Social Network Analysis - Assignment 2

Group 7

Floris Vermeulen Martijn van Iterson Niek Fleerakkers Patryk Grodek
Samir Sabitli

2025-11-16

Table of contents

1 Introduction

With the advent of online peer-to-peer (P2P) lending platforms, the traditional methods of financial intermediation have been usurped by individual choice. The surging popularity and accessibility of services such as Prosper.com has grown to become one of the most important avenues through which individuals can secure micro-loans (Cai, Lin, Xu, & Fu, 2016). Arguably, while the democratisation of credit alleviates numerous pre-existing issues, it serves to exaggerate well-known informational asymmetries associated with determining creditworthiness (Mingfeng Lin, N.R. Prabhala, & Siva Viswanathan, 2009). Per, Chen, Zhou, & Wan (2016), from a traditional financial lens, these asymmetries can contribute to moral hazard and adverse selection, ultimately causing systemic financial losses. This arises from the fact the users are anonymous, challenging the veracity of their information. Further, the unsecured nature of credit makes its collection relatively ineffective. Jointly, these factors characterise the lemons market theory by Akerlof (1978).

However, social networks can act a crucial avenue through which information transfer could be facilitated, alleviate some of the existing asymmetries (Mingfeng Lin et al., 2009). There are several prominent theories and research that attempt to bridge this gap. First, the theory of social capital is particularly salient because it relates directly to the reputation and trustworthiness of the users (Adler & Kwon, 2002). While its definition is not strictly defined, according to Putnam (2015), it can be seen as a means through which social organisation facilitate coordination and trust. Nahapiet & Ghoshal (1998) define three dimensions, including structural, relational, and cognitive. We primarily study the structural component, whereby the network properties relating to tie formation between borrowers is considered. Following this perspective, Chen et al. (2016) argue that social capital is earned through connections to others. However, per the authors, the internet facilitates weak forms of these ties, diminishing the benefits of social capital on P2P platforms, especially for non-friendship networks. Notably, they cite a low interdependence and closed structures to this effect.

With this, we arrive at our first research problem, aiming to understand **to what extent are borrower networks weak in P2P networks?** More specifically, we formulate the hypothesis

$\mathbf{H}_1^a$: Borrower networks based on similarity have a low degree of centralisation

To understand social capital, we employ techniques related to the occurance of dyadic connections rather than studying their structural components. As such, this problem can be studied with the use of the Quadratic Assignment Procedure (QAP) and Conditional Uniform Graph (CUG) tests, whereby the theoretical distributions of network characteristics can be tested.

Our second study attempts to model social capital as a statistical model, specifically considering the structural dimension to social capital. According to the comprehensive review of Bachmann et al. (2011), past studies have considered financial, demographic, and soft informational factors in the financial outcomes of P2P lending, however, these perspectives have not been considered with relation to each other. With respect to this premise, we posit **what is the role of structural and non-structural factors in the formation of social capital?** This problem is arguably best studied with the use of Exponential Random Graph Models (ERGM) as they are considered to work well in identifying a combination of structural and exogenous patterns in the context of a variety of network specifications (Jackson, 2011).

In studying demographic factors, Pope & Sydnor (2008) suggest that age plays a significant role in funding success. In particular, compared to individuals aged 35-60, those aged 35 or younger have a

40-90 basis point higher success rate. Additionally, those aged 60 and above are 1.1-2.3 percentage points less successful. While this is not a direct relationship with social capital, it may suggest that these individuals may be placed lower in the P2P market, per Putnam (2015).

In studying gender, Barasinska & Schaefer (2010) find that female lenders are less risk averse than male lenders, funding credit with lower interest rates at a higher frequency. In conjunction, Pope & Sydnor (2008) find that single women pay 0.4% less interest than men. These factors may indicate that edges are more likely to form between borrowers if they are women.

The structural dynamics of the network could be explained by the theory of Relational Herding, as formulated by De Liu, Brass, Lu, & Chen (2015). The authors explain that in the face of uncertainty, actors tend to exhibit a clustering behaviour, even putting aside their own private information. Though, other social dynamics may also explain the phenomenon. Devenow & Welch (1996) posit that herding may also occur if individuals blindly follow others without rational analysis. In the P2P context, Herzenstein, Dholakia, & Andrews (2011) show evidence of strategic herding when borrowers observe that a loan has gotten funding in the past.

Further, Podolny (1993) explains that the ways in which information flows in social networks can explain its outcomes. In particular, if the reputation or status of an individual can be observed, they are said to be "pipes" for other actors to transact with them. However, if this reputation is merely *perceived*, then it said to be a "prism". In our context, prism seems more appropriate as users usually cannot directly observe prestige-related characteristics, however, this can be tested by constructing a variable related to latent status.

Hypotheses	**Type**	**Dyad**	**ERGM Term**	**Motivation**
$\mathbf{H}_2^a$: Younger borrowers tend to form more relationships with each other	Exogenous	Independent	nodecov()	The term captures how the age covariate affects the likelihood of edge formation. Since we want a numeric range, this term is most appropriate.
$\mathbf{H}_3^a$: Women tend to form more relationships with other borrowers, regardless of gender	Exogenous	Independent	nodefactor()	The term captures the tendency of women to form more edges, regardless of the other gender.
$\mathbf{H}_4^a$: Borrowers tend to exhibit herding around each other	Endogenous	Interdependent	gwesp()	The term captures the tendency for triadic relationships to form. In other words, for ties to form due to the presence of other ties in the dyad's neighbourhood.
$\mathbf{H}_5^a$: Borrowers tend to form ties around other high-status borrowers	Endogenous	Interdependent	gwd() & nodecov()	The term describes degree distribution, indicating whether a borrower is popular to the degree specified. It does not necessarily relate to community. It is necessary to also use an indicator of prestige to understand whether pipe or prism dynamics may be more prevalent.

Our research makes notable contributions to existing research on social network analyses on P2P networks. The study primarily expands upon the existing ideas of social capital by analysing it with respect to European P2P lending market and evaluating the interactions between structural and non-structural factors such as demographics. Typically, research tends to consider them independently and utilises predictive, rather than purely statistical models.

Following this, the report will outline the methodology employed. In this section, we will discuss the dataset utilised, specific data processing steps and other considerations relating to the empirical or network environment. Further, we justify the empirical structure of the analysis, evaluating the specific methodologies utilised with respect to the wider quantitative landscape. Subsequently, the results are outlined for each model and hypothesis, alongside their interpretations for our research problem. Finally, we conclude the report by summarising the core research problem, our empirical set-up alongside

considerations for future research.

2 Methodology

Within this section, we construct the empirical layout of our research problem, including the dataset and data processing steps.

2.1 Dataset

The study utilises a publicly-retrieved dataset from a leading European P2P platform called Bondora. The dataset contains detailed information on both defaulted and non-defaulted loans given to users between February 2009 and July 2021. Prior to any manipulation, the dataset contains a range of numeric, binary, categorical, and time-series attributes across 85,087 unique users and 179,235 individual loans. The data was originally collected by Siddhartha (n.d.) in 2021 and published on kaggle.com. While the user published the data, they simply downloaded it from a now-defunct page on Bondora's website. Public data is no longer offered by Bondora, access to newer data is not possible.

2.2 Data Processing and Network Construction

To make the data usable for our research, it was processed. The full steps can be seen in the Appendix. First, only attributes relevant to the research were kept for resource efficiency and ease of use. Following this, any rows with missing values were entirely removed to preserve a complete dataset. This step did not remove a significant number of data points. Following this, it is mandatory to reduce the size of the overall dataset to ensure that any analyses conducted can converge and do so in a timely manner. We did this by randomly removing data points until the dataset had 500 remaining observations. While a sampling bias is technically possible as a result, the pseudo-random data reduction should minimize this effect.

To construct the network, we need to specify edges between users. While there are many ways to do this, we prefer an approach that does not introduce unnecessary complexity to models' interpretations and maintains a meaningful semantic relationship between borrowers. Ultimately, we create edges based on how similar borrowers are across several dimensions, computed by cosine similarity. The intuition here arises from the fact that certain behaviours or social processes *groups* borrowers together. This technique has the advantage of discounting vector size for its angle, which reduces the influence of outliers in defining how similar individuals are. However, to avoid a fully connected network that would make ERGM completely redundant, we establish a threshold of `cosine_sim=0.5` to determine whether an edge can exist. This enables us to control how similar individuals must be for our analysis while maintaining a reasonable density.

Defining the dimensions that make an individual similar to another is a difficult task, however, we opted for attributes describing users' loans rather than their demographic or financial characteristics. This is *crucial* because we must isolate these variables from any outcome being studied. Otherwise, our analysis will be severely biased by data leakage and possibly even simultaneous equation bias. Further, we chose to standardise the attributes chosen for cosine similarity to avoid bias introduced by differing variable scales. Ultimately, the edges formed are weighted [1] by how similar individuals are to each

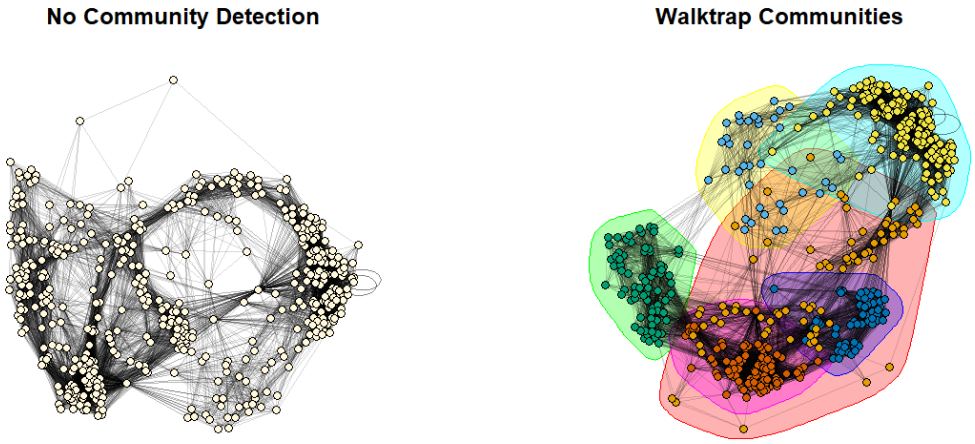
[1] We understand that weighted edges make ERGMs significantly more difficult. We can definitely make them binary for further analysis. We believe that making them weighted can make the analysis more meaningful because it would allow

other. The attributes excluded from this similarity measure were later added to the `igraph` network object as vertex attributes.

2.3 Descriptive Statistics and Network Overview

Having processed the dataset and created a weighted network of similar borrowers, we provide an overview of the descriptive statistics relating to the network and wider dataset. The figure below shows an overview of the network, however, since we rely on sampling for the nodes, the network is not equivalent to the underlying population. Observing the figure, we see that there are indeed prevalent communities, however, there is also significant overlap between them.

Figure 1: Plot of the Bondera Network with and without Community Detection



[2]

Observing the figure below, we see low betweenness centrality and degree distribution without any logarithmic adjustments. The latter indicates that most nodes have very low connectivity. This may have been a result of the threshold set for the similarity. The former suggests that few nodes act as the bridge between different communities of borrowers and that the few highly connected nodes are crucial for the connection of the wider P2P network.

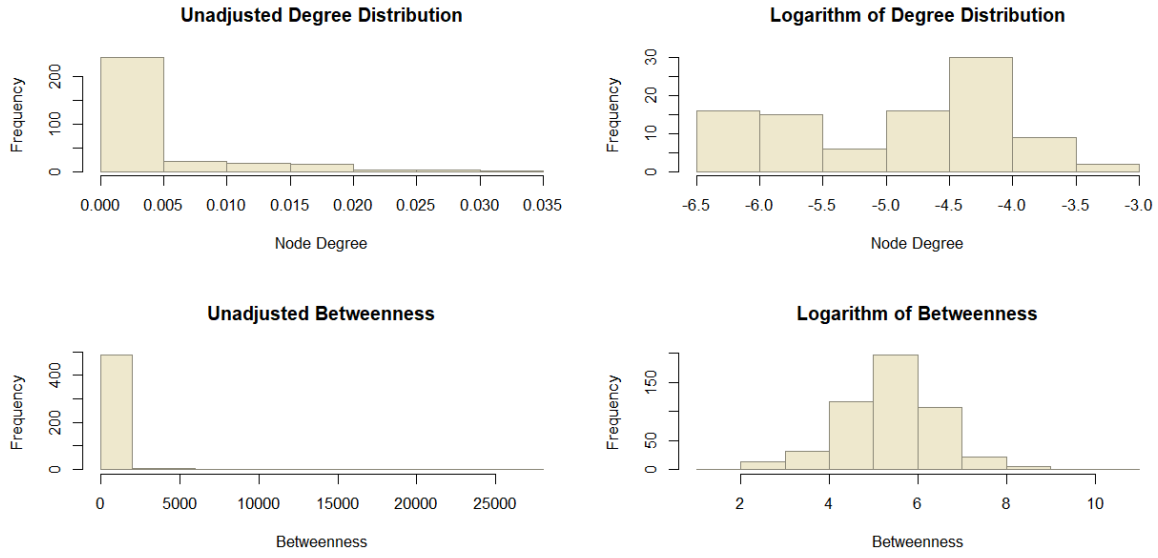
us to determine the degree to which a predictor makes someone similar which can be important in determining something like the relational herding or prism/pipe effects. To be discussed.

[2]We recognise that this reciprocity may not beideal and we are looking for ways to deal with the 100% reciprocity. Potential point for discussion.

Table 1: Network Descriptive Statistics

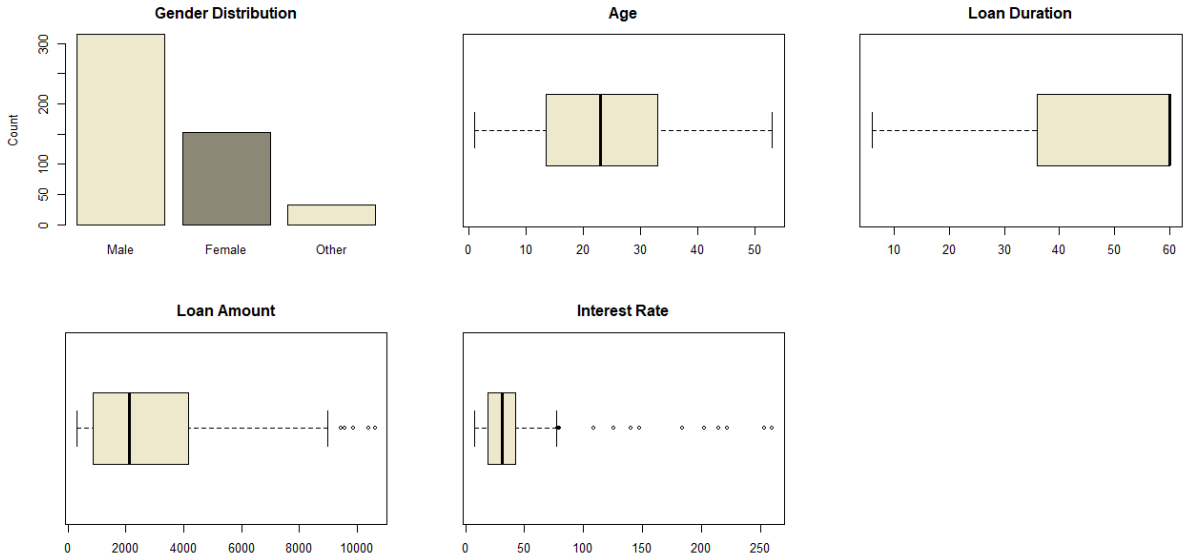
Network Measure	**Value**
Number of Vertices	495
Number of Edges	14,017
Density	0.109
Reciprocity	1.000
Transitivity	0.692
Mean Distance	2.112
Dyad Census: Mutual	13,269
Dyad Census: Null	108,996

Figure 2: Degree and Betweeness Distribution of the P2P Network



The figure below shows various aspects of the wider dataset. Firstly, we observe that there are significantly more males within our subsample borrowing than females. Notably, some users did note "Other" for their gender. It is unclear whether this relates to the user being non-binary or whether it was a data error. Generally, borrowers tend to be young, with the median hovering around 22. The interquartile range is approximately 10 years, suggesting that most users lean younger. The duration of users' loans tends to be extremely left skewed, indicating the the majority of users request very long loans. There is quite a large range in terms of the loan amounts and their interest rates. The median loan amount hovers around 2000 euros, with most of the users not borrowing more than 4000. This distribution reaffirms that users prefer loans on the lower end. For most users, the interest rate is tightly dispersed and below 40%. This could suggest that loan providers are quite selective in their clientèle. Notably, some users do exceed 100% interest rates, suggesting that the data has users with extraordinary circumstances.

Figure 3: Descriptive Statistics of the wider Dataset



A Supplements

A.1 Data Preprocessing

A.2 Source Code - Data Preprocessing

```r
#-------------------------------------------------------------------------#
#install.packages("viridis") # For Colours
#-------------------------------------------------------------------------#
# Import the Bondora P2P Dataset
obj_paths = "resources/objects/"
bondora_raw <- read.csv("dataset/LoanData_Bondora.csv")
raw_cols <- colnames(bondora_raw)
#-------------------------------------------------------------------------#
# Select Columns to Keep
keep_cols <- c("LoanId", "UserName","Age", "Gender",
               "Country", "Amount", "Interest","LoanDuration",
              "UseOfLoan", "Education", "MaritalStatus",
              "Rating", "Restructured", "MonthlyPayment")

bondora <- bondora_raw[keep_cols]

# Remove Rows with any NAs -> Complete Dataset Preferred
print(paste("NA Count |",sum(is.na(bondora)),"rows"))
bondora <- na.omit(bondora)
print(paste("NA Count |",sum(is.na(bondora)),"rows"))

# Remove users with only -1 Stated UseofLoan
bondora_clean <- bondora[bondora$UseOfLoan != -1, ]

# Remove Users with only One Loan
user_counts <- table(bondora_clean$UserName)
multi_users <- names(user_counts[user_counts > 5])
bondora_clean <- bondora_clean[bondora_clean$UserName %in% multi_users, ]

# See if Ratings are Properly Encoded
unique(bondora_clean$Rating)

# See distribution of UserName Counts
hist(table(bondora_clean$UserName))
barplot(table(bondora_clean$UseOfLoan))

# Extract UseofLoan Types and Turn into Factor
bondora_clean$UseOfLoan_factor <- as.factor(bondora_clean$UseOfLoan)
unique(bondora_clean$UseOfLoan_factor)
levels(bondora_clean$UseOfLoan_factor) <- c(
  "Loan Consolidation", "Real Estate","Home Improvement",
  "Business","Education","Travel","Vehicle","Other","Health")

bondora$UseOfLoan_factor <- as.factor(bondora$UseOfLoan)
unique(bondora$UseOfLoan_factor)
levels(bondora$UseOfLoan_factor) <- c(
  "Unknown", "Loan Consolidation", "Real Estate","Home Improvement",
  "Business","Education","Travel","Vehicle","Other","Health",
  "Machinery Purchase", "Acquisition Real Estate","Other Business"
)
#-------------------------------------------------------------------------#
# Observe Descriptive Statistics

cols <- viridis::viridis(30)

# Make function to save plots
save_plot <- function(plt_nam) {
  plt <- recordPlot()
  saveRDS(plt, here::here("resources","objects","preprocessing",
                          paste0(plt_nam,".Rds")))
}

# Make function to consistently plot comparisons
plot_desc_hists <- function(df1, df2, col_name, type) {

```

```
  par(mfrow=c(1,3))

  hist(df1[[col_name]], xlab=type, col=cols[15], main="", breaks=10)
  mtext("Full Sample", side=3, adj=0, line=0.25, cex=1, font=2)

  hist(df2[[col_name]], xlab=type, col=cols[15], main="", breaks=10)
  mtext("Processed Sample", side=3, adj=0, line=0.25, cex=1, font=2)

  qqplot(df1[[col_name]], df2[[col_name]], main="", cex=1,
         xlab="Full Sample", ylab="Subsample", line=0.25)
  abline(0, 1, lty=2)

  mtext("QQ Plot", side=3, adj=0, line=0.25, cex=1, font=2)

  mtext(type, outer = TRUE, line = -2, side=3, cex = 1.3, font = 2)

  # Reset plot window
  par(mfrow=c(1,1), mar=c(5,4,4,2)+0.1)
}

plot_desc_bar <- function(df1, df2, col_name, type) {

  par(mfrow=c(1,2), mar=c(5,10,4,2))

  barplot(sort(table(df1[[col_name]]), decreasing = F),
          xlab=type, col=cols[15], horiz=TRUE, las=1)
  mtext("Full Sample", side=3, adj=0, line=0.25, cex=1, font=2)

  barplot(sort(table(df2[[col_name]]), decreasing = F),
          xlab=type, col=cols[15], horiz=TRUE, las=1)
  mtext("Processed Sample", side=3, adj=0, line=0.25, cex=1, font=2)

  mtext(type, outer = TRUE, line = -2, side=3, cex = 1.3, font = 2)

  # Reset plot window
  par(mfrow=c(1,1), mar=c(5,4,4,2)+0.1)
}

plot_desc_hists(bondora, bondora_clean, "Amount", "Amount")
save_plot("hist_amt")

plot_desc_hists(bondora, bondora_clean, "Interest", "Interest")
save_plot("hist_int")

plot_desc_hists(bondora, bondora_clean, "LoanDuration", "Loan Duration")
save_plot("hist_loandur")

plot_desc_hists(bondora, bondora_clean, "MonthlyPayment", "Monthly Payment")
save_plot("hist_monpmt")

plot_desc_hists(bondora, bondora_clean, "Age", "Age")
save_plot("hist_age")

plot_desc_bar(bondora, bondora_clean,"UseOfLoan_factor","Loan Purpose")
save_plot("bar_loanuse")

plot_desc_bar(bondora, bondora_clean,"Rating","Credit Rating")
save_plot("bar_rating")
#------------------------------------------------------------------#
# Convert Dataset into Incidence Matrix to form Network Object (for ERGM)
bondora_slim <- bondora_clean

# Create the Incidence Matrix for Use of Loan
bondora_matrix <- table(
  bondora_slim$UserName, bondora_slim$UseOfLoan)
bondora_matrix[bondora_matrix > 0] <- 1 # Given that ergm.counts fails with GOF
```

```r

# Create network object with counts as Edge attribute
bondora_net <- network::network(
  bondora_matrix, directed=FALSE, bipartite=nrow(bondora_matrix),
  ignore.eval = FALSE, names.eval="frequency", loops=FALSE)

# Set the bipartite Attribute - UNNCESSARY GIVEN bipartite=length(n)
len <- dim(bondora_matrix)[1]
len_b2 <- dim(bondora_matrix)[2]
b_indicator <- c(rep(1,len),rep(2,len_b2))
#network::set.vertex.attribute(
#  bondora_net, "bipartite", value = rep(len,len), v=1:len
#)

# Extract Partition 2 Labels
loan_use <- levels(bondora_clean$UseOfLoan_factor)

# Create Loan Type Attribute for Partition 2
b2_loantype <- rep(NA, len)
b2_loantype <- c(b2_loantype, loan_use)

if (length(b2_loantype) == network::network.size(bondora_net)) {
  network::set.vertex.attribute(
    bondora_net, "b2_loantype", value = b2_loantype)
  }

# Add Age Vertex Attribute to B1
age <- bondora_clean$Age[match(
  rownames(bondora_matrix), bondora_clean$UserName)]
b1_age <- c(age, rep(NA, len_b2))
network::set.vertex.attribute(
  bondora_net, "b1_age", value=b1_age
)

# Add Gender Vertex Attribute to B1
gender <- bondora_clean$Gender[match(
  rownames(bondora_matrix), bondora_clean$UserName)]
unique(gender) # Check to see if encoded properly
gender <- ifelse(gender == 0, "male","female")
b1_gender <- c(gender, rep(NA, len_b2))
network::set.vertex.attribute(
  bondora_net, "b1_gender", value = b1_gender
)

# Save the network object
saveRDS(bondora_net, file=paste0(obj_paths,"preprocessing/","bondora_net.Rds"))
#------------------------------------------------------------------------#
# Get the Adjacency Matrix for Loan Use Similarity (Dependent QAP Variable)
adj_mat_loan_use <- bondora_matrix %*% t(bondora_matrix)
# Remove self weights
diag(adj_mat_loan_use) <- 0

# Get the Adjacency Matrix for Credit Rating Similarity (Predictor in QAP)
incidence_rating <- table(bondora_slim$UserName, bondora_slim$Rating)
adj_mat_rating <- incidence_rating %*% t(incidence_rating)
diag(adj_mat_rating) <- 0

# Get the Adjacency Matrix for Loan Amount (Control in QAP) - BINARY
# First, bin the Loan Amounts
bondora_slim$Amount_bins <- cut(
  bondora_slim$Amount, breaks=c(0,2000,4000,6000,8000,10000),
  labels = c(1:5)
)
incidence_amount_bins <- table(bondora_slim$UserName, bondora_slim$Amount_bins)
adj_mat_amount_bins <- incidence_amount_bins %*% t(incidence_amount_bins)
diag(adj_mat_amount_bins) <- 0
```

```r

# Continous Absolute Difference Approach
avg_amount <- tapply(bondora_slim$Amount, bondora_slim$UserName, mean)
adj_mat_amount_diff <- outer(avg_amount, avg_amount,
                             FUN = function(x,y) abs(x - y))
diag(adj_mat_amount_diff) <- 0

# Get Matrix for Differences in Age
incidence_age <- table(bondora_slim$UserName, bondora_slim$Age)
borrower_ages <- as.numeric(colnames(incidence_age)[max.col(incidence_age)])
names(borrower_ages) <- rownames(incidence_age)
adj_mat_age <- outer(borrower_ages, borrower_ages,
                     FUN = function(x, y) abs(x - y))
rownames(adj_mat_age) <- colnames(adj_mat_age) <- names(borrower_ages)

# Get Adjacency Matrix for (same) Gender
incidence_gender <- table(bondora_slim$UserName, bondora_slim$Gender)
adj_mat_gender <- incidence_gender %*% t(incidence_gender)
# Make the matrix binary for homophily
adj_mat_gender <- ifelse(adj_mat_gender > 0, 1, 0)
diag(adj_mat_gender) <- 0

# Get Adjacency Matrix for Differences in Average Loan Duration
borrower_loandur <- sapply(tapply(bondora_slim$LoanDuration,
                                  bondora_slim$UserName, unique),
                           mean)
adj_mat_loandur_diff <- outer(borrower_loandur, borrower_loandur,
                              FUN=function(x,y) abs(x-y))
diag(adj_mat_loandur_diff) <- 0

# Get Adjacency Matrix for Homophily in Restructure of Loans
incidence_restructure <- table(bondora_slim$UserName, bondora_slim$Restructured)
adj_mat_rest <- incidence_restructure %*% t(incidence_restructure)
diag(adj_mat_rest) <- 0

# Save the objects for the QAP Regression in different Script
qap_paths = paste0(obj_paths,"/qap/")
saveRDS(b_indicator, file=paste0(
  "resources/objects/preprocessing/indicator.Rds"))

saveRDS(adj_mat_loan_use, file=paste0(qap_paths,"adj_mat_loanuse.Rds"))
saveRDS(adj_mat_rating, file=paste0(qap_paths,"adj_mat_rating.Rds"))
saveRDS(adj_mat_amount_diff, file=paste0(qap_paths,"adj_mat_amtdiffs.Rds"))
saveRDS(adj_mat_age, file=paste0(qap_paths,"adj_mat_agediffs.Rds"))
saveRDS(adj_mat_gender, file=paste0(qap_paths,"adj_mat_gender.Rds"))
saveRDS(adj_mat_loandur_diff, file=paste0(qap_paths,"adj_mat_loandurdiffs.Rds"))
saveRDS(adj_mat_rest, file=paste0(qap_paths,"adj_mat_rest.Rds"))
# ------------------------------------------------------------------------ #
```

scripts/bondora_preprocessing.R

A.3 Source Code - QAP Linear Regression

```r
#-------------------------------------------------------------------------#
# If not already Installed
install.packages("viridis") # For Colours

# Set colour palette
cols <- viridis::viridis(30)
#-------------------------------------------------------------------------#
# Load Relevant Files
qap_path="resources/objects/qap/"
loan_use_mat <- readRDS(paste0(qap_path,"adj_mat_loanuse.RDS"))
rating_mat <- readRDS(paste0(qap_path,"adj_mat_rating.RDS"))
amt_diffs_mat <- readRDS(paste0(qap_path,"adj_mat_amtdiffs.RDS"))
age_diffs_mat <- readRDS(paste0(qap_path,"adj_mat_agediffs.RDS"))
gender_mat <- readRDS(paste0(qap_path,"adj_mat_gender.RDS"))
loandur_diffs_mat <- readRDS(paste0(qap_path,"adj_mat_loandurdiffs.RDS"))
rest_mat <- readRDS(paste0(qap_path,"adj_mat_rest.RDS"))
#-------------------------------------------------------------------------#
# Create function to determine significance from t-value statistic
t_to_stars <- function(t) {
  stars <- rep("", length(t))
  stars[abs(t) >= 1.96]  <- "*"
  stars[abs(t) >= 2.576] <- "**"
  stars[abs(t) >= 3.291] <- "***"

  return(stars)
}

# Make function to save plots
save_qap_plot <- function(plt_nam) {
  plt <- recordPlot()
  saveRDS(plt, here::here("resources","objects","qap",
                          paste0(plt_nam,".Rds")))
}

var_names <- c("Intercept", "Rating","Loan Amount","Age","Gender",
               "Loan Duration","Restructured")
pred_vars <- list(rating_mat, amt_diffs_mat, age_diffs_mat,
                  gender_mat, loandur_diffs_mat, rest_mat)
#-------------------------------------------------------------------------#
# Basic QAP Linear Regression
qap_m1 <- sna::netlm(y = loan_use_mat,
                     x = list(rating_mat, amt_diffs_mat, age_diffs_mat,
                              gender_mat, loandur_diffs_mat, rest_mat),
                     nullhyp = "qapspp", reps = 2500)
qap_m1$names <- var_names
summary(qap_m1)

results_m1 <- qap_m1$coefficients
names(results_m1) <- var_names
results_sig_m1 <- paste(round(results_m1,3), t_to_stars(qap_m1$tstat))

# Plot Residuals
hist(qap_m1$residuals, main="QAP Residuals", col = cols[15])

# Save Model
saveRDS(qap_m1, file = paste0(qap_path,"qap_m1.RDS"))
#-------------------------------------------------------------------------#
# Standardised QAP Linear Regression
scaled_dep <- scale(loan_use_mat)
scaled_pred <- lapply(pred_vars, scale)

qap_m2 <- sna::netlm(y = scaled_dep,
                     x = scaled_pred,
                     nullhyp = "qapspp", reps = 2500)
qap_m2$names <- var_names
```

```r
summary(qap_m2)

# Plot the result
results_m2 <- qap_m2$coefficients
names(results_m2) <- var_names
results_sig_m2 <- paste(round(results_m2,3), t_to_stars(qap_m2$tstat))

# Plot Residuals
hist(qap_m2$residuals, main="QAP Residuals", col = cols[15])

# Save Model
saveRDS(qap_m2, file = paste0(qap_path,"qap_m2.RDS"))
# ------------------------------------------------------------------------ #
# Plot the result
par(mfrow=c(1,2))

qap_plot_m1 <- barplot(results_m1, col = cols[15],  border = cols[10],
                       ylim = c(min(results_m1) + min(results_m1)*0.15,
                                max(results_m1) + max(results_m1)*0.15),
                       main="QAP Model Results (Unstandardised)")
text(x = qap_plot_m1,
     y = results_m1 + sign(results_m1)*(0.075*diff(range(results_m1))),
     labels = results_sig_m1, font = 2)


qap_plot_m2 <- barplot(results_m2, col = cols[15],  border = cols[10],
                       ylim = c(min(results_m2) + min(results_m2)*0.15,
                                max(results_m2) + max(results_m2)*0.15),
                       main="QAP Model Results (Standardised)")
text(x = qap_plot_m2,
     y = results_m2 + sign(results_m2)*(0.075*diff(range(results_m2))),
     labels = results_sig_m2, font = 2)

save_qap_plot("unstd_std_plot")

par(mfrow=c(1,1))
```

scripts/qap_network_analysis.R

A.4 Source Code - ERGM Network Analysis

```
#-------------------------------------------------------------------------#
# If not already Installed
install.packages("viridis") # For Colours
install.packages("ergm.count")
install.packages("Rglpk") # additional solver for ERGMs
install.packages("here")

# Import the Network and Other Object
ergm_path <- "resources/objects/ergm/"
bondora_net <- readRDS(here::here(
  "resources","objects","preprocessing","bondora_net.Rds"))
b_indicator <- readRDS(here::here(
  "resources","objects","preprocessing","indicator.Rds"))

# Set colour palette
cols <- viridis::viridis(30)

# Determine acceptable core count
n_cores <- parallel::detectCores() - 3 # Leave some out for other processes
print(paste("You have",n_cores,"usable cores"))

# Replicability
seed(42)

# Save plots
save_ergm_plot <- function(plt_nam) {
  plt <- recordPlot()
  saveRDS(plt, here::here("resources","objects","ergm",
                          paste0(plt_nam,".Rds")))
}
#-------------------------------------------------------------------------#
# Copy network for plotting
bondora_plot <- bondora_net

# Get node type for plotting
type_indicator <- ifelse(b_indicator == 2, TRUE,FALSE)
shape <- ifelse(type_indicator,"square","circle")
network::set.vertex.attribute(bondora_plot, "shape", shape)

# Get Category Count for Vertex Size
counts <- summary(bondora_plot ~ b2sociality)
counts_att <- ifelse(type_indicator, counts^0.75, counts^0.6)
network::set.vertex.attribute(bondora_plot, "size", counts_att)

# Colours for the Node Types
plot_cols <- ifelse(type_indicator, cols[5], cols[30])
network::set.vertex.attribute(bondora_plot, "color", plot_cols)

# Legend Plotting
type_legend <- ifelse(type_indicator, "Borrowers", "Loan Type")
type_legend <- as.factor(type_legend)

# Plot the Network
plot(snafun::to_igraph(bondora_plot),
     main = "Bipartite User-LoanUse",
     edge.arrow.size = 0.3,
     edge.color = rgb(0,0,0, alpha = 0.35),
     vertex.frame.color = "black",
     vertex.label = NA,
     vertex.frame.size = 3,
     edge.curved = FALSE,
     layout=igraph::layout.fruchterman.reingold)
legend("bottomleft", legend = levels(type_legend),
       inset = c(0.1, 0.02),
       col = c(cols[30], cols[5]),
```

```
        pch = c(16,15),
        title = "Node Partitions", title.font = 2,
        cex = 0.8,
        lwd = 1,
        bg = rgb(0,0,0, alpha=0.025))
save_ergm_plot("network_plot")

# Summary Statistics
snafun::g_density(bondora_net)
snafun::g_centralize(bondora_net)

# Degree and Betweenness Distribution
deg_dist <- snafun::g_degree_distribution(bondora_net)
bet_dist <- snafun::v_betweenness(bondora_net)

# Other Descriptives
summary(bondora_net ~ b1degree(1:9))
summary(bondora_net ~ b2degree(1:10))

par(mfrow = c(1,2))

hist(deg_dist,
     main = "Unadjusted Degree Distribution",
     xlab = "Node Degree",
     col = cols[15],
     border = cols[10])

hist(bet_dist,
     main = "Unadjusted Betweenness",
     xlab = "Betweenness",
     col = cols[15],
     border = cols[10])

par(mfrow = c(1,1))
# ------------------------------------------------------------------------ #
# Make function to calculate probabilities from log odds
lodds_to_prob <- function(l_odd) {
  return(exp(l_odd) / (1 + exp(l_odd)))
}
# Make function to save ERGM object
save_ergm <- function(object, id) {
  saveRDS(object, file=here::here(
    "resources","objects","ergm",id,".Rds"))
}
# Make function to conduct ERGMs automatically
auto_ergm <- function(model, mcmc, name) {

  # Conducts the GOF Diagnostics and then saves the model,
  # mcmc diagnostics and gof object in a list.
  # This list can be imported as an .RDS object into the R environment

  # Diagnostics
  if (mcmc) {
    ergm::mcmc.diagnostics(model)
  }

  # The GOF must be adjusted otherwise it takes too long
  # We do not limit the GOF by changing its range of parameters
  gof <- ergm::gof(model,
                   control = ergm::control.gof.ergm(
                     nsim = 200,
                     MCMC.burnin = 5000,
                     MCMC.interval = 1000,
                     parallel = n_cores,
                     parallel.type = "PSOCK"
                   ))
```

```
132
133   # Return List to view each item separately
134   result <- list(model, gof)
135   names(result) <- c("model","gof")
136   save_ergm(result, paste0(name,"_panel"))
137
138   return(result)
139 }
140 #------------------------------------------------------------------------------#
141 # Find max degree
142 (max_deg <- max(summary(bondora_net ~ b2factor("b2_loantype"))))
143 #------------------------------------------------------------------------------#
144 # Base Model + GOF
145 formula_base_model <- bondora_net ~ edges
146 base_ergm <- ergm::ergm(formula_base_model)
147 base_ergm_panel <- auto_ergm(base_ergm, mcmc = FALSE, name = "ergm_base")
148 snafun::stat_plot_gof(base_ergm_panel$gof)
149 models = list(base_ergm)
150 texreg::screenreg(models)
151 #------------------------------------------------------------------------------#
152 # Base Model + Edge Counts + GOF
153 #base_model_counts <- ergm::ergm(bondora_net ~ edges, response="frequency",
154 #                                reference = ~ Poisson)
155 #basemodel_counts_gof <- ergm::gof(base_model_counts)
156 #snafun::stat_plot_gof(basemodel_counts_gof)
157 #
158 #texreg::screenreg(list(base_model, base_model_counts))
159 #------------------------------------------------------------------------------#
160 # Iteration 1 + MCMC Diagnostics + GOF
161 model_1_params <- bondora_net ~ edges +
162   # b1 decay can be very low since 9 b2
163   gwb1degree(decay=0.15, fixed=TRUE)
164
165 model_1 <- ergm::ergm(
166   model_1_params,
167
168   # Max b2 degree is 72, so this constraint is reasonable
169   # and helps convergence significantly.
170   # Technically in the Bondora population this can be
171   # far higher but we are studying a subsample.
172   #constraints = ~ bd(minout = 0, maxout = 80),
173
174   control = ergm::control.ergm(
175     # Greater burn-in for cleaner result
176     MCMC.burnin = 20000,
177     # Greater sample size for greater stability
178     MCMC.samplesize = 100000,
179     seed = 42,
180     MCMC.interval = 1000,
181     # Only needed for convergence pvals to improve
182     MCMLE.maxit = 45,
183     # Smaller steps for stability
184     MCMLE.steplength = 0.25,
185     parallel = n_cores,
186     parallel.type = "PSOCK"
187   )
188 )
189
190 model_1_panel <- auto_ergm(model=model_1, mcmc=TRUE, name="ergm_m1")
191 model_1_panel$gof
192 snafun::stat_plot_gof(model_1_panel$gof)
193 texreg::screenreg(list(base_ergm, model_1))
194 #------------------------------------------------------------------------------#
195 # Iteration 2 + MCMC Diagnostics + GOF
196 model_2_params <- bondora_net ~ edges +
197   # low decay important because there is high clustering around low degrees
```

```
  gwb1degree(decay=0.15, fixed=TRUE) +
  # decay should be higher due to wider variation in degree but
  # too high of degree makes the traces concentrated around the tails.
  gwb1dsp(decay=0.5, fixed=TRUE)

model_2 <- ergm::ergm(
  model_2_params,

  # Max b2 degree is 72, so this constraint is reasonable
  # and helps convergence significantly.
  # Technically in the Bondora population this can be
  # far higher but we are studying a subsample.
  constraints = ~ bd(minout = 0, maxout = 80),

  control = ergm::control.ergm(
    # Greater burn-in for cleaner result
    MCMC.burnin = 20000,
    # Greater sample size for greater stability
    MCMC.samplesize = 100000,
    seed = 42,
    MCMC.interval = 1000,
    # Only needed for convergence pvals to improve
    MCMLE.maxit = 45,
    # Smaller steps for stability
    MCMLE.steplength = 0.25,
    parallel = n_cores,
    parallel.type = "PSOCK"
    )
  )

model_2_panel <- auto_ergm(model=model_2, mcmc=TRUE, name="ergm_m2")
snafun::stat_plot_gof(model_2_panel$gof)
model_2_panel$gof
models <- list(base_ergm, model_1, model_2)
texreg::screenreg(models)
# ------------------------------------------------------------------------ #
# Iteration 3 + MCMC Diagnostics + GOF
model_3_params <- bondora_net ~ edges +
  # low decay important because there is high clustering around low degrees
  gwb1degree(decay=0.15, fixed=TRUE) +
  # decay should be higher due to wider variation in degree but
  # too high of degree makes the traces concentrated around the tails.
  gwb1dsp(decay=0.5, fixed=TRUE) +
  # See differences across genders (implicitly, since b1nodemix unavailable)
  b1nodematch("b1_gender", diff=FALSE)

model_3 <- ergm::ergm(
  model_3_params,

  # Max b2 degree is 72, so this constraint is reasonable
  # and helps convergence significantly.
  # Technically in the Bondora population this can be
  # far higher but we are studying a subsample.
  constraints = ~ bd(minout = 0, maxout = 80),

  control = ergm::control.ergm(
    # Greater burn-in for cleaner result
    MCMC.burnin = 20000,
    # Greater sample size for greater stability
    MCMC.samplesize = 100000,
    seed = 42,
    MCMC.interval = 1000,
    # Only needed for convergence pvals to improve
    MCMLE.maxit = 45,
    # Smaller steps for stability
    MCMLE.steplength = 0.25,
```

```
    parallel = n_cores,
    parallel.type = "PSOCK"
  )
)

model_3_panel <- auto_ergm(model=model_3, mcmc=TRUE, name="ergm_m3")
snafun::stat_plot_gof(model_3_panel$gof)
model_3_panel$gof
models <- list(base_ergm, model_1, model_2, model_3)
texreg::screenreg(models)
# ------------------------------------------------------------------------ #
# Iteration 4 + MCMC Diagnostics + GOF
model_4_params <- bondora_net ~ edges +
  # low decay important because there is high clustering around low degrees
  gwb1degree(decay=0.15, fixed=TRUE) +
  # decay should be higher due to wider variation in degree but
  # too high of degree makes the traces concentrated around the tails.
  gwb1dsp(decay=0.5, fixed=TRUE) +
  # See differences across genders (implicitly, since b1nodemix unavailable)
  b1nodematch("b1_gender", diff=TRUE) +
  # See if higher ages make a difference
  b1cov("b1_age")

model_4 <- ergm::ergm(
  model_4_params,

  # Max b2 degree is 72, so this constraint is reasonable
  # and helps convergence significantly.
  # Technically in the Bondora population this can be
  # far higher but we are studying a subsample.
  constraints = ~ bd(minout = 0, maxout = 80),

  control = ergm::control.ergm(
    # Greater burn-in for cleaner result
    MCMC.burnin = 20000,
    # Greater sample size for greater stability
    MCMC.samplesize = 100000,
    seed = 42,
    MCMC.interval = 1000,
    # Only needed for convergence pvals to improve
    MCMLE.maxit = 45,
    # Smaller steps for stability
    MCMLE.steplength = 0.25,
    parallel = n_cores,
    parallel.type = "PSOCK"
  )
)

model_4_panel <- auto_ergm(model=model_4, mcmc=TRUE, name="ergm_m4")
model_4_panel$gof
snafun::stat_plot_gof(model_4_panel$gof)
models <- list(base_ergm, model_1, model_2, model_3, model_4)
texreg::screenreg(models)
# ------------------------------------------------------------------------ #
```

scripts/ergm_network_analysis.R

References

Adler, P. S., & Kwon, S.-W. (2002). Social capital: Prospects for a new concept. *Academy of Management Review*, *27*(1), 17–40.

Akerlof, G. A. (1978). The market for "lemons": Quality uncertainty and the market mechanism. In *Uncertainty in economics* (pp. 235–251). Elsevier.

Bachmann, A., Becker, A., Buerckner, D., Hilker, M., Kock, F., Lehmann, M., … Funk, B. (2011). Online peer-to-peer lending – a literature review. *Journal of Internet Banking and Commerce*, *16*.

Barasinska, N., & Schaefer, D. (2010). Does gender affect funding success at the peer-to-peer credit markets? Evidence from the largest german lending platform. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.1738837

Cai, S., Lin, X., Xu, D., & Fu, X. (2016). Judging online peer-to-peer lending behavior: A comparison of first-time and repeated borrowing requests. *Information & Management*, *53*(7), 857–867. https://doi.org/10.1016/j.im.2016.07.006

Chen, X., Zhou, L., & Wan, D. (2016). Group social capital and lending outcomes in the financial credit market: An empirical study of online peer-to-peer lending. *Electronic Commerce Research and Applications*, *15*, 1–13. https://doi.org/10.1016/j.elerap.2015.11.003

De Liu, Brass, D. J., Lu, Y., & Chen, D. (2015). Friendships in online peer-to-peer lending. *MIS Quarterly*, *39*(3), 729–742. Retrieved from https://www.jstor.org/stable/26629629

Devenow, A., & Welch, I. (1996). Rational herding in financial economics. *Papers and Proceedings of the Tenth Annual Congress of the European Economic Association*, *40*(3), 603–615. https://doi.org/10.1016/0014-2921(95)00073-9

Herzenstein, M., Dholakia, U. M., & Andrews, R. L. (2011). Strategic herding behavior in peer-to-peer loan auctions. *Journal of Interactive Marketing*, *25*(1), 27–36. https://doi.org/10.1016/j.intmar.2010.07.001

Jackson, M. O. (2011). An overview of social networks and economic applications. In *Handbook of social economics* (Vol. 1, pp. 511–585). Elsevier. https://doi.org/10.1016/B978-0-444-53187-2.00012-7

Mingfeng Lin, N.R. Prabhala, & Siva Viswanathan. (2009). *Social networks as signaling mechanisms: Evidence from online peer-to-peer lending*. Retrieved from https://pages.stern.nyu.edu/~bakos/wise/papers/wise2009-p09_paper.pdf

Nahapiet, J., & Ghoshal, S. (1998). Social capital, intellectual capital, and the organizational advantage. *Academy of Management Review*, *23*(2), 242–266.

Podolny, J. M. (1993). A status-based model of market competition. *American Journal of Sociology*, *98*(4), 829–872. https://doi.org/10.1086/230091

Pope, D., & Sydnor, J. (2008). What's in a picture? Evidence of discrimination from prosper.com. *Monetary Economics eJournal*, *46*. https://doi.org/10.1353/jhr.2011.0025

Putnam, R. D. (2015). Bowling alone: America's declining social capital. In *The city reader* (pp. 188–196). Routledge.

Siddhartha, M. (n.d.). Bondora peer to peer lending loan data. Retrieved October 9, 2025, from https://www.kaggle.com/datasets/sid321axn/bondora-peer-to-peer-lending-loan-data

B Technology Statement

During the preparation of this work, we used ChatGPT in order to generate select parts of the R script utilised to process the dataset. Specifically, the tool was used to transform the processed dataset into a format that `igraph` would accept as a network object. No AI

tool was utilised to write parts of the report. The following parts of the assignment were affected/generated by AI tool usage: **DATASET;** the data described within this section was processed partly by some code drafted by ChatGPT and edited by the group. After using this tool/service, **Samir Sabitli** evaluated the validity of the tool's outputs, including the sources that generative AI tools have used, and edited the content as needed. As a consequence, **Samir Sabitli** takes full responsibility for the content of their work.