Troble-making Monks

SSIM912 - Social Network Analysis

Final Essay

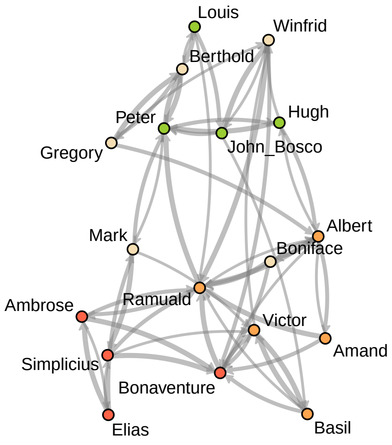
Philipp Holz

02 July 2021

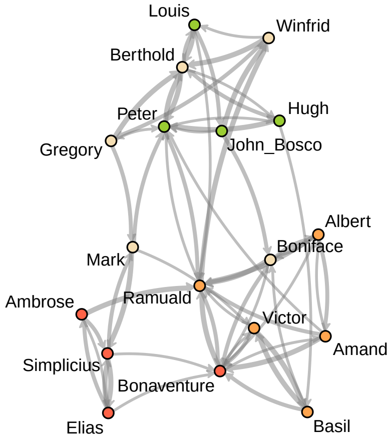
Contents

1 Part 1

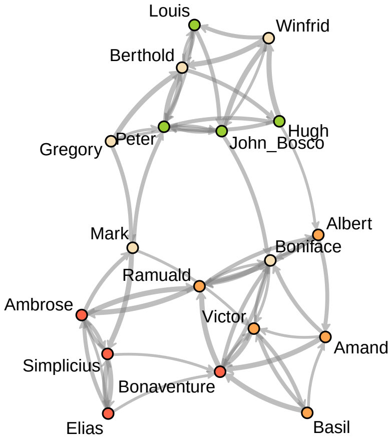
Liking at Time Point 1



Liking Time Point 2



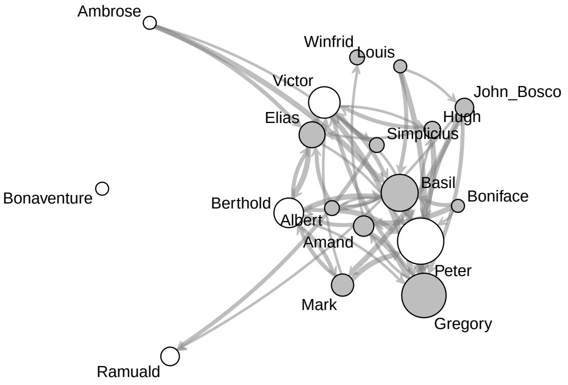
Liking Time Point 3

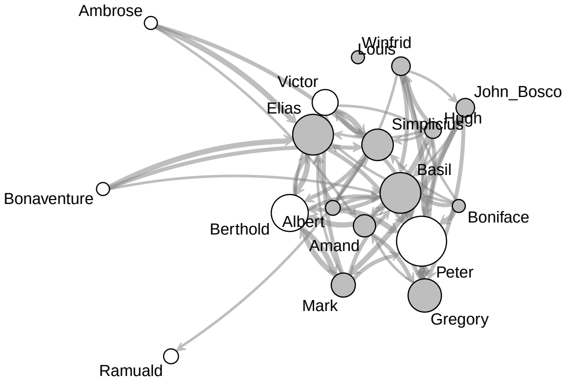


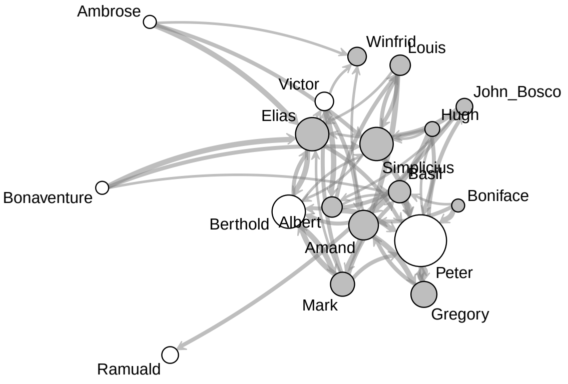
Now, did the groups get more cohesive?

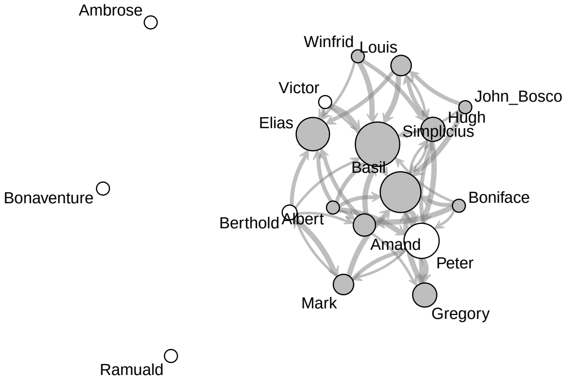
- Graph-level centrality for each of the two subgraphs

2 Part 2


Dislike at Time Point 3


Disesteem at Time Point 3


Negative Influence at Time Point 3


Blame at Time Point 3

Now on to the regression analysis. We try to find out, whether attending the cloisterville seminar reduced the weighted in-degree in the negative sentiment networks. We have to correct for group membership, because it is likely that the two groups extended more negative feelings towards each other than among themselves. Together with the unequal sizes of these groups, this is likely to lead to a bias in more equal sentiment per actor extended towards the smaller of the two groups.

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu % Date and time: Fr, Jul 02, 2021 - 16:31:51

Table 1: Results

	Dependent variable:			
	Dislike,	Disesteem,	Neg. Influence,	Blame
	(1)	(2)	(3)	(4)
Attendance Cloisterville	2.444	1.088	1.121	−3.572
	(2.957)	(3.317)	(2.807)	(2.874)
Group membership	3.333	2.059	2.451	−0.863
	(2.859)	(3.207)	(2.714)	(2.779)
Constant	3.111	5.059^{**}	4.229^{**}	6.026^{***}
	(2.077)	(2.330)	(1.972)	(2.019)
Observations	18	18	18	18
R^2	0.112	0.032	0.058	0.096
Adjusted R^2	−0.006	−0.098	−0.068	−0.025
Residual Std. Error (df = 15)	5.894	6.612	5.595	5.729
F Statistic (df = 2; 15)	0.950	0.244	0.461	0.795
Note:				*p<0.1; **p<0.05; ***p<0.01

3 Code

```r
# Load in the data
sampson.paj <- read.paj("Sampson.paj")

sampson.dat <- read.dl("sampson.dat")

# Extract individual adjacency matrices
SAMPLK1 <- sampson.dat[1:18,1:18,1]
SAMPLK2 <- sampson.dat[1:18,1:18,2]
SAMPLK3 <- sampson.dat[1:18,1:18,3]
SAMPDLK <- sampson.dat[1:18,1:18,4]
SAMPDES <- sampson.dat[1:18,1:18,6]
SAMPNIN <- sampson.dat[1:18,1:18,8]
SAMPNPR <- sampson.dat[1:18,1:18,10]
```

```
# Get the data into an igraph network object
SAMPLK1_igraph <- graph.adjacency(as.matrix(SAMPLK1), mode = "directed", weighted = TRUE
SAMPLK2_igraph <- graph.adjacency(as.matrix(SAMPLK2), mode = "directed", weighted = TRUE
SAMPLK3_igraph <- graph.adjacency(as.matrix(SAMPLK3), mode = "directed", weighted = TRUE
SAMPDLK_igraph <- graph.adjacency(as.matrix(SAMPDLK), mode = "directed", weighted = TRUE
SAMPDES_igraph <- graph.adjacency(as.matrix(SAMPDES), mode = "directed", weighted = TRUE
SAMPNIN_igraph <- graph.adjacency(as.matrix(SAMPNIN), mode = "directed", weighted = TRUE
SAMPNPR_igraph <- graph.adjacency(as.matrix(SAMPNPR), mode = "directed", weighted = TRUE

# Convert to network object
SAMPLK1_net <- asNetwork(SAMPLK1_igraph)
SAMPLK2_net <- asNetwork(SAMPLK2_igraph)
SAMPLK3_net <- asNetwork(SAMPLK3_igraph)
SAMPDLK_net <- asNetwork(SAMPDLK_igraph)
SAMPDES_net <- asNetwork(SAMPDES_igraph)
SAMPNIN_net <- asNetwork(SAMPNIN_igraph)
SAMPNPR_net <- asNetwork(SAMPNPR_igraph)

# Include vertex attributes from description

# attendance to cloisterville
attend_value <- c(1,1,1,1,1,0,1,0,0,0,0,0,0,0,0,0,0,0)

# having been expulsed
expulsed_value <- c(0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1)

# voluntary departure
voluntary_immediate_value <- c(1,0,0,0,0,0,1,0,0,0,0,0,0,1,1,1,0,0)

voluntary_later_value <- c(0,0,0,1,0,0,0,1,0,1,0,1,1,0,0,0,0,0)

# remained
remained_value <- c(0,0,0,0,1,1,0,0,1,0,1,0,0,0,0,0,0,0)


# Get node colors representing attendance to cloisterville
attend_col <- vector()

for (i in 1:length(attend_value)){
  if (attend_value[i] == 1) {
    attend_col[i] <- "white"
  } else {
    attend_col[i] <- "grey"
  }
```

```
}

# Get node colors representing leaving status
leave_col <- vector()

for (i in 1:length(expulsed_value)){
  if (expulsed_value[i] == 1) {
    leave_col[i] <- "tomato"
  } else if (voluntary_immediate_value[i] == 1) {
    leave_col[i] <- "tan1"
  } else if (voluntary_later_value[i] == 1) {
    leave_col[i] <- "wheat"
  } else {leave_col[i] <- "yellowgreen"}
}

# Get a good-looking layout
like_coords <- gplot(SAMPLK3_net)

# Visualise the networks
par(mfrow = c(3,1))

gplot(SAMPLK1_net,
      coord = like_coords,
      usecurve = T,
      edge.curve = 0.05,
      displaylabels = T,
      label.cex = 1,
      arrowhead.cex = 0.5,
      vertex.cex = 0.7,
      vertex.col = leave_col,
      edge.col = alpha("gray50", 0.5),
      edge.lwd = (SAMPLK1_net %e% "weight")*1.5,
      main = "Liking at Time Point 1")

gplot(SAMPLK2_net,
      coord = like_coords,
      usecurve = T,
      edge.curve = 0.05,
      displaylabels = T,
      label.cex = 1,
      arrowhead.cex = 0.5,
      vertex.cex = 0.7,
      vertex.col = leave_col,
      edge.col = alpha("gray50", 0.5),
      edge.lwd = (SAMPLK2_net %e% "weight")*1.5,
      main = "Liking Time Point 2")

gplot(SAMPLK3_net,
      coord = like_coords,
      usecurve = T,
      edge.curve = 0.05,
      displaylabels = T,
      label.cex = 1,
      arrowhead.cex = 0.5,
      vertex.cex = 0.7,
      vertex.col = leave_col,
      edge.col = alpha("gray50", 0.5),
      edge.lwd = (SAMPLK3_net %e% "weight")*1.5,
      main = "Liking Time Point 3")

# Split nodes into these two groups
# 1 - group that remained, 0 - group that left
group_value <- c(0,0,0,1,1,1,0,1,1,1,1,0,0,0,0,0,0,0)

# Get a good-looking layout
dislike_coords <- gplot(SAMPNPR_net)

# Get weighted indegrees
DLK_in <- strength(SAMPDLK_igraph, mode = "in", loops = TRUE, weights = E(SAMPDLK_igraph
DES_in <- strength(SAMPDES_igraph, mode = "in", loops = TRUE, weights = E(SAMPDES_igraph
NIN_in <- strength(SAMPNIN_igraph, mode = "in", loops = TRUE, weights = E(SAMPNIN_igraph
NPR_in <- strength(SAMPNPR_igraph, mode = "in", loops = TRUE, weights = E(SAMPNPR_igraph

# Visualise negative feeling networks
par(mfrow = c(2,2))

gplot(SAMPDLK_net,
      coord = dislike_coords,
      usecurve = T,
      edge.curve = 0.05,
      displaylabels = T,
      label.cex = 1,
      arrowhead.cex = 0.5,
      vertex.cex = DLK_in*0.1+0.7,
      vertex.col = attend_col,
      edge.col = alpha("gray50", 0.5),
      edge.lwd = (SAMPDLK_net %e% "weight")*1.5,
      main = "Dislike at Time Point 3",
      pad = 2)

gplot(SAMPDES_net,
      coord = dislike_coords,
      usecurve = T,
      edge.curve = 0.05,
      displaylabels = T,
      label.cex = 1,
      arrowhead.cex = 0.5,
      vertex.cex = DES_in*0.1+0.7,
      vertex.col = attend_col,
      edge.col = alpha("gray50", 0.5),
      edge.lwd = (SAMPDES_net %e% "weight")*1.5,
      main = "Disesteem at Time Point 3",
      pad = 2)

gplot(SAMPNIN_net,
      coord = dislike_coords,
      usecurve = T,
      edge.curve = 0.05,
      displaylabels = T,
      label.cex = 1,
      arrowhead.cex = 0.5,
      vertex.cex = NIN_in*0.1+0.7,
      vertex.col = attend_col,
      edge.col = alpha("gray50", 0.5),
      edge.lwd = (SAMPNIN_net %e% "weight")*1.5,
      main = "Negative Influence at Time Point 3",
      pad = 2)

gplot(SAMPNPR_net,
      coord = dislike_coords,
      usecurve = T,
      edge.curve = 0.05,
      displaylabels = T,
      label.cex = 1,
      arrowhead.cex = 0.5,
      vertex.cex = NPR_in*0.1+0.7,
      vertex.col = attend_col,
      edge.col = alpha("gray50", 0.5),
      edge.lwd = (SAMPNPR_net %e% "weight")*1.5,
      main = "Blame at Time Point 3",
      pad = 2)
```

```
# Calculate centrality measures
df <- as.data.frame(cbind(attend_value, DLK_in, DES_in, NIN_in, NPR_in))

# Run regression model
lm_DLK <- lm(DLK_in ~ attend_value + group_value, data = df)
lm_DES <- lm(DES_in ~ attend_value + group_value, data = df)
lm_NIN <- lm(NIN_in ~ attend_value + group_value, data = df)
lm_NPR <- lm(NPR_in ~ attend_value + group_value, data = df)

# Then comparing the models

## for printing in the pdf document, change type to "latex"
stargazer(lm_DLK, lm_DES, lm_NIN, lm_NPR,
          type="latex",
          covariate.labels=c("Attendance Cloisterville", "Group membership"),
          dep.var.labels = c("Dislike,", "Disesteem,", "Neg. Influence,", "Blame"),
          title = "Results",
          column.sep.width = "10pt")

# shows coefficients and standard errors, none of the results are significant at the p
```