Multivariate Change point Models for NO2

Introduction

```r
library(InspectChangepoint)
library(hdbinseg)
library(changepoint.geo)
```

```r
source("../R/univaraite_nsp.R")
source("../R/utils.R")
source("../R/pre_process_clean_data.R")
source("../R/multivaraite_nsp/__init__.R")
```

Load time series data for the 400 largest cities in Europe.

```r
n.cities <- 400
largest.cities <- read.csv("../data/countries_and_country_codes/largest_cities/largest_cities.csv",
                           row.names = 1
                           )[,1][1:n.cities]

xx <- stabilize.varaince.detrend(path.to.data = "../data/clean/NO2/",
                                 cities = largest.cities,
                                 target.year = "2020",
                                 trend.fit.years = "2018",
                                 show.prog.bar = TRUE
                                 )
```

Competing methods

Assuming a pieicewise constsant mean did not seem appropriate for the nuivariate time series. Nevertheless below I apply chanegpoint methods appropriate for detecting changes in pieicewise constant means. The Inspect alforithm is overly sensitive to noise and returns a very large number of changeoint locations.

```r
other.methods <- fit.multivaraute.changepoints(t(xx$whitened))

par(mfrow = c(2,2))

for (ii in 1:length(other.methods))
{
  df.plot.with.dates.axis(xx$whitened, main = names(other.methods)[ii], col = "grey")
  for (jj in other.methods[[ii]]) abline(v = jj, col = "blue")
}
```

Of the three methods that return a sensible number of changepoints there is some disagreement between the changepoint locations. Most European countries began implementing nationwide lockdowns in March and we would therefore expect to detect a change some time after.

```r
paste("geometric mapping:", paste(rownames(xx$whitened)[other.methods$`geometric mapping`], collapse =
paste("double cusum:", paste(rownames(xx$whitened)[other.methods$`double cusum`], collapse = ", "))
paste("sparsified bs:", paste(rownames(xx$whitened)[other.methods$`sparsified binary segmentation`], col
```

We can apply the same algorithms to first differenced data.

```
yy <- apply(xx$whitened, 2, diff)
other.methods.diff <- fit.multivaraute.changepoints(t(yy))

par(mfrow = c(2,2))

for (ii in 1:length(other.methods.diff))
{
  df.plot.with.dates.axis(yy, main = names(other.methods.diff)[ii], col = "grey")
  for (jj in other.methods.diff[[ii]]) abline(v = jj, col = "blue")
}
```

Again check for agreements in the detected changepoints… there is none!

```
paste("inspect:", paste(rownames(yy)[other.methods.diff$inspect], collapse = ", "))
paste("geometric mapping:", paste(rownames(yy)[other.methods.diff$`geometric mapping`], collapse = ", "
paste("double cusum:", paste(rownames(yy)[other.methods.diff$`double cusum`], collapse = ", "))
paste("sparsified bs:", paste(rownames(yy)[other.methods.diff$`sparsified binary segmentation`], collaps
```

Narrowest Significance Pursuit

```
thresh <- MC.thresh(n = dim(xx$whitened)[2],
                    tt = dim(xx$whitened)[1],
                    alpha = 0.05,
                    K = 500)
```

NSP with pieiceise constant means

```
m.nsp.pcwsConst <- panel_nsp(t(xx$whitened), thresh = thresh, deg = 0)
```

```
df.plot.with.dates.axis(xx$whitened, col = "grey", main = "NSP with piecewise constant means")
draw_mnsp_rects(m.nsp.pcwsConst$nsp.out, c(-100,100))
```

```
m.nsp.pcwsConst
```

Try isolating only the L2 aggregation

```
m.nsp.pcwsConst.L2 <- panel_nsp(t(xx$whitened), thresh = c(Inf, thresh[2]), deg = 0)
```

```
df.plot.with.dates.axis(xx$whitened, col = "grey", main = "(L2 aggregation only)")
draw_mnsp_rects(m.nsp.pcwsConst.L2$nsp.out, c(-100,100))
```

```
m.nsp.pcwsConst.L2
```

NSP with piecewise linear means

```
m.nsp.pcwsLin <- panel_nsp(t(xx$whitened), thresh = thresh, deg = 1)
```

```
df.plot.with.dates.axis(xx$whitened, col = "grey", main = "NSP with piecewise constant means")
draw_mnsp_rects(m.nsp.pcwsLin$nsp.out, c(-100,150))
```

```
m.nsp.pcwsLin
```

And again isolating only the L2 aggregations

```r
m.nsp.pcwsLin.L2 <- panel_nsp(t(xx$whitened), thresh = c(Inf, thresh[2]), deg = 1)
```

```r
df.plot.with.dates.axis(xx$whitened, col = "grey", main = "(L2 aggregation only)")
draw_mnsp_rects(m.nsp.pcwsLin.L2$nsp.out, c(-100,150))
```

```r
m.nsp.pcwsLin.L2
```