Assignment 4: Centrality and efficiency

Himanshu Mayank

2023-04-03

Introduction

The aim of this assignment is to use USPTO patent examiner data to create a variable for application processing time, and then use linear regression models to estimate the relationship between centrality and application processing time while controlling for other examiner characteristics. We will also explore whether this relationship differs by examiner gender by including an interaction term in our models. Finally, we will discuss our findings and their implications for the USPTO.

```
##
## Attaching package: 'arrow'
```

```
## The following object is masked from 'package:utils':
##
##     timestamp
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
##
## Attaching package: 'igraph'
```

```
## The following object is masked from 'package:tidyr':
##
##     crossing
```

```
## The following objects are masked from 'package:dplyr':
##
##     as_data_frame, groups, union
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##     union
```

```
# Select only the columns "ego_examiner_id" and "alter_examiner_id" from the "edges" data frame
data_path <- "D:\\MMA Material\\Term 4\\ORBB\\672_project_data\\"
applications <- read_parquet(paste0(data_path,"output.parquet"))
edges <- read_csv(paste0(data_path,"edges_sample.csv"))
```

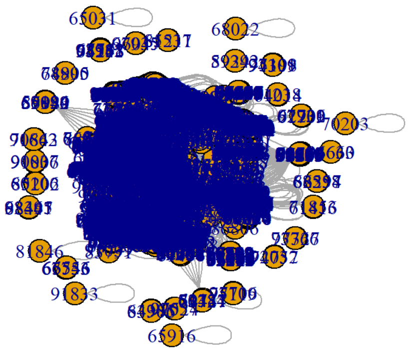
```
## Rows: 32906 Columns: 4
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (1): application_number
## dbl  (2): ego_examiner_id, alter_examiner_id
## date (1): advice_date
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Creation of graph

A graph is created and the different centralities of each node is calculated

```
library(knitr)
edges_subset <- select(edges, ego_examiner_id, alter_examiner_id)
# Remove any rows with null values
edges_subset <- drop_na(edges_subset)
# Create a graph from the edges_subset tibble
g <- graph_from_data_frame(edges_subset, directed = FALSE)
# Set the node names based on the ego_examiner_id and alter_examiner_id columns
node_ids <- unique(c(edges_subset$ego_examiner_id, edges_subset$alter_examiner_id))
V(g)$name <- as.character(node_ids[node_ids %in% V(g)$name])

# Plot the graph
plot(g)
```



```
# Calculate the degree, betweenness centrality, and closeness centrality of each node
degree <- degree(g)
betweenness <- betweenness(g)
closeness <- closeness(g)

# Create a dataframe with the node IDs and centrality measures
result_df <- data.frame(node_id = V(g)$name,
                            degree = degree,
                            betweenness = betweenness,
                            closeness = closeness)

# Print the dataframe
kable(head(result_df,10), caption = "Centrality Measures for Patent Examiner Network")
```

Centrality Measures for Patent Examiner Network

	node_id	degree	betweenness	closeness
84356	84356	34	12616.279	5.61e-05
92953	92953	6	1567.132	5.28e-05
61767	61767	13	26083.551	6.26e-05
72253	72253	42	39173.607	5.61e-05
67078	67078	6	5530.994	5.76e-05
91688	91688	23	3183.135	5.14e-05
61797	61797	84	21161.091	6.21e-05
94270	94270	50	10980.279	6.09e-05
73223	73223	32	31647.835	6.95e-05
60128	60128	9	9944.715	5.77e-05

Data Transformation

Processing time of each application is calculated from filing_date to patent_issue_dat or patent_abandon_date

```
applications$filing_date <- as.Date(applications$filing_date)
#applications$application_result_date <- as.Date(applications$application_result_date, format = "%Y-%m-%d")
applications$application_result_date <- ifelse(!is.na(applications$patent_issue_date),
                                               as.Date(applications$patent_issue_date),
                                               as.Date(applications$abandon_date))
applications$application_result_date <- as.Date(applications$application_result_date,
                                                format = "%Y-%m-%d", origin = "1970-01-01")
applications$application_processing_time <- as.integer(difftime
                                                       (applications$application_result_date,
                                                         applications$filing_date, units = "days"))
```

```
# Convert node_id column in result_df to double
result_df$node_id <- as.numeric(result_df$node_id)

process_data <- select(applications, examiner_id, examiner_art_unit, tc, race, tenure_days, gender, application_processing_t
ime)
process_data$examiner_id <- as.numeric(process_data$examiner_id)

# Perform left join
# join process_data and result_df by examiner_id and node_id, respectively
# remove rows with NaN values in result_df
result_df <- result_df[complete.cases(result_df),]
# remove rows with NaN values in process_data
process_data <- process_data[complete.cases(process_data),]
process_data$tc = as.character(process_data$tc)
joined_data <- left_join(process_data, result_df, by = c("examiner_id" = "node_id"))
# remove rows with NAs
joined_data <- na.omit(joined_data)
```

```
joined_data$tc <- factor(joined_data$tc)
joined_data$race <- factor(joined_data$race)
joined_data$gender <- factor(joined_data$gender)
joined_data[, c("tenure_days", "degree", "betweenness", "closeness")] <- scale(joined_data[, c("tenure_days", "degree", "bet
weenness", "closeness")])
kable(head(joined_data,10), caption = "Final table")
```

Final table

examiner_id	examiner_art_unit	tc	race	tenure_days	gender	application_processing_time	degree	betweenness	closeness
63213	1752	1700	white	0.6296442	female	-1170	-0.4888784	-0.4512543	-0.1017199
73788	1648	1600	white	0.6107444	female	1481	-0.4579314	-0.4641175	-0.1019633
77294	1762	1700	white	0.6121983	male	261	0.6252166	0.1292200	-0.1018691
77112	1755	1700	white	0.6340057	female	644	0.1919574	1.8594665	-0.1017974
92931	1642	1600	white	0.6383672	female	294	-0.5198255	-0.4697439	-0.1019925
75406	1733	1700	white	0.6281904	male	693	0.7799520	0.9623240	-0.1018076
63176	1722	1700	white	0.6165598	male	1048	2.2963591	3.8243426	-0.1017815
59816	1751	1700	white	0.6340057	male	2387	-0.4579314	-0.2529363	-0.1017278
64507	1644	1600	white	0.6063829	male	1210	-0.3960372	-0.3435523	-0.1018607
82563	1714	1700	white	0.6340057	male	1946	-0.4579314	-0.4208963	-0.1018572

```
library(dplyr)
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
library(fastDummies)
```

```
## Warning: package 'fastDummies' was built under R version 4.2.3
```

```
encoded_data <- joined_data %>%
  dummy_cols(select_columns = c("gender", "race", "tc"))
kable(head(encoded_data,10), caption = " Transformed application data")
```

Transformed application data

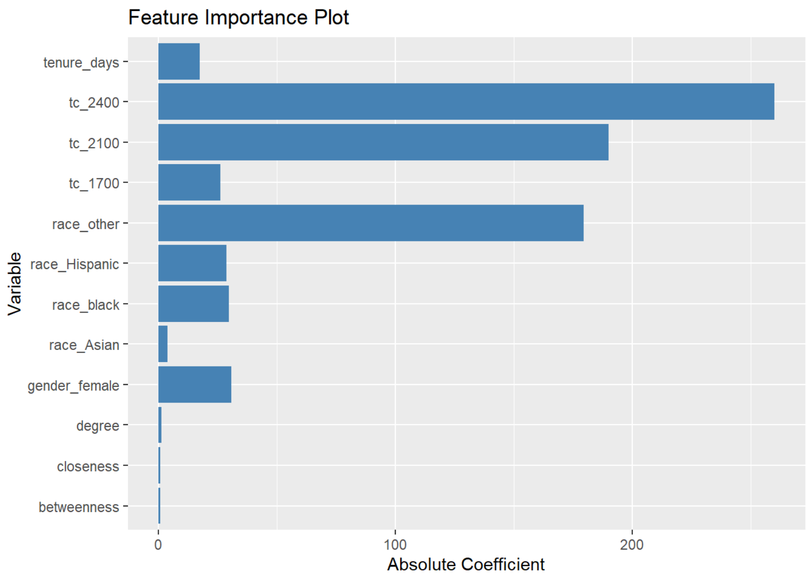
examiner_id	examiner_art_unit	tc	race	tenure_days	gender	application_processing_time	degree	betweenness	closeness	gender_female	ge
63213	1752	1700	white	0.6296442	female	-1170	-0.4888784	-0.4512543	-0.1017199	1	
73788	1648	1600	white	0.6107444	female	1481	-0.4579314	-0.4641175	-0.1019633	1	
77294	1762	1700	white	0.6121983	male	261	0.6252166	0.1292200	-0.1018691	0	
77112	1755	1700	white	0.6340057	female	644	0.1919574	1.8594665	-0.1017974	1	
92931	1642	1600	white	0.6383672	female	294	-0.5198255	-0.4697439	-0.1019925	1	
75406	1733	1700	white	0.6281904	male	693	0.7799520	0.9623240	-0.1018076	0	
63176	1722	1700	white	0.6165598	male	1048	2.2963591	3.8243426	-0.1017815	0	
59816	1751	1700	white	0.6340057	male	2387	-0.4579314	-0.2529363	-0.1017278	0	
64507	1644	1600	white	0.6063829	male	1210	-0.3960372	-0.3435523	-0.1018607	0	
82563	1714	1700	white	0.6340057	male	1946	-0.4579314	-0.4208963	-0.1018572	0	

Creating linear regression models

```
model <- lm(application_processing_time ~ tc_1700 + tc_2100 + tc_2400 + race_Asian + race_black + race_Hispanic + race_other
+ tenure_days + gender_female + degree + betweenness + closeness, data = encoded_data)
summary(model)
```

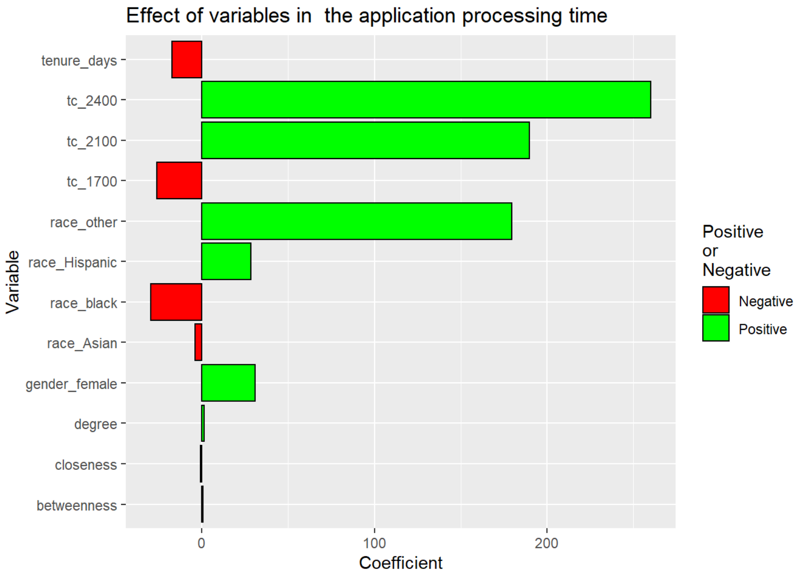
```
##
## Call:
## lm(formula = application_processing_time ~ tc_1700 + tc_2100 +
##     tc_2400 + race_Asian + race_black + race_Hispanic + race_other +
##     tenure_days + gender_female + degree + betweenness + closeness,
##     data = encoded_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7416.1  -417.7  -105.2   289.8  4965.8
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1134.9993     1.5881 714.687  < 2e-16 ***
## tc_1700        -26.0716     1.7525 -14.877  < 2e-16 ***
## tc_2100        189.9264     2.0455  92.849  < 2e-16 ***
## tc_2400        259.9813     2.3063 112.729  < 2e-16 ***
## race_Asian      -3.7885     1.5380  -2.463   0.0138 *
## race_black     -29.6865     3.6603  -8.110 5.05e-16 ***
## race_Hispanic   28.5915     4.5483   6.286 3.25e-10 ***
## race_other     179.5359    19.1086   9.396  < 2e-16 ***
## tenure_days    -17.2730     0.6702 -25.774  < 2e-16 ***
## gender_female   30.7230     1.4573  21.083  < 2e-16 ***
## degree           1.1913     0.8485   1.404   0.1603
## betweenness      0.6542     0.8478   0.772   0.4403
## closeness       -0.7918     0.6533  -1.212   0.2255
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 616.1 on 902838 degrees of freedom
## Multiple R-squared:  0.03574,    Adjusted R-squared:  0.03573
## F-statistic:  2789 on 12 and 902838 DF,  p-value: < 2.2e-16
```

```
library(ggplot2)
# Create a data frame of coefficients and their corresponding variables
coef_df <- data.frame(variable = names(model$coefficients)[-1],
                      coefficient = abs(model$coefficients[-1]))
# Create a bar plot of the absolute values of the coefficients
ggplot(coef_df, aes(x = variable, y = coefficient)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  coord_flip() +
  xlab("Variable") +
  ylab("Absolute Coefficient") +
  ggtitle("Feature Importance Plot")
```



```
# Extract the coefficients and their corresponding variables
coef_df <- data.frame(variable = names(model$coefficients)[-1],
                      coefficient = model$coefficients[-1])

# Create a bar chart showing the positive and negative impact of variables
library(ggplot2)
ggplot(coef_df, aes(x = variable, y = coefficient, fill = coefficient > 0)) +
  geom_bar(stat = "identity", color = "black") +
  scale_fill_manual(values = c("red", "green"), labels = c("Negative", "Positive")) +
  coord_flip() +
  labs(x = "Variable", y = "Coefficient", fill = "Positive\nor\nNegative") +
  ggtitle("Effect of variables in  the application processing time")
```



- Tenure days have negative impact on processing time. Higher the tenure date lower is the processing time.
- The various centrality measures are not important variables when predicting the application processing times
- The application processing time is higher for females compared to males
- The processing times for tc_2400 and tc_2100 are higher while that of tc_1700 is lower
- The processing time for the asian and black race is lower compared to hispanic and other race

Capturing the interaction between gender and various centality measures

Multiplying two columns together can create a new feature that captures an interaction effect between the two original features. If the new feature (i.e., the product of the two columns) is significant in a model while one of the original features is not significant, it could mean that the interaction effect captured by the new feature is important in predicting the outcome variable.

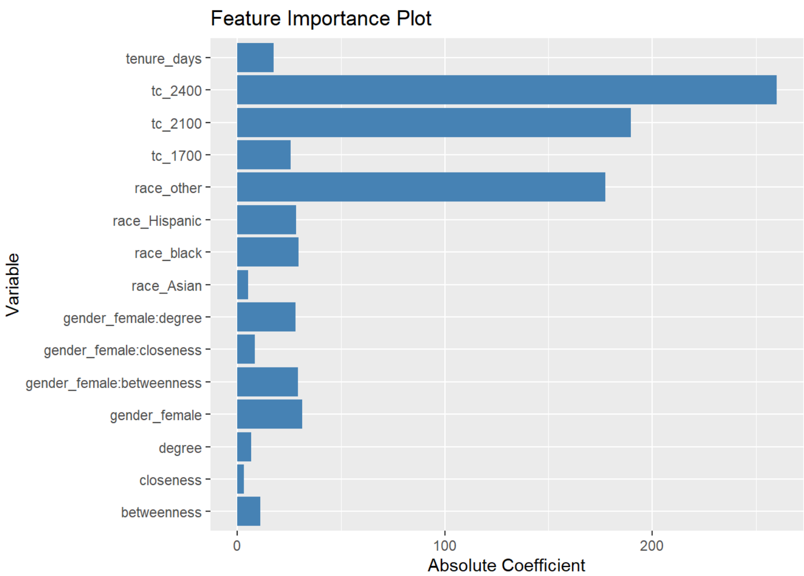
In other words, the interaction between the two variables might have a stronger relationship with the outcome variable than either of the variables on their own. So while one of the variables on its own might not be significant, its interaction with the other variable might be important for predicting the outcome.

```
model <- lm(application_processing_time ~ tc_1700 + tc_2100 + tc_2400 + race_Asian + race_black + race_Hispanic + race_other
+ tenure_days + gender_female + degree + betweenness + closeness + degree*gender_female + betweenness*gender_female + closen
ess *gender_female  , data = encoded_data)
summary(model)
```

```
##
## Call:
## lm(formula = application_processing_time ~ tc_1700 + tc_2100 +
##     tc_2400 + race_Asian + race_black + race_Hispanic + race_other +
##     tenure_days + gender_female + degree + betweenness + closeness +
##     degree * gender_female + betweenness * gender_female + closeness *
##     gender_female, data = encoded_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7413.9  -417.6  -105.2   289.9  4967.3
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 1135.3774     1.5891 714.475  < 2e-16 ***
## tc_1700                      -25.6445     1.7534 -14.625  < 2e-16 ***
## tc_2100                      189.7510     2.0474  92.680  < 2e-16 ***
## tc_2400                      259.7838     2.3088 112.517  < 2e-16 ***
## race_Asian                    -5.2572     1.5412  -3.411 0.000647 ***
## race_black                   -29.6127     3.6596  -8.092 5.89e-16 ***
## race_Hispanic                 28.4326     4.5485   6.251 4.08e-10 ***
## race_other                   177.4031    19.1056   9.285  < 2e-16 ***
## tenure_days                  -17.6787     0.6710 -26.347  < 2e-16 ***
## gender_female                 31.3512     1.4574  21.511  < 2e-16 ***
## degree                        -6.6676     0.9793  -6.808 9.87e-12 ***
## betweenness                   11.1030     1.0675  10.401  < 2e-16 ***
## closeness                     -3.0901     0.7720  -4.003 6.25e-05 ***
## gender_female:degree          28.0770     1.9937  14.083  < 2e-16 ***
## gender_female:betweenness    -29.4249     1.7580 -16.738  < 2e-16 ***
## gender_female:closeness        8.5594     1.4401   5.944 2.79e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 616 on 902835 degrees of freedom
## Multiple R-squared:  0.0361, Adjusted R-squared:  0.03608
## F-statistic:  2254 on 15 and 902835 DF,  p-value: < 2.2e-16
```

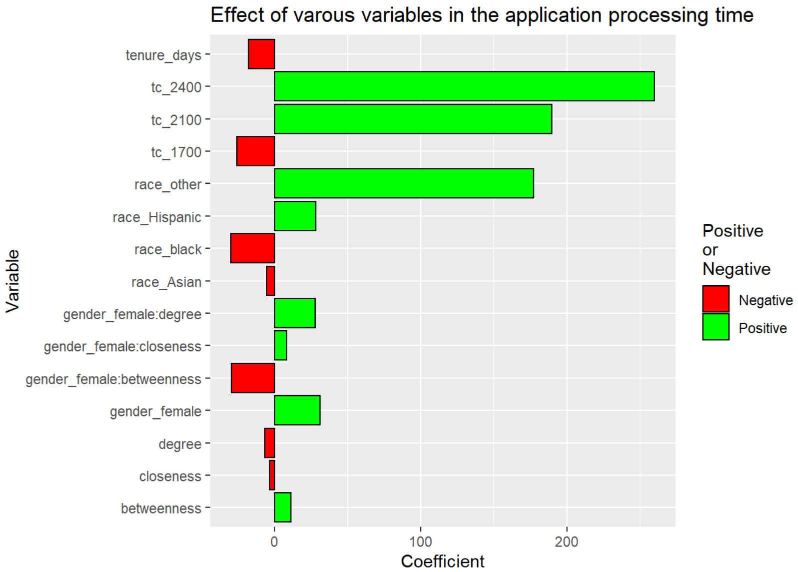
```
# Create a data frame of coefficients and their corresponding variables
coef_df <- data.frame(variable = names(model$coefficients)[-1],
                      coefficient = abs(model$coefficients[-1]))

# Create a bar plot of the absolute values of the coefficients
ggplot(coef_df, aes(x = variable, y = coefficient)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  coord_flip() +
  xlab("Variable") +
  ylab("Absolute Coefficient") +
  ggtitle("Feature Importance Plot")
```




```
# Extract the coefficients and their corresponding variables
coef_df <- data.frame(variable = names(model$coefficients)[-1],
                      coefficient = model$coefficients[-1])

# Create a bar chart showing the positive and negative impact of variables
ggplot(coef_df, aes(x = variable, y = coefficient, fill = coefficient > 0)) +
  geom_bar(stat = "identity", color = "black") +
  scale_fill_manual(values = c("red", "green"), labels = c("Negative", "Positive")) +
  coord_flip() +
  labs(x = "Variable", y = "Coefficient", fill = "Positive\nor\nNegative") +
  ggtitle("Effect of varous variables in the application processing time")
```



Insights and Interpretations

- The tc_2400 and tc_2100 have a positive effect on the application processing time while tc_1700 has negative effect. The tc_1700 processes the applications All these variables are significant in predicting the application processing time.
- Looking at USPTO overall, hispanic and other race have a positive effect on application processing time while for asian and black it has a negative effect implying that asians and black take less time in processing the applications
- The gender variable (specifically when it is female) has a positive effect on application processing time, meaning that women process applications faster than men on average.
- The Betweenness centrality variable (a measure of how important a node is in a network) has a positive effect on application processing time when considered alone. However, when looking at women with high betweenness centrality, the effect on application processing time is inversely proportional, meaning that the processing time actually increases for these women.
- Women with high betweeness centrality,ie, the women who lie in the critical path of information flow generally have a lower processing time. People in general with high betweenness centrality on the other hand take longer time to process the applications. A possible interpretation is that generally these women have high expertise leading to lower processing time for applications. Their expertise could be the reason why other people always consult them or consult other people through these women.
- For women with high degree centrality and closeness centrality the processing time increases. Women who are well connected in the network and have a large amount of information flowing through them generally take more to process the applications. It could be due to the fact that the applications that they are processing might require additional consultations with other reviewers leading to a longer processing time. Women with high closeness centrality can quickly communicate with other people in the network and as a result are consulted more by other people and thus their application processing time takes longer