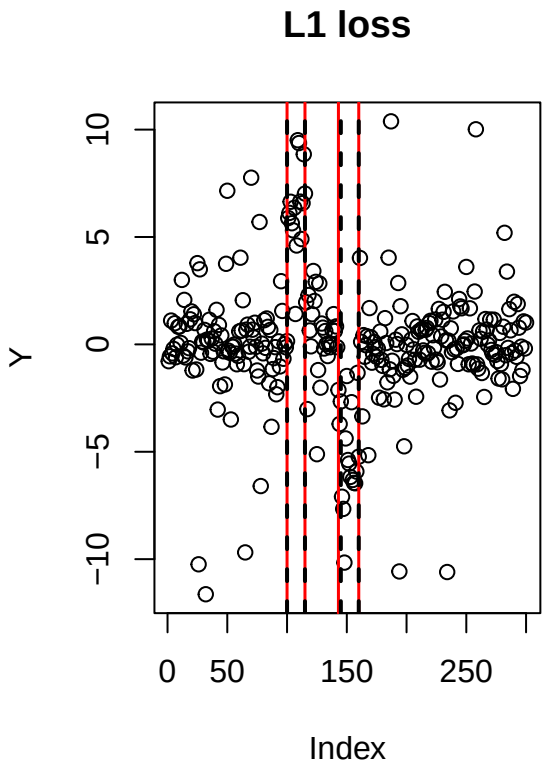
The L^2 algorithm failed to find the anomalies. The noise structure with t_3-distribution is too long tailed for the L^2 loss. We can improve the detection by removing outliers before using `optimisticCBS`.

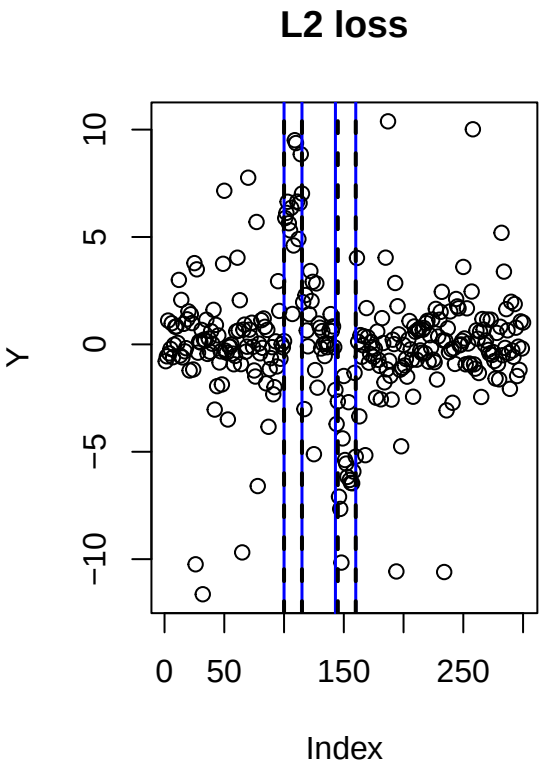
```
Y <- OCBS::SmoothData(X)
plot(X, ylab = "Y in green")
points(which(X!=Y),Y[which(X!=Y)], col="green", pch=16)
points(which(X!=Y),X[which(X!=Y)], col="red")
```



Now that the outliers are removed, also the L^2 algorithm will find the change points.

```
cpsL1 <- OCBS::optimisticCBS(Y, optimization = "L1", method = "advanced")
cpsL2 <- OCBS::optimisticCBS(Y, optimization = "L2", method = "advanced")
par(mfrow=c(1,2))
plot(Y); title("L1 loss")
abline(v=cpsL1, col="red", lty=1, lwd=1.5)
abline(v=cps_true, lty=2, lwd=2)
plot(Y); title("L2 loss")
abline(v=cpsL2, col="blue", lty=1, lwd=1.5)
abline(v=cps_true, lty=2, lwd=2)
```


L1 loss
Y
10
5
0
−5
−10
0
50
150
250
Index


L2 loss
Y
10
5
0
−5
−10
0
50
150
250
Index