Emotional dynamics in online videos: Integrating machine learning and content analysis

Sara Major Aleksandar Tomašević

Prepare data

We analyze this video of Aleksandar Vučić. It's a media conference after decision of Kosovo government to not permit Serbian licence plates.

https://www.youtube.com/watch?v=knhZJcISbWQ

The video was processed in our previous paper (Major and Tomašević 2023). We can import the resulting CSV file from GitHub

```
gh_url <- "https://raw.githubusercontent.com/atomashevic/face-of-populism/main/data-clean/
vid <- read.csv(gh_url)
head(vid[1:9])
```

```
  X                 box0 angry0 disgust0 fear0 happy0 sad0 surprise0 neutral0
1 0 [581, 161, 137, 165]   0.65        0  0.03   0.14 0.10      0.00     0.08
2 1 [639, 106, 141, 187]   0.01        0  0.00   0.00 0.02      0.00     0.97
3 2 [596, 132, 146, 191]   0.01        0  0.01   0.07 0.01      0.00     0.89
4 3 [603, 144, 161, 219]   0.01        0  0.01   0.69 0.03      0.01     0.25
5 4 [570, 146, 147, 198]   0.12        0  0.15   0.01 0.18      0.04     0.50
6 5 [632, 144, 142, 188]   0.12        0  0.04   0.02 0.06      0.05     0.71
```

Each row of `vid` is a frame of the video and we have total of 268 frames.

From the `vid` we extract the time series of:

```
ts_mv = cbind(vid$angry0,vid$disgust0,vid$fear0,vid$happy0,vid$sad0,vid$surprise0, vid$neu

ts_ng = vid$angry0 + vid$disgust0 + vid$sad0 + vid$fear0 #<2>
```

```
ts_nt = vid$neutral0 #<3>
```

① all emotions,
② negative emotions
③ and neutral expression

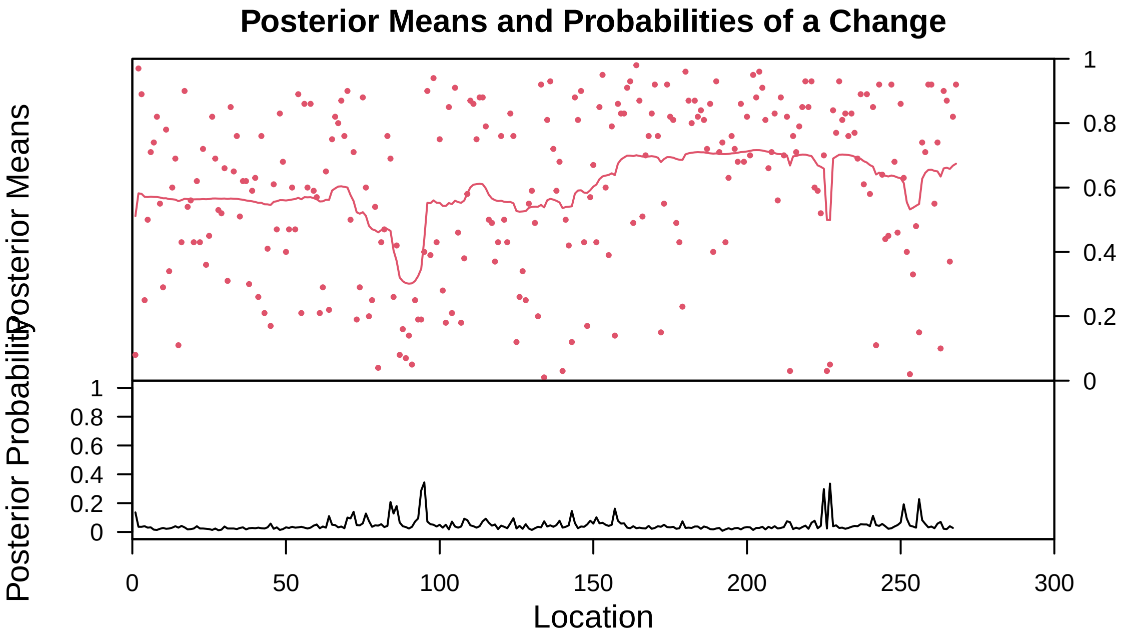
Change-point detection

We take three different approaches to change point detection in order to assure the robustness of our results.

Bayesian change-point detection `bcp`

Neutral expressions

```
bcp_nt <- bcp(ts_nt)
plot(bcp_nt)
```



```
bcp_res = which(bcp_nt$posterior.prob>0.8) #<1
```

```
if (length(bcp_res) == 0) {
  print("No change points detected")
} else {
    print(paste("Number of change points detected:", length(bcp_res)))
    paste("Frames:",bcp_res)
}
```

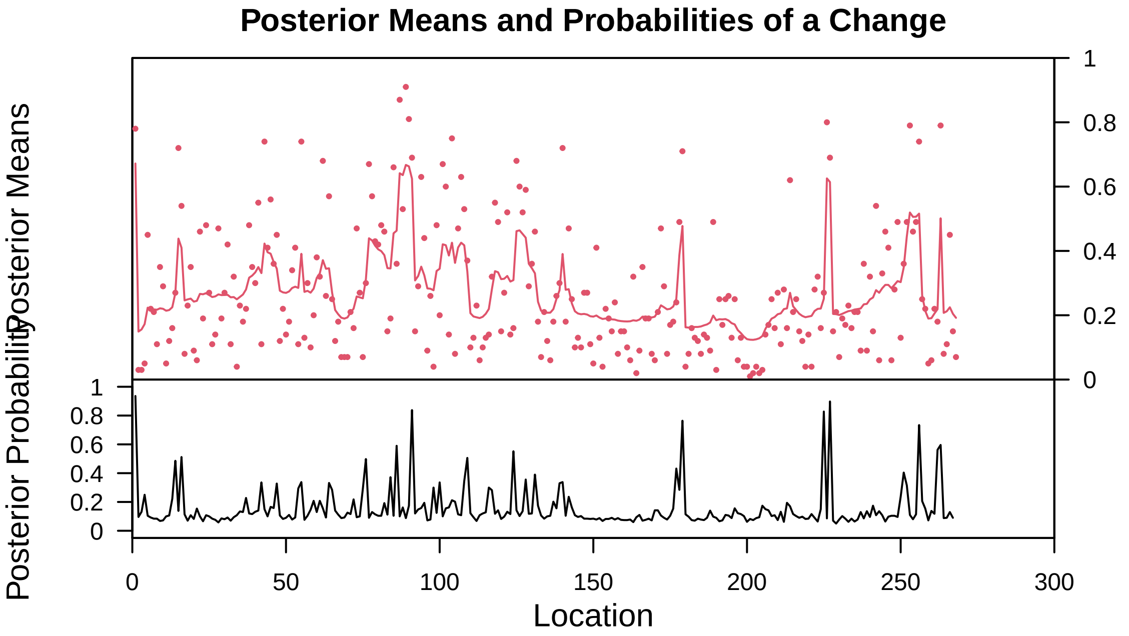
```
[1] "No change points detected"
```

1. We set the of posterior probability of change point occurring in a specific frame is greater than 0.8.

In case of neutral expressions, `bcp` detects no change points with posterior probability greater than 0.8.

Negative emotions

```
bcp_ng <- bcp(ts_ng)
plot(bcp_ng)
```



```
bcp_res = which(bcp_ng$posterior.prob>0.8) #<1>
if (length(bcp_res) == 0) {
  print("No change points detected")
} else {
    print(paste("Number of change points detected:", length(bcp_res)))
    paste("Frames:",bcp_res)
}
```

① We set the of posterior probability of change point occurring in a specific frame is greater than 0.8.

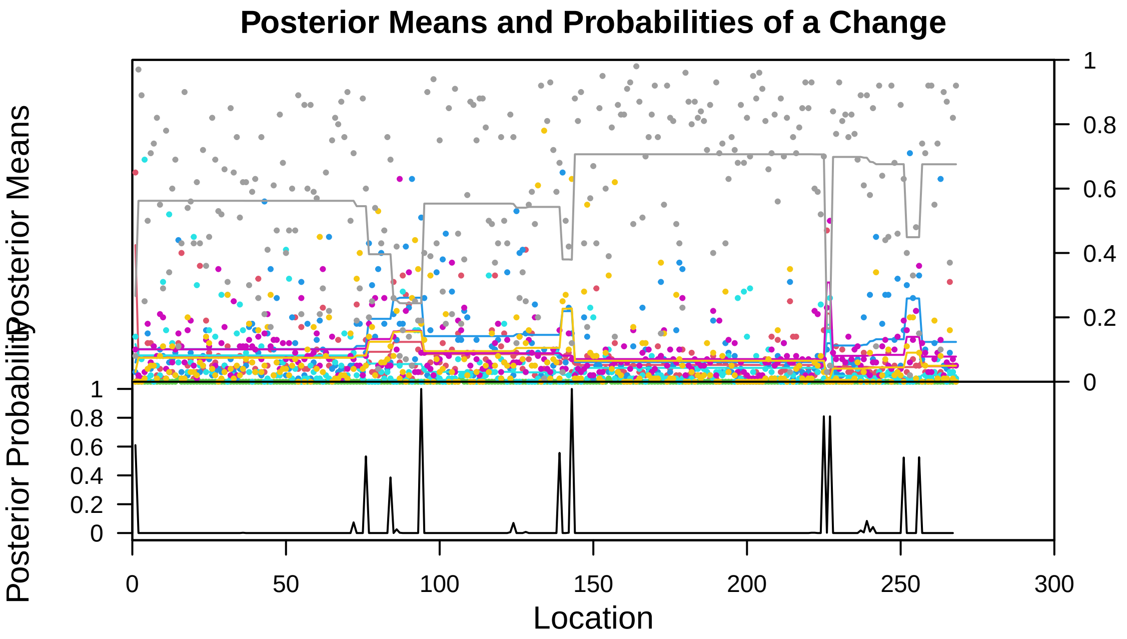
```
[1] "Number of change points detected: 4"
[1] "Frames: 1"   "Frames: 91"  "Frames: 225" "Frames: 227"
```

In case of negative emotions, we have detected change points in frames: 1, 91, 225, 227.

All emotions

Finally, with `bcp` we can perform multivariate change-point detection on all emotions.

```
bcp_mv <- bcp(ts_mv)
plot(bcp_mv)
```



```
bcp_res = which(bcp_mv$posterior.prob>0.8) #<1>
if (length(bcp_res) == 0) {
  print("No change points detected")
} else {
    print(paste("Number of change points detected:", length(bcp_res)))
    paste("Frames:",bcp_res)
}
```

① We set the of posterior probability of change point occurring in a specific frame is greater than 0.8.

```
[1] "Number of change points detected: 4"
[1] "Frames: 94"  "Frames: 143" "Frames: 225" "Frames: 227"
```

In case of all emotions, we have detected change points in frames: 94, 143, 225, 227.

High dimensional change point detection `hdcd`

We use `hdcd` to detect change points in the time series of all emotions.

```
tree <- hdcd::hdcd(ts_mv, method = "glasso", optimizer = "section_search")
```

```
Warning in hdcd::hdcd(ts_mv, method = "glasso", optimizer = "section_search"):
Lambda for glassocd set by asymptotic theory to 0.00253990031187346
```

```
print('Change points')
```

```
[1] "Change points"
```

```
hdcd::get_change_points_from_tree(tree)
```

```
[1] 157 213
```

We have two change points detected: 157 and 213.

SoloCP `solocp`

We use SoloCP approach for neutral and negative emotions.

Neutral expressions

```
sigma <- sd(ts_nt) #<1>

scp_nt <- solocp_single(ts_nt, sigma)

print('Change points:')
which(scp_nt$ratio>0.2) #<2>
```

① We estimate the standard deviation of the time series.
② TODO Explain what ratio is!

```
[1] "Change points:"
[1] 228 268
```

We have detected change points in frames: 228, 268.

Negative emotions

```
sigma <- sd(ts_ng) #<1>

scp_ng <- solocp_single(ts_ng, sigma)

print('Change points:')
which(scp_ng$ratio>0.2) #<2>
```

① We estimate the standard deviation of the time series.
② TODO Explain what ratio is!

```
[1] "Change points:"
[1]   1 180 264
```

We have detected change points in frames: 1, 180, 264.

Summary

When we combine the results of all three change point detection methods we get the following change points.

```
cps <- c(76, 94, 157, 180, 213, 227, 264)
```

We can convert them to minutes of the video.

```
cps_m <- sapply(cps, frame_to_min)
```

Graphs

Time series of all 6 emotions

Before we plot time series we need to prepare x-axis to be in minutes.

```
x <- seq(0, nrow(vid), 1) #<1>
x_m <- sapply(x, frame_to_min)
```

① We create a sequence of numbers from 0 to the number of frames in the video.

Now we need to prepare a data frame of rolling means for each emotion.

```
emos <- as.data.frame(matrix(0, nrow = nrow(ts_mv)-11, ncol = 6)) #<1>

for (i in 1:6){
  emos[,i] <- rollmean(ts_mv[,i], 12, align = "right") #<2>
}

colnames(emos) <- c('angry', 'disgust', 'fear', 'happy', 'sad', 'surprise')

cols <- c(cpal[7],cpal[5],cpal[1],cpal[8],cpal[6],cpal[4]) #<3>
```

① Since we will take rolling mean of 12 frames (roughly corresponding to one-minute period), the resulting data frame will have 11 rows less than the original time series data frame.
② We taking rolling mean of 12 values, aligned with the right, which means that we take into account emotional states expressed in the last minute (12 frames, spacing between the frames is roughly 5 seconds) when evaluating the current frame (right alignment).
③ We assign colors to each emotion.

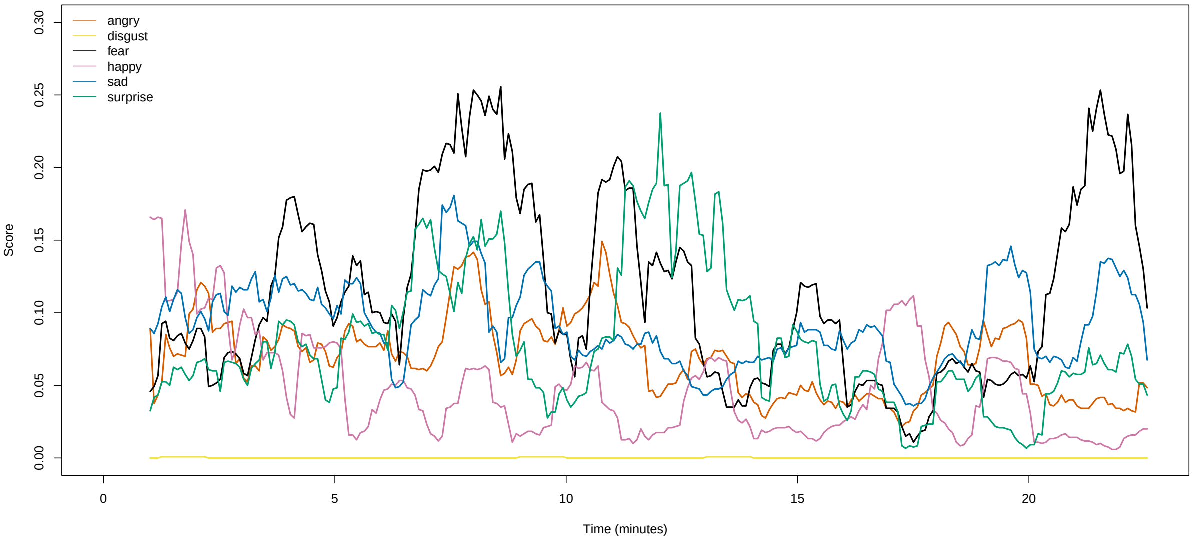
Now we can plot the time series of all emotions.

```
plot(NULL, ylim=c(0,0.3),xlim=c(0,max(x_m)),
xlab='Time (minutes)',
ylab='Score') #<1>

for (i in 1:6){
    lines(x_m[13:269],(as.vector(t(emos[,i]))),col = cols[i], lwd=2, lty=1) #<2>
}

legend("topleft", legend = colnames(emos), col = cols[1:6], lty = 1, bty = "n") #<3>
```

① We plot an empty plot with the x-axis ranging from 0 to the maximum number of minutes in the video and y-axis ranging from 0 to 0.3.
② We plot the time series of each emotion.
③ We add a legend to the plot.



Annotated time series of negative emotions and neutral expression

First, we need to create rolling averages of negative emotions and neutral expression.

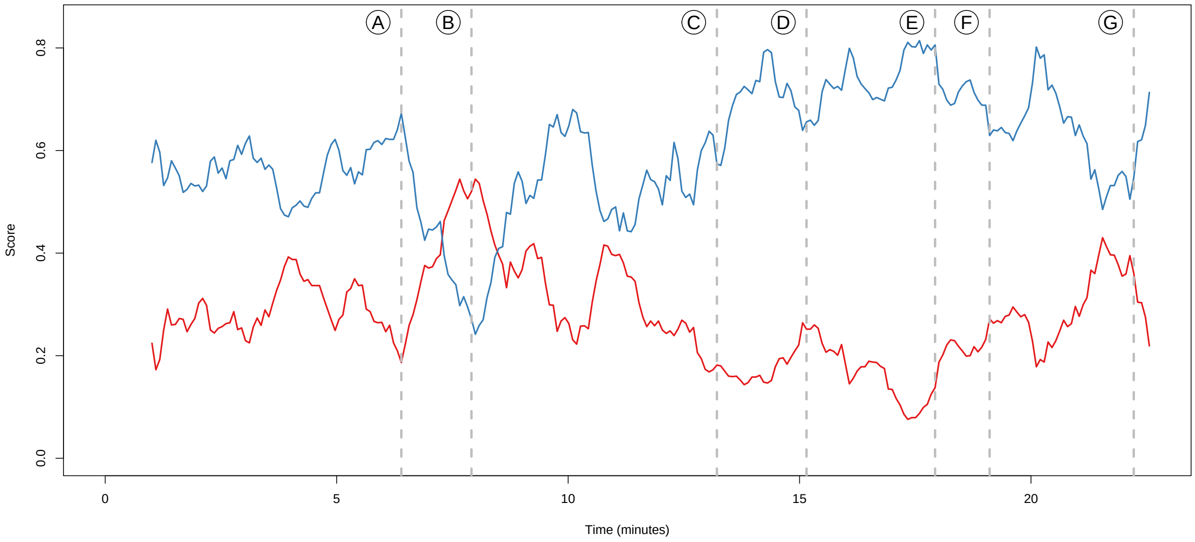
```
ng = rollmean(ts_ng, 12, align = 'right')
nt = rollmean(ts_nt, 12, align = 'right')
```

We will plot the time series of positive and negative emotions with the change points annotated.

```r
plot(NULL, ylim=c(0,0.85),xlim=c(0,max(x_m)),
xlab='Time (minutes)',
ylab='Score') #<1>

lines(x_m[13:269],(as.vector(t(ng))),col = cpal2[1], lwd=2, lty=1) #<2>
lines(x_m[13:269],(as.vector(t(nt))),col = cpal2[2], lwd=2, lty=1) #<3>
for (i in 1:length(cps_m)){
    abline(v=cps_m[i], col="grey", lwd=3, lty=2) #<4>
    points(cps_m[i]-0.5, 0.85, pch=21, bg="white", cex=4.14)
    text(cps_m[i]-0.5, 0.85, paste(LETTERS[i]), cex=1.5, col="black") #<5>
}
```

① We plot an empty plot with the x-axis ranging from 0 to the maximum number of minutes in the video and y-axis ranging from 0 to 0.85.
② We plot the time series of negative emotions.
③ We plot the time series of neutral expression.
④ We add grey vertical lines at the change points.
⑤ We add annotations to designate change points A through G.



Table

Finally, we create a table with the change points and descriptions.

Table 1: Change points

lts	cps	ms	desc
A	76	6m20s	Rise of negative emotions to its maximum
B	94	7m51s	Decline of negative emotions
C	157	13m9s	Rise of neutral expression
D	180	15m6s	Continual rise of neutral expression to its maximum
E	213	17m52s	Burst of negative emotions
F	227	19m3s	Continual rise of negative emotions
G	264	22m10s	Rise of neutral expression, decline of negative emotions

It has to contain: 1. Change point symbol (A-G) 2. Change point frame number 3. Change point time in minutes 4. Description of the change point (what happens in terms of negative/neutral dynamics)

```
lts <- c('A', 'B', 'C', 'D', 'E', 'F', 'G')
ms <- c()
for (i in 1:length(cps_m)){
    ms[i] <- paste(floor(cps_m[i]),"m" ,floor((cps_m[i] - floor(cps_m[i]))*60) - 3, "s", s
}
```

Now, let's give a short description for each change point.

```
desc <- c(
  'Rise of negative emotions to its maximum',
  'Decline of negative emotions',
  'Rise of neutral expression',
  'Continual rise of neutral expression to its maximum',
  'Burst of negative emotions',
  'Continual rise of negative emotions',
  'Rise of neutral expression, decline of negative emotions'
)
```

```
table <- data.frame(lts, cps, ms, desc)
kable(table, caption = "Change points")
```

Major, Sara, and Aleksandar Tomašević. 2023. "The Face of Populism: Examining Differences in Facial Emotional Expressions of Political Leaders Using Machine Learning." arXiv. https://arxiv.org/abs/2304.09914.